Zero-Sum Regression
Scale Invariant Molecular Data Analysis

Dissertation zur Erlangung des Doktorgrades
der Naturwissenschaften (Dr. rer. nat.)
der Fakultät für Physik
der Universität Regensburg

vorgelegt
von
Thorsten Rehberg

aus München

im Jahr 2018

Promotionsgesuch eingereicht am: 22.05.2018
Die Arbeit wurde angeleitet von: Prof. Dr. Elmar Lang

Termin Promotionskolloquium:

Abstract

In biomedicine, it is still an outstanding issue that the absolute scale of omics data gets lost due to technical limitations. This causes that the original scale first has to be approximated by normalization techniques before analysis methods can be applied. However, there are competing normalization strategies based on different assumptions about the structure of the underlying data. Due to these different assumptions, normalization methods can yield different results, which can also affect the outcome of concluding analysis methods. Thus, another concept is to resolve this issue by using scale invariant data analysis methods.

This thesis details how generalized linear regression methods can be extended with a scale invariance for omics data by enforcing an additional constraint called zero-sum. This constraint was first proposed by Lin et al. [53] for compositional data and by M. Altenbuchinger and T. Rehberg in [3] for the application on omics data to circumvent scaling and normalization.

The first chapter outlines scaling and normalization and exemplifies in a simulation the concept of scale invariance. Afterwards, the state-of-the-art procedure for solving linear, binomial logistic, multinomial logistic and Cox proportional hazard regression is detailed in the second chapter. The third chapter shows how the zero-sum constraint can be incorporated into these regression types and how an efficient solving strategy based on a coordinate descent algorithm can be developed. Moreover, it is shown how convergence issues of a naive coordinate descent approach can be resolved by using search space rotations, a concluding local search procedure and *warm starts*. Finally, the convergence behavior is evaluated by comparing coordinate descent with general purpose optimization methods.

In chapter four, the scale invariance of zero-sum regression is examined in several simulation scenarios where increasingly larger sample-wise alterations are applied. These alterations cause regressions without scale invariance to loose predictive power, while zero-sum regressions are not affected and maintain their predictivity.

Finally, zero-sum regression is applied on omics data sets to demonstrate the advantages of scale invariance. First, the normalization independency of zero-sum regression is shown on microbiome and metabolomics data sets. Afterwards, it is shown how the generalized lasso regularization in combination with the zero-sum constraint can be applied on NMR spectra to improve linear models. Next, a DNA methylation data set is analyzed using zero-sum multinomial regression to mitigate the effects of tissue background contamination in order to better distinguish between primary tumor and metastatic cells. Finally, zero-sum Cox proportional hazard regression is applied on gene expression data and corresponding survival data of lymphoma patients.

All algorithms presented in this thesis have been implemented in the software *zeroSum*, which is publicly available.

Preface

Acknowledgement:
This work was carried out in the Department of Statistical Bioinformatics of the Institute of Functional Genomics at the University of Regensburg. I am very grateful to Rainer Spang for giving me the opportunity to work in his group and to write this thesis under his guidance.
Moreover, I would like to thank my supervisors Rainer Spang and Elmar Lang for their constant advice and Michael Altenbuchinger for the good cooperation.
I thank all past and present colleagues for the good working atmosphere and the fruitful scientific and non-scientific discussions. Special thanks go to Simon Obermeyer, Matthias Jobst, Vincent Müller, Paula Perez-Rubio and Evelyn Rößler for proofreading and commenting on drafts of this thesis.

Publications and Contribution:
So far, the algorithms and their implementations in the software *zeroSum* presented in this thesis have contributed to the publications [3, 4, 88]. My contribution to these publications is the development of the algorithms and the software *zeroSum*, which therefore is the focus of this thesis. To illustrate *zeroSum* in practice the following parts of these publications are also presented:

- The concept of the simulations shown in chapter 4 is part of the publication [3], but has been extended to binomial logistic, multinomial logistic and Cox proportional hazard regression.
- The application of *zeroSum* on microbiome data in section 5.1 is also part of the publication [3].
- The application of *zeroSum* on metabolomics data in section 5.2 was first published in [88].

All applications and simulations presented in this thesis have been performed with version 1.1.1 of the software *zeroSum*, which is publicly available at:

`https://github.com/rehbergT/zeroSum`

Contents

1 Introduction

This thesis presents an approach for combining generalized linear regression with a scale invariance constraint which addresses the problem of normalization and reference point selection in the statistical analysis of molecular measurements. Such measurements allow to gain new insights into the inner workings of a cell and are crucial for answering biological and medical questions. For this purpose, molecular profiles are generated using high-throughput technologies and analyzed with statistical methods. However, before such methods can be applied, preprocessing steps, such as normalization and reference point selection, are necessary to compensate for unwanted alterations caused by technical limitations. Resulting from these limitations, the true scale of the data gets lost and therefore has to be approximated by normalization methods. However, the choice of a suitable normalization is not clear, since there are competing normalization strategies based on different assumptions about the structure of the underlying data. Due to these different assumptions, normalization methods can yield significantly different results [11, 74]. Therefore, the outcome of a concluding analysis is also affected by the chosen normalization. Another approach for avoiding normalization is to employ scale invariant statistical methods, which additionally prevent that the results are distorted by the scale of the data.
This thesis shows how scale invariance is achieved in generalized linear models by incorporating the zero-sum constraint. Moreover, regression algorithms are proposed and a reference implementation is provided as a program called zeroSum. The first chapter contains a brief overview of the different types of molecular measurements and illustrates normalization and scaling problems.

1.1 Molecular Measurements

During the last decade, a better understanding of the molecular dynamics in cells and their effects on whole organisms has been acquired by the rapid development of novel high-throughput technologies. These technologies allow to assess high dimensional molecular profiles on the level of transcriptomics, proteomics, metabolomics and others, which are summarized under the umbrella term *omics* data [1, 42].
Each *omics* field primarily focuses on a specific molecular level of a cell. In simple terms, these levels form a chain of effects, where the lowest element is the genome containing the genetic information. This information is stored in the DNA as a sequence of four different nucleobases (adenine, cytosine, guanine, thymine) and is the subject of the research field genomics. Small sections of the genome – called genes – are accessed by proteins known as transcription factors and RNA polymerases, which copy the sequence of a gene and store it as RNA transcripts. Due to a complex regulating network, genes become differently transcribed and the research field analyzing the resulting distribution of RNA transcripts is termed transcriptomics. On the next level, these RNA transcripts are translated by ribosomes to proteins, creating a distribution of proteins that is investigated by the proteomics research field. One of the highest levels of *omics* research is metabolomics, which focuses on the metabolite composition in cells.
Nowadays, the molecular composition of each *omics* level can be almost completely monitored by high-throughput techniques. As an example, DNA or RNA sequencing is used for capturing genomics or transcriptomics profiles, whereas mass- or NMR-spectrometry are applied to produce proteomics and metabolomics profiles. From these profiles not only unknown interactions and properties of cells can be discovered, but also defects like mutations detected. These defects can be the cause for many diseases,

such as cancer [13, 14]. The capability to identify such defects and the insights gained from *omics* research are shaping the vision of personalized and precision medicine in which every patient gets an individually adapted therapy targeting the specific molecular cause of a disease [12, 14, 28, 40, 50].
In order to identify the cause of a disease it is important to discover the differences between healthy and diseased tissue. Therefore, it is required to capture molecular profiles of each tissue type in sufficient numbers and to probe for significant differences. However, sample preparation steps and measurement procedures often cause molecular data being systematically biased. Hence, the data first has to be calibrated to a common scale in order make the molecular profiles evaluable by well-established statistical methods [11, 16, 54, 65].

1.2 Normalization and Scaling Issues of Omics Data

The measurement process of *omics* data involves multiple steps, which can distort the scale of the resulting data due to technical limitations. An example of such a limitation can be the varying sequencing depth of RNA sequencing data, which is a measure for how often a transcript has been detected on average [54]. A biological limitation, for instance, is the varying concentration of metabolites in urine specimen of humans, which is affected by different kinds of drinking behavior, transpiration of the examined persons [16, 68]. As a consequence, the data has to be rescaled to compensate for these distortions to make the data comparable. Such rescaling methods are known as normalization.
Multiple normalization strategies have been developed to equalize the scale between different *omics* profiles [11]. For example so called housekeeping genes, which are assumed to be mandatory for basic cell functions and therefore should be equally transcribed in every cell, are used as a reference to bring gene expression data on the same scale. However, the search for such general applicable housekeeping genes has been of limited success [20, 23]. A similar normalization strategy to adjust metabolomics data is to use reference metabolites which are assumed to be equally concentrated in each specimen [82]. However, identifying perfect reference metabolites is also unfeasible. The concentration of the commonly used reference metabolite creatinine, for example, depends on sex, age and other individual characteristics of the probands and is thus not an indisputable reference [18, 72, 81].
Moreover, all data has to be scaled to a common reference point, like a certain number of cells, a certain amount of RNA or a certain amount of blood. However, the choice of a reference point is not always clear and can have a significant influence on the drawn conclusions. Such a reference point used in transcriptomics, for instance, is the measurement of a fixed amount of RNA under the assumption that the total amount of RNA per cell is roughly the same. However, this is not always the case and it was shown that the total amount of RNA in cells can be amplified by a factor of 2–3 by inducing the expression of the gene *MYC* [52, 61, 74]. Such a change would not be detected if the total amount of RNA was used as a reference. If, on the other hand, the number of cells is taken as a reference point, the change of the amount of RNA per cell becomes detectable [74].
Furthermore, these scaling issues cause problems beyond the above mentioned ones: More and more data sets are publicly available but results of a dataset like a biomarker for a specific disease are difficult to transfer to other data, since the results depend on the scale of the initial data [4, 27]. One approach to tackle this issue is the use of *addon* normalization procedures, which adjust new data so that the scale of it conforms to the scale of original data [36, 46]. A related problem is the transfer of results learned on data which has been generated with one measurement platform like a molecular signature to data that was generated with another measurement platform with a different sensitivity [4].
In summary, the selection of an appropriate normalization method and scaling of *omics* data is still an unsolved issue and susceptible to many flaws. For that reason, the aim of the method proposed in this thesis is to bypass these issues by making data analysis methods robust against scaling problems.

Therefore, generalized linear regression is extended by an additional constraint called zero-sum, which makes regression scale invariant. This constraint was first proposed by Lin et al. [53] for lasso regularized linear regression and for the application on compositional data. M. Altenbuchinger and myself showed in [3] that the zero-sum constraint can also be used to generate scale invariant linear models for *omics* data. My contribution to [3] and to the second related publication [88] was the algorithmic concept and software implementation. Therefore, the focus of this thesis is on the zero-sum regression optimization problem and the corresponding algorithmic approaches. Moreover, this thesis shows how this constraint can be applied to generalized linear models, which, for example, can be used for classification (binomial logistic and multinomial logistic regression) and survival analysis (Cox proportional hazard regression). This extended generalized linear regression is referred to in this thesis as zero-sum regression and the corresponding regression software is named *zeroSum*.

1.3 Influence of Scaling and Normalization on Data Analysis of Omics Profiles

In this section the influence of normalization methods on results of a common *omics* data analysis workflow is exemplified in a simulation.

Typically, *omics* data has to be log-transformed in order to be accessible by linear models. The reason for that is, that *omics* data ranges over several orders of magnitudes and often exhibits a skewed distribution. Both issues can be resolved by applying a log-transformation [19, 49, 80]. Hence, a multiplicative scaling of samples becomes an additive shift on the log-transformed data. Thus, a scale invariant method has to be insensitive to additive sample-wise data shifts.

For demonstrating the effects of different normalizations, log-transformed three dimensional data for 8 observations is simulated as follows: for all observations the feature x is selected randomly from a uniform distribution on the interval $[0, 1]$. To imitate a housekeeping gene, feature y is sampled from a normal distribution with the mean 0.5 and a standard deviation of 0.1. To simulate two different groups, A and B, feature z of half of the observations is generated from a uniform distribution on the interval $[-1, 0]$ (group A), while the other half is sampled from a uniform distribution on the interval $(0, 1]$ (group B). This unaltered data is shown as dark blue and dark red spheres in figure 1.1. Afterwards, mean centering is applied as an example of a normalization method, which is based on the assumption that the total amount of features should be more or less the same per sample. The resulting shifts are shown in figure 1.1 by arrows and the altered data is shown in lighter colors. Finally, a separating plane generated using logistic regression is shown as blue plane.

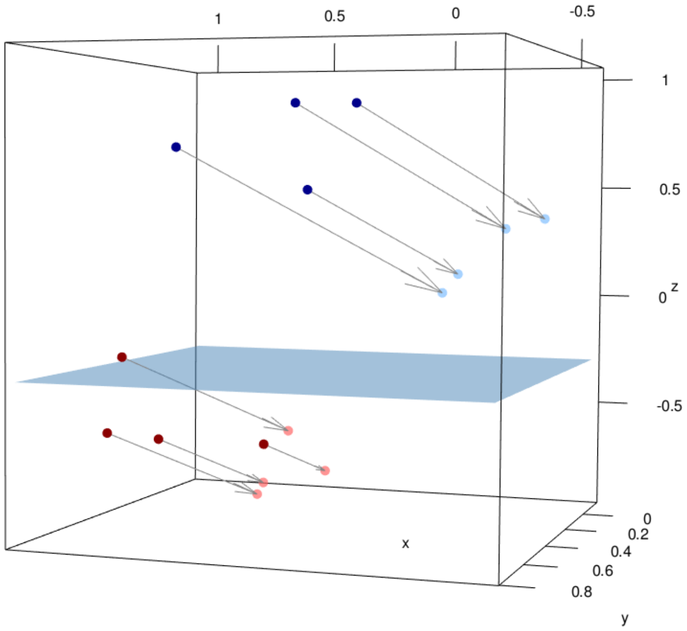


Figure 1.1: Shown are the samples of the groups A and B as dark red and dark blue spheres. In light colors the mean centered data is shown. The blue plane shows where a logistic regression classifier separates the mean centered data into group A and B. Underneath the plane a sample is labeled to group A and above the plane to group B.

As another normalization, the mean centering is replaced by a housekeeping normalization using feature y as reference. Such a normalization is shown in figure 1.2 by cyan and yellow colored spheres and the corresponding separating plane is shown in red. Note that, two different normalization methods lead to two different results, which in this case are two different separating planes.
Data shifts caused by normalization alter the data only in one direction, which is associated with uncertainties due to technical limitations. Thus, the position of data varies along this direction.
By using zero-sum regression separating planes which are parallel to the direction of normalization can be calculated. Such a plane is shown in figure 1.3 in green. Note that the same plane is obtained irrespectively of whether zero-sum regression is applied on the raw, mean-centered or housekeeper normalized data, since scaling has no influence on zero-sum regression.
This is the fundamental principle of zero-sum regression and this thesis describes how this invariance can be enforced in a generalized linear regression framework.

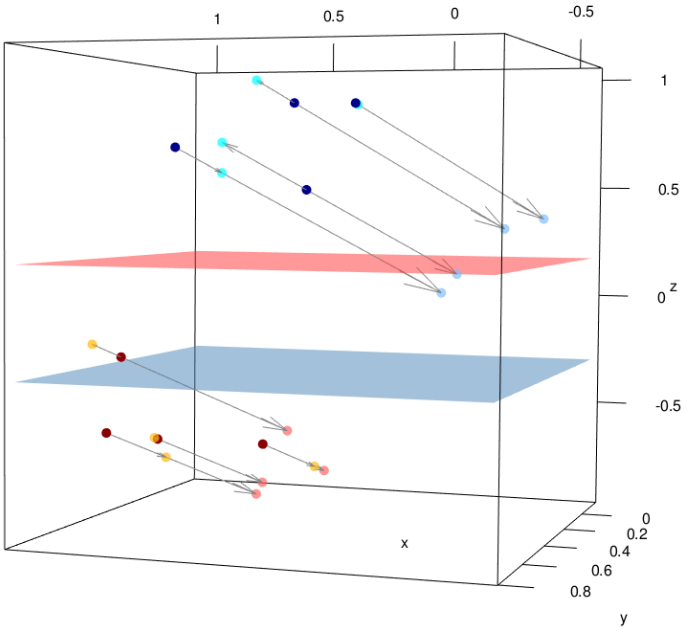


Figure 1.2: The same data and classifier as in figure 1.1. Additionally, the same data but with a housekeeping normalization is shown as yellow dots for group A and cyan for group B. The red plane is the separating plane of a logistic regression classifier learned on the housekeeping normalized data.

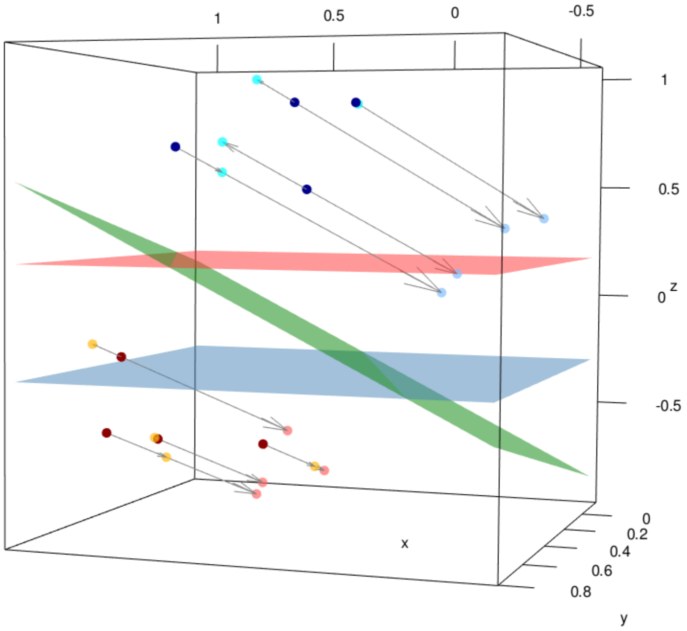


Figure 1.3: The same data and classifier as in the figures 1.1 and 1.2. A scaling invariant classifier learned with zero-sum logistic regression is shown in green.

1.4 Thesis Organization

This thesis develops the mathematical concepts and algorithmic solutions of zero-sum regression and demonstrates the advantages not only in simulations but also on *omics* data. The structure of this thesis is:

Generalized Linear Regression: Chapter 2 provides an overview of generalized linear models and in particular details linear, binomial logistic, multinomial logistic and Cox proportional hazard models. Moreover, the elastic net and generalized lasso regularizations are described. Furthermore, the current state-of-the-art for fitting these models to data based on coordinate descent algorithms is described.

Zero-Sum Regression: The third chapter shows how the zero-sum constraint can be incorporated in the cost functions of generalized linear models and how an efficient algorithm based on coordinate descent can be developed. Moreover, it is detailed how convergence problems of a naive coordinate descent approach can be resolved by using search space rotations, a concluding local search procedure and warm starts. Finally, the convergence behavior is evaluated by comparing coordinate descent with general purpose optimization methods.

Simulations: In chapter 4 the scale invariance of zero-sum regression is examined in several simulation scenarios where increasingly larger sample-wise alterations are applied and the results compared to traditional generalized linear regression.

Applications: In chapter 5, zero-sum regression is applied on *omics* data sets to demonstrate the advantages of scale invariance. First, the normalization independency of zero-sum regression is shown on a microbiome and metabolomics data set. Afterwards, it is shown how the generalized lasso regularization in combination with the zero-sum constraint can be applied on NMR spectra to improve linear models. Next, a DNA methylation data set is analyzed using zero-sum multinomial regression to mitigate the effects of tissue background contamination. The last data set covers gene expression data and survival data of lymphoma patients and is evaluated using zero-sum Cox proportional hazard regression.

2 Generalized Linear Models

Regularized generalized linear models are well established for the analysis of molecular profiles, due to their reasonable performance on high dimensional data and particularly due to their clear interpretability. In contrast to normal linear models, generalized linear models are highly versatile. Logistic regression can be used, for example, for classification and Cox proportional hazard regression can be used for survival analysis. However, one of the main obstacles for the application of generalized linear models on omics data is that the amount of samples is, in contrast to the number of features, almost always very limited. Therefore, the regression problems are underdetermined, which makes regularization methods necessary to obtain unique solutions and to prevent overfitting.

The first part of this chapter outlines generalized linear models and ordinary linear regression. Afterwards, the elastic net regularization and the state-of-the-art approach for solving the resulting optimization problem is detailed. The subsequent sections describe how this approach can be extended to binomial logistic, multinomial logistic and Cox proportional hazard regression. Concluding, the generalized lasso regularization and the effects of data standardization on linear models are described.

2.1 Generalized Linear Models

Generalized linear models estimate a response y_i of a sample i by a corresponding predictor $\boldsymbol{x}_i$ with a link function f and a set of coefficients $(\beta_0, \boldsymbol{\beta})$ [39, 58]:

$$y_i = f(\beta_0 + \boldsymbol{x}_i^T \boldsymbol{\beta}) + \epsilon_i\,. \tag{2.1}$$

Throughout this thesis the following conventions are used: the predictor $\boldsymbol{x}_i$ of a sample i is a vector of length p, which for example contains a molecular profile with p features. The predictors of all samples can be combined to a $N \times p$ matrix $\boldsymbol{x}$, where N is the number of samples. Individual components of $\boldsymbol{x}$ are denoted as x_{ij} and, for instance, could be the expression of the gene j of the sample i. The coefficient vector $\boldsymbol{\beta}$ is of length p and is used to weight the features of $\boldsymbol{x}_i$. Moreover, the responses and errors of all samples are combined to a vector $\boldsymbol{y}$ and $\boldsymbol{\epsilon}$, which are both of length N. The concept of generalized linear regression is to adjust the coefficients $\boldsymbol{\beta}$ and the intercept β_0 to minimize the error $\boldsymbol{\epsilon}$ and thus to approximate $\boldsymbol{y}$ as accurate as possible.

The type of the response $\boldsymbol{y}$ and the used link function can be very diverse for different regression methods: in the linear regression case the response $\boldsymbol{y}$ is a numeric value and the identity function is used as link function to predict $\boldsymbol{y}$. Furthermore, binomial logistic and multinomial logistic regression are used for classifying samples into different categories. Thus, the response $\boldsymbol{y}$ annotates a group, which is encoded with 0 for the first group, 1 for the second group etc. and can, for example, represent different disease types. In this thesis the common abbreviations logistic regression and multinomial regression are consistently used. The difference between these two types is that logistic regression is limited to only two cases, whereas multinomial regression is a generalization of logistic regression and is not limited in the number of classes. However, the binomial case is very common and can be solved more efficiently by an adapted algorithm. Hence, logistic and multinomial regression will be treated separate throughout this thesis.

This thesis covers linear, logistic, multinomial and Cox proportional hazard regression, but the approaches presented in this thesis can be transferred to other generalized linear regression types.

2.2 Linear Regression

Standard linear regression uses the identity function as link function and the response is a numeric value. Therefore, (2.1) simplifies to

$$y_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} + \epsilon_i \,. \tag{2.2}$$

For minimizing the error ϵ_i across all samples, the residual sum of square (RSS) is commonly used as cost function. The optimal values for the coefficients β_0 and $\boldsymbol{\beta}$ are thus given by

$$\min_{(\beta_0,\boldsymbol{\beta})\in\mathbb{R}^{p+1}} \mathrm{RSS}(\beta_0,\boldsymbol{\beta}) = \min_{(\beta_0,\boldsymbol{\beta})\in\mathbb{R}^{p+1}} \left[\sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \right]. \tag{2.3}$$

The intercept can be incorporated into the predictor matrix $\boldsymbol{x}$ by appending a column of 1s on the left and including β_0 into $\boldsymbol{\beta}$ as first component. The resulting RSS in matrix notation is

$$\mathrm{RSS}(\boldsymbol{\beta}) = ||\boldsymbol{y} - \boldsymbol{x\beta}||_2^2 \,. \tag{2.4}$$

This cost function can be solved analytically if $\boldsymbol{x}$ has full column rank. In this case, the optimal solution $\hat{\boldsymbol{\beta}}$ is

$$\hat{\boldsymbol{\beta}} = \left(\boldsymbol{x}^T \boldsymbol{x} \right)^{-1} \boldsymbol{x}^T \boldsymbol{y} \,. \tag{2.5}$$

Since *omics* data sets include almost always more features than samples, the problem is underdetermined and many equivalent solutions exist. To tackle this issue, the elastic net regularization, which is detailed in the next section, can be used. Overdetermined systems are not covered in this thesis, since they have no practical relevance for *omics* data.

2.3 The Elastic Net

As described in the last section, a regression problem with more samples than variables is underdetermined. To solve such a problem, additional assumptions about the structure of the effects, which are modeled by the coefficients, can be incorporated into the corresponding cost function. This is achieved by imposing additional data independent constraints on the coefficients β. Such constraints are called regularization and can, for instance, be incorporated into cost functions by using penalties [32, 39]. Another advantage is that regularization is effective in preventing overfitting [32, 60].

The widely adopted elastic net regularization was proposed by Zou and Hastie [90] and is used in many well established machine learning programs like the R-package *glmnet* [26] or the python library *scikit-learn* [63]. The elastic net is a generalization of two regularizations: the ridge regularizations [34], which restricts the square of the l_2 norm of the coefficients, and the lasso regularization (least absolute shrinkage and selection operator) proposed by Tibshirani [76], which limits the l_1 norm of the coefficients. This is achieved by combining the two regularizations with a weight factor α:

$$P_{\text{elastic net}}(\boldsymbol{\beta})_\alpha = \frac{1-\alpha}{2} \underbrace{||\boldsymbol{\beta}||_2^2}_{\text{ridge}} + \alpha \underbrace{||\boldsymbol{\beta}||_1}_{\text{lasso}} \qquad \text{with } \alpha \in [0,1] \,. \tag{2.6}$$

For $\alpha = 1$ the elastic net corresponds to the lasso, while for $\alpha = 0$ the elastic net is equivalent to the ridge regularization. Both regularizations have a shrinkage effect on the coefficients, but behave differently in the case of correlated predictors [26, 32, 39].
The ridge regularization has the effect that coefficients of correlated features are forced to similar values. This becomes obvious by looking at the case of two perfectly correlated features, which are on the same scale and both perfectly predict the corresponding response with a coefficient a.
The residual sum of squares of an estimator $\boldsymbol{\beta}$ does not change, no matter how the amount of a is distributed among these two coefficients as long as the sum of these two coefficients stays the same. However, the smallest square sum is obtained by setting both coefficients to the same value.
The lasso regularization induces a sparse set of coefficients, but has issues with correlated features. Using the lasso in the above mentioned case yields no unique solution and randomly sets one coefficient to a and the other to zero. This problem can be prevented by using the elastic net and adding a small ridge regularization by assigning a value to α, that is a bit lower than one [26, 32]. By using this approach, both coefficients would be set to the same value, but the whole set of coefficients would still be sparse.
In the following, the elastic net with *a priori* defined feature weights $\boldsymbol{v}$ ($v_j \geq 0$ for all j) is used:

$$P_{\text{elastic net}}(\boldsymbol{\beta})_\alpha = \sum_{j=1}^{p} v_j \Big(\frac{1-\alpha}{2} \beta_j^2 + \alpha |\beta_j| \Big) . \tag{2.7}$$

The feature weights allow to incorporate variables, which should not be penalized and always be part of the model. This can be achieved by assigning a value of zero to the corresponding component in $\boldsymbol{v}$. As default, and if not stated otherwise, these weights are set to 1 throughout this thesis.

2.4 Cost Function

The central concept is to obtain a generalizable cost function by extending the residual sum of squares with observation weights w_i. The reason for this is that these weights can be used to extend this cost function to logistic, multinomial and Cox proportional hazard regression. This extended cost function will be referred to in the following as weighted residual sum of squares (WRSS) and is defined as

$$\text{WRSS}(\beta_0, \boldsymbol{\beta}) = \frac{1}{2} \sum_{i=1}^{N} w_i \Big(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \Big)^2 . \tag{2.8}$$

The factor $1/2$ is a convention and cancels when calculating the derivative of the WRSS. The additional observation weights w_i ($w_i > 0$ for all i) can also be used to incorporate data specific properties. For example, in the case of 10 samples of class A and 20 samples of class B, the weights of the first group can be set to 1 and the weights of the second group can be set to 0.5 to balance out the contribution to the cost function of the different amount of samples per group.
The WRSS can be combined with the elastic net (2.7) by using the additional weight λ. The cost function $\mathcal{H}$ of the resulting optimization problem is thus defined as [26]

$$\min_{(\beta_0, \boldsymbol{\beta}) \in \mathbb{R}^{p+1}} \mathcal{H}_\lambda(\beta_0, \boldsymbol{\beta}) = \min_{(\beta_0, \boldsymbol{\beta}) \in \mathbb{R}^{p+1}} \Big[\text{WRSS}(\beta_0, \boldsymbol{\beta}) + \lambda P_{\text{elastic net}}(\boldsymbol{\beta})_\alpha \Big] \tag{2.9}$$

$$= \min_{(\beta_0, \boldsymbol{\beta}) \in \mathbb{R}^{p+1}} \Bigg[\frac{1}{2} \sum_{i=1}^{N} w_i \Big(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \Big)^2 + \lambda \sum_{j=1}^{p} v_j \Big(\frac{1-\alpha}{2} \beta_j^2 + \alpha |\beta_j| \Big) \Bigg] . \tag{2.10}$$

The minimum of this cost function corresponds to the optimal solution of the coefficients $(\beta_0, \boldsymbol{\beta})$ for a specific value of λ. By varying the parameter λ it is possible to control the trade-off between the prediction accuracy of the linear model on the training data and the shrinkage effect of the regularization. Such parameters, which allow to alter the optimization problem, are known as hyperparameters and are an additional optimization problem.

2.5 Coordinate Descent

One of the most efficient solvers for regularized regression problems (2.10) are coordinate descent algorithms [25, 26]. For the application of such algorithms the partial derivatives of the cost function are required. By setting the k-th partial derivative to zero and solving for the corresponding coefficient under the assumption that all other coefficients are held fixed, the following update scheme for the local optimal value $\hat{\beta}_k$ for the k-th coefficient is obtained (see [25, 26] and for the derivation A.1):

$$\hat{\beta}_k \leftarrow \frac{1}{\sum_{i=1}^N w_i x_{ik}^2 + \lambda(1-\alpha)v_k} \cdot \begin{cases} \Big(\sum_{i=1}^N w_i x_{ik}\Big(y_i - \beta_0 - \sum\limits_{\substack{j=1 \\ j\neq k}}^{p} \beta_j x_{ij}\Big) + \lambda\alpha v_k\Big) & \text{if } f_k < 0 \wedge g_k < |f_k| \\ \Big(\sum_{i=1}^N w_i x_{ik}\Big(y_i - \beta_0 - \sum\limits_{\substack{j=1 \\ j\neq k}}^{p} \beta_j x_{ij}\Big) - \lambda\alpha v_k\Big) & \text{if } f_k > 0 \wedge g_k < |f_k| \\ 0 & \text{if } g_k \geq |f_k| \end{cases}, \tag{2.11}$$

with

$$f_k := \sum_{i=1}^N w_i x_{ik}\Big(y_i - \beta_0 - \sum_{\substack{j=1 \\ j\neq k}}^{p} \beta_j x_{ij}\Big), \tag{2.12}$$

$$g_k := \lambda\alpha v_k \,. \tag{2.13}$$

The intercept β_0 can be updated in the same way (see (A.1.6) in A.1):

$$\hat{\beta}_0 \leftarrow \frac{\sum_{i=1}^N w_i\Big(y_i - \sum_{j=1}^p \beta_j x_{ij}\Big)}{\sum_{i=1}^N w_i} \,. \tag{2.14}$$

Since the cost function is convex, this update scheme can be iterated over all coefficients until the optimum is reached. A more efficient approach is based on the sparseness assumption and uses active set cycling to enhance the performance [26, 47, 55]. The structure of such an algorithm is:

1. Start with $\boldsymbol{\beta} = \vec{0}$ or initialize $(\beta_0, \boldsymbol{\beta})$ with values obtained by a previous calculation as *warm start*.
2. Do one complete cycle over all coefficients and update each with (2.11).
3. Cycle over all $\beta_j \neq 0$ and update each with (2.11) until convergence.
4. Repeat a complete cycle: if the active set changes go back to 3. else your done.

By iterating only on the active set, considerable performance improvements can be achieved.

2.6 Optimizing the Hyperparameter λ

The cost function (2.10) depends on the hyperparameter λ, for which an optimal value can be determined by minimizing the cross-validation (CV) error [26, 32, 39].

Cross-validation is a method for assessing how well a model learned on training data can be transferred to independent data, which has not been used for fitting the model. Therefore, the samples are randomly divided into M subsets and one subset separated from the data set. Usually, 10 subsets are used ($M = 10$), but M has to be less than or equal to the number of samples N. The remaining samples N^* are used to learn a model, which is then applied on the separated $N - N^*$ samples to determine the weighted residual sum of squares (WRSS) defined by (2.8). This procedure is iterated over all subsets, so that the WRSS for each subset is determined. The average WRSS corresponds to the cross-validation error and is given by

$$\text{CV-error} = \frac{1}{M} \sum_{m=1}^{M} \text{WRSS}_m \ . \tag{2.15}$$

Accordingly, the standard error (SE) of the cross-validation error is given by

$$\text{SE-CV-error} = \sqrt{\frac{\sigma^2}{M}} \ , \tag{2.16}$$

where σ is the variance of the cross-validation error.

A cross-validation using M subsets is denoted as M-fold cross-validation.

By calculating the cross-validation error for different values of λ, it can be determined which value yields the most general valid models.

Following Friedman et al. [26], a reasonable sequence of values for λ can be determined by first calculating the lowest possible value $\lambda_{\max}$, which still leaves all coefficients at zero. This value can be approximated with the update scheme (2.11) by demanding that the third case, which sets the coefficient to zero, should occur for all coefficients ($g_k \geq |f_k|$ for all k) and by using the assumption that all coefficients are initially zero. Therefore, all values for λ which fulfill

$$\lambda \alpha v_k \geq \left| \sum_{i=1}^{N} w_i x_{ik} (y_i - \beta_0 - \sum_{\substack{j=1 \\ j \neq k}}^{p} \beta_j x_{ij}) \right| \qquad \text{for all } k, \tag{2.17}$$

yield a model, which only consists of zeros. Hence, the lowest possible value which satisfies (2.17) is denoted as $\lambda_{\max}$ and is given by:

$$\lambda_{\max} \approx \max_{\forall k} \frac{\left| \sum_{i=1}^{N} w_i x_{ik} (y_i - \beta_0) \right|}{\alpha v_k} \ . \tag{2.18}$$

Since (2.18) is undefined for the ridge regularization ($\alpha = 0$), this case has to be avoided by temporary assigning a small value (typically 0.01) to α. Thereby, a sufficiently good approximation for $\lambda_{\max}$ can be obtained [30]. Note that this approximation depends on β_0, which thus has to be calculated in advance by using the offset update formula (2.14).

The parameter $\lambda_{\max}$ then serves as an upper bound for the λ-sequence where all coefficients remain zero. A rule of thumb for the lower bound λ_{low} of the λ sequence is

$$\lambda_{\text{low}} = 0.001 \cdot \lambda_{\max} \ . \tag{2.19}$$

Constantly lowering the value of λ allows more and more coefficients to be non-zero. Between $\lambda_{\max}$ and λ_{low} typically 100 intermediate values, which are distributed logarithmically declining, are evaluated. For each value the resulting model is tested in cross-validation and the optimal λ is chosen according to the smallest cross-validation error. The common procedure is to begin with $\lambda_{\max}$ and to use the solution as a *warm start* for the next lower value. This procedure not only reduces the computing effort but also allows for early stopping if the cross-validation error significantly raises again for lower values [26].
An exemplary approximation of an optimal λ for a linear lasso ($\alpha = 1$) regression is shown in figure 2.1. As one can see, the cross-validation error (black dots) is high for high values of λ and drops for lower values until it reaches a minimum (left dotted vertical line). For even smaller values the cross-validation error raises again, which indicates that the model looses generalizability and overfitting starts to occur. The error bars correspond to the standard error of the cross-validation error.

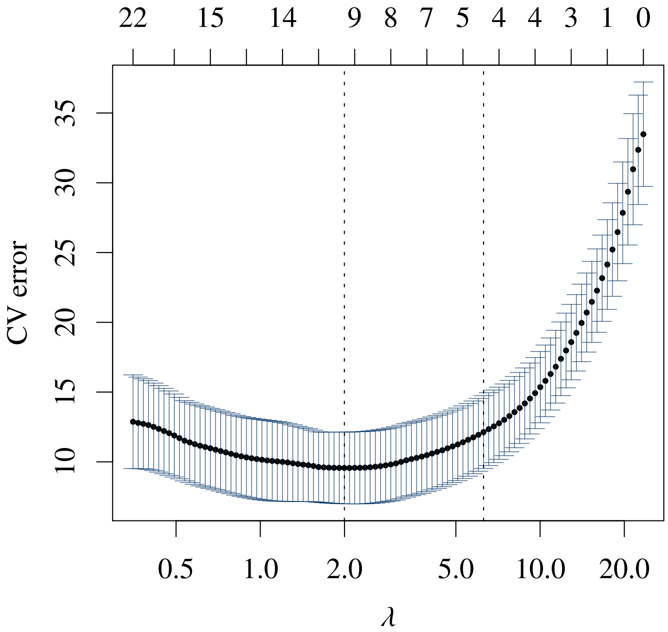


Figure 2.1: This figure shows the cross-validation (CV) error of a linear lasso ($\alpha = 1$) regression as a function of λ. The left dotted vertical line indicates the minimum of the CV error and the right dotted vertical lines depicts the $\lambda_{1\text{SE}}$. At the top, the number of non-zero coefficients is shown.

The optimal value for λ is the one which minimizes the cross-validation error (left dotted vertical line) and is called $\lambda_{\min}$. Another occasionally used choice is $\lambda_{1\text{SE}}$ (right dotted vertical line), which is the higher value of λ where the cross-validation error (almost) equals the minimal cross-validation error plus one standard error of the minimum. The ratio behind this choice is to obtain a more sparse model, that is roughly as accurate as the model obtained with $\lambda_{\min}$ [48].
At the top axis the number of non-zero coefficients of the model is shown. It can be seen that, the lower the value of λ, the higher the number of non-zero coefficients.

2.7 Logistic Regression

Logistic regression is used for categorical responses and uses the logistic function for predicting the probabilities of different categories. A distinction is made between a binary response, which is approximated with logistic regression and a multi categorical response, which is estimated with multinomial regression. Even though logistic regression is a special case of multinomial regression, it is advisable to use logistic regression in the case of a binary response, since this regression problem can be solved more efficiently. Note that for a binary response modeled by $y = 1$ for the first class ($G = 1$) and $y = 0$ for the second class ($G = 2$), it is only necessary to approximate the probability $\Pr(G = 1|\boldsymbol{x}_i)$ of one class, since the probability of the other class $\Pr(G = 2|\boldsymbol{x}_i)$ is given by $1 - \Pr(G = 1|\boldsymbol{x}_i)$.
To approximate $\Pr(G = 1|\boldsymbol{x}_i)$ the logistic function is used as a link function [26, 32]:

$$\Pr(G = 1|\boldsymbol{x}_i) = \frac{1}{1 + e^{-(\beta_0 + \boldsymbol{x}_i^T \boldsymbol{\beta})}}, \tag{2.20}$$

$$\Pr(G = 2|\boldsymbol{x}_i) = \frac{1}{1 + e^{(\beta_0 + \boldsymbol{x}_i^T \boldsymbol{\beta})}} = 1 - \Pr(G = 1|\boldsymbol{x}_i). \tag{2.21}$$

In the following the abbreviation $p(\boldsymbol{x}_i) = \Pr(G = 1|\boldsymbol{x}_i)$ is used.
For logistic regression the parameters $(\beta_0, \boldsymbol{\beta})$ need to be chosen so that the following likelihood function l is maximized [32]:

$$l(\beta_0, \boldsymbol{\beta}) = \prod_{i=1}^{N} \Pr(G_i|\boldsymbol{x}_i)^{w_i} \tag{2.22}$$

$$= \prod_{i=1}^{N} p(\boldsymbol{x}_i)^{w_i y_i} \cdot (1 - p(\boldsymbol{x}_i))^{w_i(1-y_i)}, \tag{2.23}$$

where w_i corresponds to the observation weights of the weighted residual sum of squares defined in (2.8).
Since it is easier to perform calculations on the logarithm of the likelihood, the log-likelihood function $\mathcal{L}$ is used [26, 32]:

$$\mathcal{L}(\beta_0, \boldsymbol{\beta}) = \log\Big(l(\beta_0, \boldsymbol{\beta})\Big) \tag{2.24}$$

$$= \sum_{i=1}^{N} w_i \Big\{ y_i \log p(\boldsymbol{x}_i) + (1 - y_i) \log(1 - p(\boldsymbol{x}_i)\Big\} \tag{2.25}$$

$$= \sum_{i=1}^{N} w_i \Big\{ y_i(\beta_0 + \boldsymbol{x}_i^T \boldsymbol{\beta}) - \log(1 + e^{\beta_0 + \boldsymbol{x}_i^T \boldsymbol{\beta}})\Big\}. \tag{2.26}$$

The intermediate steps from (2.25) to (2.26) are detailed in the appendix A.2.
The cost function of the logistic regression optimization problem combined with the elastic net regularization is thus

$$\min_{(\beta_0, \boldsymbol{\beta}) \in \mathbb{R}^{p+1}} \mathcal{H}_\lambda(\beta_0, \boldsymbol{\beta}) = \min_{(\beta_0, \boldsymbol{\beta}) \in \mathbb{R}^{p+1}} \Bigg[- \mathcal{L}(\beta_0, \boldsymbol{\beta}) + \lambda P_{\text{elastic net}}(\boldsymbol{\beta})_\alpha \Bigg] \tag{2.27}$$

$$= \min_{(\beta_0, \boldsymbol{\beta}) \in \mathbb{R}^{p+1}} \Bigg[- \sum_{i=1}^{N} w_i \Big\{ y_i(\beta_0 + \boldsymbol{x}_i^T \boldsymbol{\beta}) - \log(1 + e^{\beta_0 + \boldsymbol{x}_i^T \boldsymbol{\beta}})\Big\} + \lambda \sum_{j=1}^{p} v_j \Big(\frac{1 - \alpha}{2} \beta_j^2 + \alpha |\beta_j| \Big) \Bigg]. \tag{2.28}$$

Note that the global minimum of this function and not the maximum corresponds to the best solution. For that reason, the negative value of the log-likelihood has to be used, since in contrast to the residual sum of squares a higher value implies a more accurate model.
In contrast to the ordinary linear regression, the partial derivative of the log-likelihood function (2.26) can no longer be solved analytically for β_0 and β_k. For this reason, the cost function cannot be directly optimized with a coordinate descent algorithm. However, by using the local approximation of the cost function (2.28), it is still possible to solve the problem analogously to the linear regression case. Therefore, the second order multidimensional Taylor expansion $\mathrm{Tf}_{\boldsymbol{a}}(\boldsymbol{x})$ of a function $f(\boldsymbol{x})$ centered at $\boldsymbol{a}$ can be transformed to be of an equivalent form as the weighted residual sum of squares (2.8) [26, 70]:

$$\mathrm{Tf}_{\boldsymbol{a}}(\boldsymbol{x}) = -\frac{1}{2}\sum_i \tilde{w}_i(\boldsymbol{a})\Big(\tilde{z}_i(\boldsymbol{a}) - x\Big)^2 + C(\boldsymbol{a})\,, \tag{2.29}$$

with

$$\tilde{w}_i(\boldsymbol{a}) = -\frac{\partial^2 f}{\partial x_i^2}(\boldsymbol{a})\,, \tag{2.30}$$

$$\tilde{z}_i(\boldsymbol{a}) = a_i - \left(\frac{\partial^2 f}{\partial x_i^2}(\boldsymbol{a})\right)^{-1}\frac{\partial f}{\partial x_i}(\boldsymbol{a})\,. \tag{2.31}$$

The derivation is detailed in the appendix A.3. $C(\boldsymbol{a})$ only depends on $\boldsymbol{a}$ and not on $\boldsymbol{x}$ and thus vanishes in the partial derivatives.
Applying this approximation to the logistic regression log-likelihood (2.26) results in the approximated log-likelihood $\mathcal{L}^*$

$$\mathcal{L}^*(\beta_0,\boldsymbol{\beta}) = -\frac{1}{2}\sum_{i=1}^{N}\tilde{w}_i\left(\tilde{z}_i - \beta_0 - \boldsymbol{x}_i^T\boldsymbol{\beta}\right)^2 + C(\tilde{\beta}_0,\tilde{\boldsymbol{\beta}})\,, \tag{2.32}$$

with

$$\tilde{w}_i = w_i\tilde{p}(\boldsymbol{x}_i)(1-\tilde{p}(\boldsymbol{x}_i))\,, \tag{2.33}$$

$$\tilde{z}_i = \beta_0 + \boldsymbol{x}_i^T\tilde{\boldsymbol{\beta}} + \frac{y_i - \tilde{p}(\boldsymbol{x}_i)}{\tilde{p}(\boldsymbol{x}_i)(1-\tilde{p}(\boldsymbol{x}_i))}\,, \tag{2.34}$$

where $(\tilde{\beta}_0,\tilde{\boldsymbol{\beta}})$ is the current set of coefficients at which the approximation is centered and $\tilde{p}(\boldsymbol{x}_i)$ denotes the probability (2.20) evaluated at $(\tilde{\beta}_0,\tilde{\boldsymbol{\beta}})$:

$$\tilde{p}(\boldsymbol{x}_i) = \frac{1}{1+e^{-(\tilde{\beta}_0+\boldsymbol{x}_i^T\tilde{\boldsymbol{\beta}})}}\,. \tag{2.35}$$

This quadratic approximation combined with the weighted elastic net regularization yields the approximated cost function $\mathcal{H}^*$ and the corresponding optimization problem:

$$\min_{(\beta_0,\boldsymbol{\beta})\in\mathbb{R}^{p+1}}\mathcal{H}^*_\lambda(\beta_0,\boldsymbol{\beta}) = \min_{(\beta_0,\boldsymbol{\beta})\in\mathbb{R}^{p+1}}\Big[-\mathcal{L}^*(\beta_0,\boldsymbol{\beta}) + \lambda P_{\text{elastic net}}(\boldsymbol{\beta})_\alpha\Big] \tag{2.36}$$

$$= \min_{(\beta_0,\boldsymbol{\beta})\in\mathbb{R}^{p+1}}\left[\frac{1}{2}\sum_{i=1}^{N}\tilde{w}_i\Big(\tilde{z}_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\Big)^2 - C(\tilde{\beta}_0,\tilde{\boldsymbol{\beta}}) + \lambda\sum_{j=1}^{p}v_j\Big(\frac{1-\alpha}{2}\beta_j^2 + \alpha|\beta_j|\Big)\right]. \tag{2.37}$$

This approximated cost function is of the same form as the cost function of the linear regression (2.10). Hence, the coordinate descent update scheme (2.11) of the linear regression can also be used for logistic regression. The only difference is that the parameters $\tilde{w}_i$, $\tilde{z}_i$ of the local approximation have to be recalculated after each successful coordinate descent update.
In order to calculate $\lambda_{\max}$, it is necessary to determine an *intercept only model.* This model can be determined without the approximation by calculating the partial derivative of (2.26) and solving for β_0 using $\boldsymbol{\beta} = \vec{0}$. This results in the following equation for the *intercept only model*:

$$\hat{\beta}_0 = -\log\left(\frac{\sum_{i=1}^{N} w_i}{\sum_{i=1}^{N} w_i y_i} - 1\right). \tag{2.38}$$

In conclusion, $\lambda_{\max}$ can be estimated as described above and a λ sequence can be constructed.

2.8 Multinomial Regression

As shown by Zhu and Hastie [89] the approach using the quadratic approximation of the cost function can also be applied to the multinomial case. Therefore, the probabilities of a multi-categorical response G with K cases have to be modeled as

$$p_h(\boldsymbol{x}_i) = \frac{e^{\beta_{0h} + \boldsymbol{x}_i^T \boldsymbol{\beta}_h}}{\sum_{k=1}^{K} e^{\beta_{0k} + \boldsymbol{x}_i^T \boldsymbol{\beta}_k}} \quad \text{, with} \quad h = 1, \dots, K\,, \tag{2.39}$$

where $p_h(\boldsymbol{x}_i) = \Pr(G = h|\boldsymbol{x}_i)$. Each category is approximated by an individual set of coefficients $\{\beta_{0h}, \boldsymbol{\beta}_h\}$ where $\boldsymbol{\beta}_h$ is a vector of length p.
For the application of a coordinate descent algorithm the response has to be transformed into a $N \times K$ indicator response matrix $\boldsymbol{y}$, where each column corresponds to a category [26]. For example:

$$\boldsymbol{y} = \begin{array}{r} \\ \text{sample 1} \\ \text{sample 2} \\ \text{sample 3} \\ \vdots \end{array} \begin{array}{c} \begin{array}{cccc} \text{cat. 1} & \text{cat. 2} & \text{cat. 3} & \text{cat. 4} \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ & & & \end{pmatrix} \end{array} . \tag{2.40}$$

Therefore, the likelihood function l is

$$l(\{\beta_{0h}, \boldsymbol{\beta}_h\}_1^K) = \prod_{i=1}^{N} \prod_{l=1}^{K} p_l(\boldsymbol{x}_i)^{w_i y_{il}}\,, \tag{2.41}$$

where w_i are sample weights. The commonly used logarithm of the likelihood function is [26]

$$\begin{align} \mathcal{L}(\{\beta_{0h}, \boldsymbol{\beta}_h\}_1^K) &= \log\left(l(\{\beta_{0h}, \boldsymbol{\beta}_h\}_1^K)\right) \tag{2.42} \\ &= \sum_{i=1}^{N} w_i \left[\sum_{l=1}^{K} y_{il}(\beta_{0l} + \boldsymbol{x}_i^T \boldsymbol{\beta}_l) - \log\left(\sum_{k=1}^{K} e^{\beta_{0k} + \boldsymbol{x}_i^T \boldsymbol{\beta}_k}\right)\right]. \tag{2.43} \end{align}$$

The intermediate steps are detailed in the appendix A.4.

Combining this log-likelihood function with the elastic net regularization yields the cost function $\mathcal{H}$ and the corresponding optimization problem:

$$\min_{(\beta_0,\boldsymbol{\beta})\in\mathbb{R}^{p+1}} \mathcal{H}_\lambda(\beta_0,\boldsymbol{\beta}) = \min_{(\beta_0,\boldsymbol{\beta})\in\mathbb{R}^{p+1}} \left[-\mathcal{L}(\{\beta_{0h},\boldsymbol{\beta}_h\}_1^K) + \lambda P_{\text{elastic net}}(\boldsymbol{\beta})_\alpha \right] \tag{2.44}$$

$$= \min_{(\beta_0,\boldsymbol{\beta})\in\mathbb{R}^{p+1}} \Bigg[-\sum_{i=1}^{N} w_i \Big[\sum_{l=1}^{K} y_{il}(\beta_{0l} + \boldsymbol{x}_i^T \boldsymbol{\beta}_l) - \log\Big(\sum_{k=1}^{K} e^{\beta_{0k}+\boldsymbol{x}_i^T\boldsymbol{\beta}_k} \Big) \Big] + \lambda \sum_{j=1}^{p} v_j \Big(\frac{1-\alpha}{2}\beta_j^2 + \alpha|\beta_j| \Big) \Bigg] . \tag{2.45}$$

Similar to the logistic case, the quadratic approximation $\mathcal{L}^*$ centered at $(\beta_{0l},\boldsymbol{\beta}_l)$ for one response type l can be used [26]:

$$\mathcal{L}^*(\{\beta_{0h},\boldsymbol{\beta}_h\}_1^K) = -\frac{1}{2}\sum_{i=1}^{N} \tilde{w}_{il}\Big(\tilde{z}_{il} - \beta_{0l} - \boldsymbol{x}_i^T\boldsymbol{\beta}_l\Big)^2 + C(\tilde{\beta}_{0l},\tilde{\boldsymbol{\beta}}_l)\,, \tag{2.46}$$

with

$$\tilde{w}_{il} = w_i \tilde{p}_l(x_i)(1-\tilde{p}_l(x_i))\,, \tag{2.47}$$

$$\tilde{z}_{il} = \tilde{\beta}_{0l} + \boldsymbol{x}_i^T\tilde{\boldsymbol{\beta}}_{\boldsymbol{l}} + \frac{y_{il} - \tilde{p}_l(x_i)}{p_l(x_i)(1-\tilde{p}_l(x_i))}\,. \tag{2.48}$$

The derivation is detailed in the appendix A.5.
This approximation combined with the extended elastic net regularization yields the following approximated cost function for one class l:

$$\min_{(\beta_{0l},\boldsymbol{\beta}_l)\in\mathbb{R}^{p+1}} \mathcal{H}_\lambda^*(\{\beta_{0h},\boldsymbol{\beta}_h\}_1^K) = \min_{(\beta_{0l},\boldsymbol{\beta}_l)\in\mathbb{R}^{p+1}} \Bigg[\frac{1}{2}\sum_{i=1}^{N} \tilde{w}_{il}\Big(\tilde{z}_{il} - \beta_{0l} - \boldsymbol{x}_i^T\boldsymbol{\beta}_l\Big)^2 - C(\tilde{\beta}_{0l},\tilde{\boldsymbol{\beta}}_l) + \lambda \sum_{l=1}^{K}\sum_{j=1}^{p} v_j\Big(\frac{1-\alpha}{2}\beta_{jl}^2 + \alpha|\beta_{jl}|\Big) \Bigg] . \tag{2.49}$$

The coordinate descent algorithm for solving the multinomial case has to be extended to update all features of all classes β_{kl} by using the following extended update scheme:

$$\hat{\beta}_{kl} \leftarrow \frac{1}{\sum_{i=1}^N \tilde{w}_{il}x_{ik}^2 + \lambda(1-\alpha)v_k} \cdot \begin{cases} \Big(\sum_{i=1}^N \tilde{w}_{il}x_{ik}\Big(\tilde{z}_{il} - \beta_{0l} - \sum\limits_{\substack{j=1\\ j\neq k}}^{p} x_{ij}\beta_{jl}\Big) + \lambda\alpha v_k \Big) & \text{if } f_{kl} < 0 \wedge g_{kl} < |f_{kl}| \\ \Big(\sum_{i=1}^N \tilde{w}_{il}x_{ik}\Big(\tilde{z}_{il} - \beta_{0l} - \sum\limits_{\substack{j=1\\ j\neq k}}^{p} x_{ij}\beta_{jl}\Big) - \lambda\alpha v_k \Big) & \text{if } f_{kl} > 0 \wedge g_{kl} < |f_{kl}| \\ 0 & \text{if } g_{kl} \geq |f_{kl}| \end{cases} , \tag{2.50}$$

with

$$f_{kl} := \sum_{i=1}^{N} \tilde{w}_{il}x_{ik}\Big(\tilde{z}_{il} - \beta_{0l} - \sum_{\substack{j=1\\ j\neq k}}^{p} x_{ij}\beta_{jl}\Big)\,, \tag{2.51}$$

$$g_{kl} := \lambda\alpha v_k\,. \tag{2.52}$$

After each successful coordinate descent step the parameters $\tilde{w}_{il}$, $\tilde{z}_{il}$ of the local approximation have to be updated.
Analogously, the intercept update scheme (2.14) for the linear regression case has the following form in the multinomial case:

$$\hat{\beta}_{0l} = \frac{\sum_{i=1}^{N} \tilde{w}_i \big(\tilde{z}_i - \sum_{j=1}^{p} \beta_{jl} x_{ij}\big)}{\sum_{i=1}^{N} \tilde{w}_i}\,. \tag{2.53}$$

In order to determine $\lambda_{\max}$ it is again necessary to calculate an *intercept only model*. This can be achieved by calculating the partial derivative of the actual log-likelihood (2.43) with respect to β_{0m} and the assumption that the other intercepts are held fixed. The initial condition $\boldsymbol{\beta} = \vec{0}$ allows to solve the partial derivative for the optimal value $\hat{\beta}_{0m}$:

$$\frac{\partial \mathcal{L}(\{\beta_{0h}\}_1^K)}{\partial \beta_{0m}} = \sum_{i=1}^{N} w_i y_{im} - \frac{e^{\beta_{0m}}}{\sum_{k=1}^{K} e^{\beta_{0k}}} \sum_{i=1}^{N} w_i \overset{!}{=} 0\,. \tag{2.54}$$

Therefore, the optimal value is given by

$$\hat{\beta}_{0m} = -\log\left[\Big(\frac{\sum_{i=1}^{N} w_i}{\sum_{i=1}^{N} w_i y_{im}} - 1\Big)\Big(\sum_{\substack{k=1\\k\neq m}}^{K} e^{\beta_{0k}}\Big)^{-1}\right]. \tag{2.55}$$

This update scheme has to be iterated over all classes K until no further changes occur. Afterwards, a $\lambda_{\max}$ approximation can be used, which is analogous to the logistic regression, except that all coefficients of all classes have to be considered. Thus, $\lambda_{\max}$ is given by:

$$\lambda_{\max} \approx \frac{\max\limits_{\forall k,l} \left|\sum_{i=1}^{N} \tilde{w}_{il} x_{ik} (\tilde{z}_{il} - \beta_{0l})\right|}{\alpha v_k}\,. \tag{2.56}$$

One important aspect of the multinomial probabilities, which is referred to as parameter ambiguity problem in [26], is that they are invariant under beta-shifts. This can be seen by considering beta shifts which are simultaneously applied on all coefficients sets l and are of the form $\beta'_{0l} = \beta_{0l} + \delta_0$ and $\boldsymbol{\beta}'_l = \boldsymbol{\beta}_l + \boldsymbol{\delta}$, where δ_0 is a scalar and $\boldsymbol{\delta}$ is a vector of length p. Such shifts do not change the multinomial probability (2.39):

$$\begin{align} p_l(\boldsymbol{x}_i) &= \frac{\exp(\beta'_{0l} + \boldsymbol{x}_i^T \boldsymbol{\beta}'_l)}{\sum_{k=1}^{K} \exp(\beta'_{0k} + \boldsymbol{x}_i^T \boldsymbol{\beta}'_k)} \tag{2.57}\\ &= \frac{\exp(\beta_{0l} + \boldsymbol{x}_i^T \boldsymbol{\beta}_l + \delta_0 + \boldsymbol{x}_i^T \boldsymbol{\delta})}{\sum_{k=1}^{K} \exp(\beta_{0k} + \boldsymbol{x}_i^T \boldsymbol{\beta}_k + \delta_0 + \boldsymbol{x}_i^T \boldsymbol{\delta})} \tag{2.58}\\ &= \frac{\exp(\beta_{0l} + \boldsymbol{x}_i^T \boldsymbol{\beta}_l) \cdot \exp(\delta_0 + \boldsymbol{x}_i^T \boldsymbol{\delta})}{\exp(\delta_0 + \boldsymbol{x}_i^T \boldsymbol{\delta}) \cdot \Big(\sum_{k=1}^{K} \exp(\beta_{0k} + \boldsymbol{x}_i^T \boldsymbol{\beta}_k)\Big)} \tag{2.59}\\ &= \frac{\exp(\beta_{0l} + \boldsymbol{x}_i^T \boldsymbol{\beta}_l)}{\sum_{k=1}^{K} \exp(\beta_{0k} + \boldsymbol{x}_i^T \boldsymbol{\beta}_k)}\,. \tag{2.60} \end{align}$$

Thus, the likelihood (2.41) and log-likelihood (2.43) are also invariant under those shifts. However, this is not the case for the elastic net regularization. As a consequence, it is possible to determine shifts which minimize the cost function [26]. Therefore, a reduced cost function R can be used to identify a shift which minimizes the elastic-net component in (2.45). The reduced cost function R is

$$\min_{\boldsymbol{\delta}\in\mathbb{R}^p} \mathrm{R}(\boldsymbol{\delta}) = \min_{\boldsymbol{\delta}\in\mathbb{R}^p} \sum_{l=1}^{K} \sum_{j=1}^{p} v_j \Big(\frac{1-\alpha}{2} (\beta_{jl} - \delta_j)^2 + \alpha |\beta_{jl} - \delta_j| \Big) . \tag{2.61}$$

This thesis follows the approach proposed by Friedman et al. [26], which is also outlined in the appendix A.6 and which yields the following update scheme

$$\delta_k \leftarrow \frac{1}{K} \sum_{l=1}^{K} \beta_{kl} - \frac{1}{K} \frac{\alpha}{1-\alpha} \sum_{l=1}^{K} \begin{cases} -1 & \text{if } \beta_{kl} - \delta_k > 0 \\ 1 & \text{if } \beta_{kl} - \delta_k < 0 \ . \\ \text{else not defined} & \end{cases} \tag{2.62}$$

One crucial property of this solution is that the value of δ_k also occurs in the case differentiation and therefore influences itself. Friedman et al. [26] proposed a solution for this problem based on the proof that the optimal value for δ_k has to be between the mean and the median of the coefficients β_{jl} with $l = 1, \dots, K$. Thus a simple bisecting search algorithm can be used in this interval to determine the optimal value for δ_k. In the case of $v_j = 0$ mean centering is performed. Since the computing time required by this step is negligible in comparison to the coordinate descent algorithm, this approach is sufficient.

This solution for the parameter ambiguity problem can be incorporated in the coordinate descent structure described in 2.5 as an additional step between 3. and 4. and updates all coefficients with (2.62).

2.9 Cox Proportional Hazard Regression

Another type of generalized linear models proposed by Cox [15] are proportional hazards models also named Cox models. These models are used to estimate the relation of input data $\boldsymbol{x}$ and the time $\boldsymbol{y}$ until a specific event occurs and are thus often applied for analyzing survival data. In the following the name Cox regression will be used.

One important aspect of survival studies is that some study participants drop out of a study without the occurrence of an event or where the event luckily did not occur. Thus, the resulting information is only that the event has not occurred in the observation period. To also make use of such data, right-censoring is applied by adding an additional response variable δ_i, which encodes if an event has occurred at y_i ($\delta_i = 1$) or if the data is censored at y_i ($\delta_i = 0$).

For setting up a likelihood function the samples with events have to be sorted in ascending order so that: $t_1 < t_2 < \cdots < t_M$, where M is the number of samples with events. The case with ties is described subsequently. The hazard for an event of an observation i at time t can then be modeled with a hazard function $h_i(t)$

$$h_i(t) = h_0(t) \cdot e^{\boldsymbol{x}_i^T \boldsymbol{\beta}} , \tag{2.63}$$

where $h_0(t)$ is a shared baseline hazard. Note that no intercept is needed because of this baseline hazard. The hazard function can be used to construct a partial likelihood function, which only depends on the succession of events:

$$l(\boldsymbol{\beta}) = \prod_{i=1}^{M} \frac{e^{\boldsymbol{x}_i^T \boldsymbol{\beta}}}{\sum_{j\in R_i} e^{\boldsymbol{x}_j^T \boldsymbol{\beta}}} . \tag{2.64}$$

R_i denotes the set of samples where no event has occurred until y_i ($y_j \geq y_i$ for all elements y_j of R_i), which are therefore still at risk at time y_i. Note that the product ranges over all samples M with events and that R_i in the denominator also includes the censored samples.

In order to also allow for ties, one approach is to use the Breslow approximation [9, 70], which incorporates sample-wise weights w_i. Therefore, all events at a time t are assembled together to a set D_t. As in the case with no ties, these sets are sorted in ascending order: $D_1 < D_2 < \cdots < D_M$, where M now denotes the number of sets with unique event times. The extended partial likelihood is then

$$l(\boldsymbol{\beta}) = \prod_{i=1}^{M} \frac{e^{\sum_{j\in D_i} w_j(\boldsymbol{x}_j^T\boldsymbol{\beta})}}{\left(\sum_{j\in R_i} w_j e^{\boldsymbol{x}_j^T\boldsymbol{\beta}}\right)^{d_i}}, \tag{2.65}$$

where d_i is defined as $d_i = \sum_{j\in D_i} w_j$.

Thus, the log partial likelihood is

$$\mathcal{L}(\boldsymbol{\beta}) = \log(l(\boldsymbol{\beta})) \tag{2.66}$$

$$= \sum_{i=1}^{M} \left(\sum_{j\in D_i} w_j(\boldsymbol{x}_j^T\boldsymbol{\beta}) - d_i \log\Big(\sum_{j\in R_i} w_j e^{\boldsymbol{x}_j^T\boldsymbol{\beta}} \Big) \right). \tag{2.67}$$

Simon et al. [70] proposed to use the local approximation (2.29) of this partial log-likelihood in order to solve the Cox regression with a coordinate descent algorithm. The approximated log-likelihood $\mathcal{L}^*$ centered at $\tilde{\boldsymbol{\beta}}$ is given by:

$$\mathcal{L}^*(\boldsymbol{\beta}) = -\frac{1}{2} \sum_{i=1}^{N} \tilde{w}_i \left(\tilde{z}_i - \boldsymbol{x}_i^T\boldsymbol{\beta}\right)^2 + C(\tilde{\boldsymbol{\beta}}), \tag{2.68}$$

with the parameters

$$\tilde{w}_i = \sum_{k\in C_i} d_k \frac{w_i e^{\boldsymbol{x}_i^T\boldsymbol{\beta}} \sum_{j\in R_k} w_j e^{\boldsymbol{x}_j^T\boldsymbol{\beta}} - w_i^2 e^{2\boldsymbol{x}_i^T\boldsymbol{\beta}}}{\left(\sum_{j\in R_k} w_j e^{\boldsymbol{x}_j^T\boldsymbol{\beta}}\right)^2}, \tag{2.69}$$

$$\tilde{z}_i = \boldsymbol{x}_i^T\tilde{\boldsymbol{\beta}} + \frac{1}{\tilde{w}_i}\left[w_i\delta_i - \sum_{k\in C_i} \frac{d_k w_i e^{\boldsymbol{x}_i^T\boldsymbol{\beta}}}{\sum_{j\in R_k} w_j e^{\boldsymbol{x}_j^T\boldsymbol{\beta}}} \right]. \tag{2.70}$$

C_i denotes the set of sets D_j where the observation time y_j is less than or equal to the observation time of the set D_i ($y_j \leq y_i$ for all elements y_j of C_i). Note that this C_i is not the same as $C(\tilde{\boldsymbol{\beta}})$ in (2.68). The derivation of this approximation is again detailed in the appendix A.7.

In conclusion, this problem can be solved with a coordinate descent algorithm with the update scheme defined by (2.11).

For estimating $\lambda_{\max}$ the assumption $\boldsymbol{\beta_j} = \vec{0}$ can be used to simplify the calculation of $\tilde{w}_i$ and $\tilde{z}_i$ as follows:

$$\tilde{w}_i = \sum_{k\in C_i} d_k \frac{w_i \sum_{j\in R_k} w_j - w_i^2}{\left(\sum_{j\in R_k} w_j\right)^2}, \tag{2.71}$$

$$\tilde{z}_i = \frac{1}{\tilde{w}_i}\left[w_i\delta_i - \sum_{k\in C_i} \frac{d_k w_i}{\sum_{j\in R_k} w_j} \right]. \tag{2.72}$$

Since there is no intercept, $w_i \cdot y_i$ in (2.18) is given by $\tilde{w}_i \cdot \tilde{z}_i$:

$$\tilde{w}_i \cdot \tilde{z}_i = w_i \delta_i - w_i \sum_{k \in C_i} \frac{d_k}{\sum_{j \in R_k} w_j} . \tag{2.73}$$

Thus, $\lambda_{\max}$ is obtained by

$$\lambda_{\max} \approx \max_{\forall k} \frac{\left| \sum_{i=1}^{N} x_{ik} \tilde{w}_i \tilde{z}_i \right|}{\alpha v_k} . \tag{2.74}$$

2.10 Fused and Generalized Lasso

This section is about another regularization, which is called fused lasso and was proposed by Tibshirani et al. [78]. The aim of this regularization is to use additional knowledge about the ordering of features to improve linear models. NMR or mass spectra, for example, are typically subdivided into small regions called *bins* and the spectral integrals of these *bins* are used to construct a predictor vector $\boldsymbol{x}_i$. Since peaks in the spectrum can be broader than the width of the *bins* and since peaks can also be slightly shifted between measurements, neighboring *bins* can carry the same information. Thus, the combination of features, which corresponds to locally wider *bins*, can be used to improve the accuracy of a linear model [85]. This can be achieved by using the fused lasso regularization:

$$P_{\text{fused lasso}}(\boldsymbol{\beta}) = \sum_{j=1}^{p-1} \left| \beta_j - \beta_{j+1} \right| . \tag{2.75}$$

The fused lasso is limited to neighboring features and therefore Tibshirani and Taylor [79] proposed a more flexible version of the fused lasso called generalized lasso which is defined as follows:

$$P_{\text{generalized-lasso}}(\boldsymbol{\beta}) = ||\boldsymbol{F}\boldsymbol{\beta}||_1 , \tag{2.76}$$

where F is a $m \times p$ matrix. The number of rows m can be selected freely and allows to incorporate more prior knowledge into the regression to generate more accurate models. A specific protein or metabolite, for instance, can cause two separate peaks in a spectrum and thus it makes sense to couple the coefficients of these wide apart regions of the spectrum.

This regularization can be incorporated into the cost function (2.10) with an additional weight parameter γ:

$$\begin{aligned} \min_{(\beta_0, \boldsymbol{\beta}) \in \mathbb{R}^{p+1}} \mathcal{H}_\lambda(\beta_0, \boldsymbol{\beta}) = \min_{(\beta_0, \boldsymbol{\beta}) \in \mathbb{R}^{p+1}} \Bigg[& \frac{1}{2} \sum_{i=1}^{N} w_i \Big(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \Big)^2 \\ & + \lambda \sum_{j=1}^{p} v_j \Big(\frac{1-\alpha}{2} \beta_j^2 + \alpha |\beta_j| \Big) + \gamma ||\boldsymbol{F}\boldsymbol{\beta}||_1 \Bigg] . \end{aligned} \tag{2.77}$$

Since $\boldsymbol{F}$ can get quite large, it is required to be sparse for large data sets in order to solve this problem with computational methods. Moreover, this optimization problem can not be solved with a coordinate descent algorithm, since the generalized lasso consists of m absolute value functions which causes that the partial derivative with respect to β_k exhibits 3^m different cases. Nevertheless, there are approaches to construct coordinate descent algorithms if the matrix F is of special sparse structures [79]. However, to be usable without any limitation to F (except sparsity) general purpose optimization algorithms can be applied. Therefore, a local search is used in this thesis.

2.11 Data Standardization

One preprocessing step which is often recommended for elastic net regularized regression is data standardization, as the elastic net is influenced by the scale of the predictor variables. Therefore, features which are on a different scale are differently penalized [32]. If, for instance, a feature s equals another feature k but on a tenths of the scale ($x_{is} = 0.1 \cdot x_{ik}$ for all i), then the elastic net would not select feature s, since it would require a 10 times higher coefficient for the same prediction.

Therefore, it is a frequent practice to perform regularized regression on standardized data and then to translate the fitted coefficients back to its original scale. Standardization consists of two steps. At first the mean $\bar{x}_j$ of each feature j over all samples is calculated and subtracted to center the data. Afterwards, the standard deviation s_j of each feature is determined and used to scale the standard deviation of each feature to one. The standardized data x^*_{ij} and standardized response y^*_i are given by:

$$x^*_{ij} = \frac{x_{ij} - \bar{x}_j}{s_j}, \qquad y^*_i = \frac{y_i - \bar{y}}{s_y}. \tag{2.78}$$

To also incorporate observation weights the weighted mean and the weighted standard deviation have to be used:

$$\bar{x}_j = \frac{1}{\sum_i w_i} \sum_{i=1}^{N} w_i x_{ij}, \qquad \bar{y} = \frac{1}{\sum_i w_i} \sum_{i=1}^{N} w_i y_i, \tag{2.79}$$

$$s_j = \sqrt{\frac{1}{\sum_i w_i} \sum_{i=1}^{N} w_i \big(x_{ij} - \bar{x}_j\big)^2}, \qquad s_y = \sqrt{\frac{1}{\sum_i w_i} \sum_{i=1}^{N} w_i \big(y_i - \bar{y}\big)^2}. \tag{2.80}$$

Standardizing the response is only possible in the normal linear case.

In order to make predictions for new samples using a linear model, that has been learned on standardized data, it is necessary to transform the coefficients back to the original scale:

$$y^*_i \sim \beta^*_0 + \sum_{j=1}^{p} \beta^*_j x^*_{ij} \tag{2.81}$$

$$\frac{y_i - \bar{y}}{s_y} \sim \beta^*_0 + \sum_{j=1}^{p} \beta^*_j \Big(\frac{x_{ij} - \bar{x}_j}{s_j}\Big) \tag{2.82}$$

$$y_i \sim \underbrace{\Big(\beta^*_0 - \sum_{j=1}^{p} \beta^*_j \frac{\bar{x}_j}{s_j}\Big) \cdot s_y + \bar{y}}_{\beta_0} + \sum_{j=1}^{p} \underbrace{\Big(\frac{\beta^*_j s_y}{s_j}\Big)}_{\beta_j} x_{ij}. \tag{2.83}$$

Thus, β_0 and β_j on the original scale can be obtained by

$$\beta_0 = \Big(\beta^*_0 - \sum_{j=1}^{p} \beta^*_j \frac{\bar{x}_j}{s_j}\Big) \cdot s_y + \bar{y}, \tag{2.84}$$

$$\beta_j = \frac{s_y}{s_j} \cdot \beta^*_j. \tag{2.85}$$

Standardization has no effect on a linear model without regularization, which implies that a back-transformed linear model learned on standardized data is identical to a model trained on the original data.

However, the effect of standardization on regularization can be directly seen in (2.85). In the aforementioned case, feature s would exhibit a ten times lower standard deviation and thus both coefficients would equally contribute to the regularization.

One additional effect of standardization is that the intercept is always zero for normal linear regression. This can be seen by rearranging (2.14) as follows:

$$\hat{\beta}_0 = \frac{\sum_{i=1}^{N} w_i y_i^*}{\sum_{i=1}^{N} w_i} - \frac{1}{\sum_{i=1}^{N} w_i} \sum_{j=1}^{p} \beta_j \sum_{i=1}^{N} w_i x_{ij}^* . \tag{2.86}$$

The first term is zero because it corresponds to the weighted mean, which is zero due the weighted mean centering The second term is also zero, since $\sum_{i=1}^{N} w_i x_{ij}^*$ is zero for all predictor variables j. For that reason, the mean centering component of standardization simplifies calculations. Hence, it is advisable to perform mean-centering and back-transformation even if no standardization is requested. Mean-centering alone has no influence on the results as long as the standard deviation is not changed. However, this is only valid for normal linear regression. For logistic, multinomial and Cox regression the weights w_i also correspond to the quadratic approximation and have to be adjusted after each coordinate descent update. Thus the intercept is non-zero.

3 Zero-Sum Regression

Omics data is prone to normalization and scaling errors which result in data displacements along a specific feature space direction. The common strategy is to resolve these errors with normalization strategies, but this is only possible to a limited extent. Therefore, zero-sum regression is based on the opposite concept: instead of correcting for these displacements and using traditional data analysis methods, scale invariant methods can be directly applied. The aim of this thesis is to exemplify this concept by developing a scale invariant generalized linear regression framework and to provide a publicly available reference implementation called zeroSum.
This chapter shows how scale invariance can be achieved by imposing the zero-sum constraint and outlines algorithmic solutions. At first, the zero-sum constraint is detailed and the corresponding optimization problem formulated. Afterwards, solutions for this optimization problem based on coordinate descent, simulated annealing and local search are proposed. At the end of this chapter, the convergence behavior of these algorithms is evaluated.

3.1 Zero-Sum Constraint

As outlined in section 1.3, *omics* data has to be log-transformed in order to be accessible by linear models, since such data ranges over several orders of magnitudes and exhibits a skewed distribution [19, 49, 80]. Therefore, sample-wise multiplicative alterations by a factor γ_i transform to additive shifts in the log-space:

$$x'_{ij} = \log(\gamma_i \cdot x_{ij}) = \log(\gamma_i) + \log(x_{ij})\,. \tag{3.1}$$

Consequently, the influence of sample-wise scaling γ_i on the prediction of a linear model depends on the sum of the coefficients:

$$y_i \sim \beta_0 + \sum_{j=1}^{p} \beta_j \cdot \log(\gamma_i x_{ij}) \tag{3.2}$$

$$\sim \beta_0 + \underbrace{\log(\gamma_i) \cdot \sum_{j=1}^{p} \beta_j}_{\text{effect of sample scaling}} + \sum_{j=1}^{p} \beta_j \cdot \log(x_{ij})\,. \tag{3.3}$$

By enforcing that the sum of all coefficients adds up to zero

$$\sum_{j=1}^{p} \beta_j \overset{!}{=} 0 \tag{3.4}$$

the linear model becomes invariant against scaling. This constraint is referred to in this thesis as zero-sum constraint and was first proposed by Lin et al. [53] for compositional data and suggested in [3] for the application on biological *omics* data to circumvent scaling and normalization.

The zero-sum constraint is applicable on the regression types described in the previous chapter. However, this constraint relaxes in the multinomial case to an *equal-sum* constraint, which implies that the sum of each coefficient set k for all categories K has to be equal:

$$\sum_j \beta_{jk} = c \qquad \text{for all } k \in \{1, K\}\,, \tag{3.5}$$

where c is a value which only has to equal for each coefficient set k.
This can be seen by looking at the probability of the category h of an sample i defined by (2.39) and by considering sample-wise shifts γ_i:

$$p_h(\boldsymbol{x}_i) = \frac{\exp\left(\beta_{0h} + \sum_j \beta_{jh} \cdot \log(\gamma_i x_{ij})\right)}{\sum_{k=1}^{K} \exp\left(\beta_{0k} + \sum_j \beta_{jk} \cdot \log(\gamma_i x_{ij})\right)} \tag{3.6}$$

$$= \frac{\exp\left(\beta_{0h} + \sum_j \beta_{jh} \cdot \log(x_{ij})\right) \cdot \exp\left(\log(\gamma_i) \cdot \sum_j \beta_{jh}\right)}{\sum_{k=1}^{K} \exp\left(\beta_{0k} + \sum_j \beta_{jk} \cdot \log(x_{ij})\right) \cdot \exp\left(\log(\gamma_i) \cdot \sum_j \beta_{jk}\right)}\,. \tag{3.7}$$

The factors in the nominator and denominator containing the shifts γ_i cancel each other if the sum $\sum_j \beta_{jk}$ is equal for all categories k.
Note that the solution of the parameter ambiguity problem, which has been detailed in section 2.8, is not affected by this *equal-sum* constraint. The reason for this is, that these shifts are applied equally on all sets of coefficients and are causing that the sum $\sum_j \beta_{jk}$ of each category k is equally changed. Therefore, it is reasonable to start with $c = 0$ which is adjusted after each parameter ambiguity optimization step.
To take this *equal-sum* constraint into account, the sum of coefficients should add up to c:

$$\sum_{j=1}^{p} \beta_j \overset{!}{=} c\,. \tag{3.8}$$

In the linear, logistic and Cox regression case c is set to zero.
This constraint can also be used for different purposes. For instance, by using $c = 1$ and the additional condition $\beta_j \geq 0$ a regression can be used to model the composition within a population which should add up to one. Furthermore, an approach based on this constraint with $c = 1$ and a coordinate descent approach has been used for detecting transition phases in financial markets [43]. However, such utilizations are beyond the scope of this thesis and are not further considered.
An additional extension to this constraint is necessary to allow for data standardization, since the back transformation (2.85) alters the coefficients and thus revokes the scale invariance. To tackle this issue the following extended zero-sum constraint can be used:

$$\sum_{j=1}^{p} u_j \beta_j = c\,. \tag{3.9}$$

The weights $\boldsymbol{u}$ with the condition $u_j \geq 0$ for all j allow to compensate for feature-wise transformations in order to preserve scale invariance. Therefore, the weights u_j have to be set to s_y / s_j:

$$\sum_{j=1}^{p} u_j \beta_j^* \overset{(2.85)}{=} \sum_{j=1}^{p} u_j \frac{s_j}{s_y} \beta_j = \sum_{j=1}^{p} \beta_j = c\,. \tag{3.10}$$

One problem of standardization in combination with the zero-sum constraint is that sample-wise shifts change the variance of the features. If $\boldsymbol{x}_j$ denotes the vector of length N containing all values of feature

j of all samples and $\boldsymbol{\gamma}$ is a vector of length N containing all sample-wise shifts γ_i, then the variance of the shifted feature $\boldsymbol{x}_j + \boldsymbol{\gamma}$ is given by:

$$\mathrm{Var}(\boldsymbol{x}_j + \boldsymbol{\gamma}) = \mathrm{Var}(\boldsymbol{x}_j) + \mathrm{Var}(\boldsymbol{\gamma}) + 2 \cdot \mathrm{Cov}(\boldsymbol{x}_j, \boldsymbol{\gamma})\,. \tag{3.11}$$

It can be seen that the variance of feature j is not only altered by the variance of the sample-wise shifts, but also by the covariance of itself with the sample-wise shifts. Hence, the variance of each feature is altered differently. This implies that two different normalization methods would lead to two different feature variances and a zero-sum regression with standardization would yield two different models. As a consequence, standardization acts contrary to the scale invariance caused by the zero-sum constraint.

Nevertheless, if technical limitations are causing sample-wise alterations and if the standard deviation of the data is additionally distorted, for instance, due to a varying feature sensitivity of the measurement platform, then the combination of standardization and zero-sum constraint can reduce the influence of both effects on the results.

To give a careful user of the *zeroSum* software the possibility to use standardization, the necessary transformations are implemented, but switched off by default.

Another aspect is that the zero-sum constraint does not protect against feature-wise scaling. This can be seen by taking feature-wise scaling δ_j in (3.1) into account:

$$x'_{ij} = \log(x_{ij} \cdot \gamma_i \cdot \delta_j) = \log(x_{ij}) + \log(\gamma_i) + \log(\delta_j)\,. \tag{3.12}$$

The prediction of a model is thus affected as follows:

$$y_i \sim \beta_0 + \sum_{j=1}^{p} \beta_j \cdot \log(\gamma_i x_{ij} \delta_j) \tag{3.13}$$

$$\sim \beta_0 + \log(\gamma_i) \cdot \sum_{j=1}^{p} \beta_j + \sum_{j=1}^{p} \beta_j \cdot \log(x_{ij}) + \underbrace{\sum_{j=1}^{p} \beta_j \cdot \log(\delta_j)}_{\text{effect of feature-wise scaling}} \quad . \tag{3.14}$$

Hence, the effects of feature-wise scaling are independent of $\boldsymbol{x}$.

Different *omics* measurement platforms can have a different sensitivity and the same probe measured with two different platforms may exhibit such features-wise shifts δ_j. Since these shifts can be specific for a platform, they result in an additional platform specific intercept. This property was used for transferring molecular signatures – which correspond to linear models – between different measurement platforms [4].

3.2 Coordinate Descent

As shown in the previous chapter linear, logistic, multinomial and Cox regression can be solved with a coordinate descent algorithm by using the update scheme (2.11). Thus, the starting point for developing an efficient coordinate descent algorithm for fitting zero-sum models is to set up an equivalent update scheme, which complies with the zero-sum constraint. Such an update has to solve the following cost function, which is a combination of the weighted residual sum of squares (2.8), the elastic net (2.7) and the extended zero-sum constraint (3.9):

$$\mathcal{H}_\lambda(\beta_0,\boldsymbol{\beta}) = \frac{1}{2}\sum_{i=1}^{N} w_i\Big(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\Big)^2 + \lambda\sum_{j=1}^{p} v_j\Big(\frac{1-\alpha}{2}\beta_j^2 + \alpha|\beta_j|\Big)$$

$$\text{subject to } \sum_{j=1}^{p} u_j\beta_j = c\,. \tag{3.15}$$

The zero-sum constraint (3.9) can be solved for a randomly selected coefficient β_s with $u_s > 0$:

$$\beta_s = \frac{c - \sum_{\substack{j=1\\ j\neq s}}^{p} u_j\beta_j}{u_s}\,. \tag{3.16}$$

By utilizing this equation to replace β_s in (3.15), the constraint can be incorporated into the cost function:

$$\begin{aligned}\mathcal{H}_\lambda(\beta_0,\boldsymbol{\beta}) = {} & \frac{1}{2}\sum_{i=1}^{N} w_i\Bigg(y_i - \beta_0 - \sum_{\substack{j=1\\ j\neq s}}^{p} x_{ij}\beta_j - \frac{x_{is}}{u_s}\Big(c - \sum_{\substack{j=1\\ j\neq s}}^{p} u_j\beta_j\Big)\Bigg)^2 + \lambda\sum_{\substack{j=1\\ j\neq s}}^{p} v_j\Bigg(\frac{1-\alpha}{2}\beta_j^2 + \alpha|\beta_j|\Bigg) \\ & + \frac{\lambda v_s(1-\alpha)}{2u_s^2}\Big(c - \sum_{\substack{j=1\\ j\neq s}}^{p} u_j\beta_j\Big)^2 + \lambda\alpha\frac{v_s}{u_s}\cdot\Big|\big(c - \sum_{\substack{j=1\\ j\neq s}}^{p} u_j\beta_j\big)\Big|\,.\end{aligned} \tag{3.17}$$

This allows to construct an update scheme, which maintains the zero-sum constraint. Therefore, the partial derivative of (3.17) with respect to β_k can be used to determine the local optimal value for β_k. This yields the following update scheme for β_k, which will be referred to in the following as *normal update*:

$$\hat{\beta}_k \leftarrow \frac{1}{a_{ks}}\cdot\begin{cases}\big(b_{ks} - \lambda\alpha(v_k - \frac{v_s u_k}{u_s})\big) & \text{if} \quad \hat{\beta}_k > 0 \ \wedge \ \hat{\beta}_s > 0 \\ \big(b_{ks} - \lambda\alpha(v_k + \frac{v_s u_k}{u_s})\big) & \text{if} \quad \hat{\beta}_k > 0 \ \wedge \ \hat{\beta}_s < 0 \\ \big(b_{ks} + \lambda\alpha(v_k + \frac{v_s u_k}{u_s})\big) & \text{if} \quad \hat{\beta}_k < 0 \ \wedge \ \hat{\beta}_s > 0 \\ \big(b_{ks} + \lambda\alpha(v_k - \frac{v_s u_k}{u_s})\big) & \text{if} \quad \hat{\beta}_k < 0 \ \wedge \ \hat{\beta}_s < 0 \\ \text{derivative not defined, skip update} \end{cases}, \tag{3.18}$$

with

$$a_{ks} = \sum_{i=1}^{N} w_i(-x_{ik} + \frac{x_{is}u_k}{u_s})^2 + \lambda v_k(1-\alpha) + \lambda(1-\alpha)\frac{v_s u_k^2}{u_s^2}\,, \tag{3.19}$$

$$b_{ks} = -\sum_{i=1}^{N} w_i(-x_{ik} + \frac{x_{is}u_k}{u_s})\Bigg(y_i - \beta_0 - \sum_{\substack{j=1\\ j\neq s,k}}^{p} x_{ij}\beta_j - \frac{x_{is}}{u_s}\Big(c - \sum_{\substack{j=1\\ j\neq s,k}}^{p} u_j\beta_j\Big)\Bigg) + \frac{\lambda v_s(1-\alpha)u_k}{u_s^2}\Big(c - \sum_{\substack{j=1\\ j\neq s,k}}^{p} u_j\beta_j\Big)\,. \tag{3.20}$$

The derivation is detailed in the appendix A.8. After using this update scheme β_s has to be adjusted with (3.16).

Note that in the case differentiation the resulting values $\hat{\beta}_k$ and $\hat{\beta}_s$ appear. For that reason, at first each case has to be evaluated and then verified whether the result is valid (fulfills the case differentiation). Otherwise the update is rejected.

The case that some specific coefficients j should not be part of the zero-sum constraint ($u_j = 0$), can be solved by using the non-zero-sum coordinate descent update described in the last chapter (section 2.11) for updating these coefficients.

A general issue of coordinate descent algorithms is that the optimal state is sometimes not accessible by coordinate wise steps. This occurs if no further improvement in the directions of the coordinate system defined by the coefficients $\boldsymbol{\beta}$ can be achieved or if the update scheme is not defined. Therefore, only an update scheme capable of changing multiple coordinates simultaneously is able to reach the optimal state. This problem will be referred to in this thesis as *inaccessible issue*.

In order to resolve this issue, an additional update scheme can be constructed which changes three coefficients β_n, β_m and β_s simultaneously. Therefore, a search space translation with t_1, t_2 and a rotation by an angle θ can be applied on two randomly selected coefficients β_n and β_m ($n \neq m$, $n \neq s$, $m \neq s$)

$$\begin{pmatrix} \beta'_n \\ \beta'_m \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} \beta_n - t_1 \\ \beta_m - t_2 \end{pmatrix} . \tag{3.21}$$

Applying this transformation on the search space defined by (3.17) yields a cost function which is detailed in the appendix (A.9.1) due to its size. In the following it is already assumed that β'_m is zero, since t_1 and t_2 will be set to β_n and β_m. By calculating the partial derivative and solving for β'_n the following update scheme can be used to obtain the local optimal value $\hat{\beta}'_n$:

$$\hat{\beta}'_n \leftarrow \frac{1}{a_{nms}} \begin{cases} b_{nms} - \lambda\alpha(\ \ v_n\cos\theta - v_m\sin\theta + (-u_n\cos\theta + u_m\sin\theta)v_s/u_s) & \text{if } \hat{\beta}_n > 0 \wedge \hat{\beta}_m > 0 \wedge \hat{\beta}_s > 0 \\ b_{nms} - \lambda\alpha(\ \ v_n\cos\theta - v_m\sin\theta + (\ \ u_n\cos\theta - u_m\sin\theta)v_s/u_s) & \text{if } \hat{\beta}_n > 0 \wedge \hat{\beta}_m > 0 \wedge \hat{\beta}_s < 0 \\ b_{nms} - \lambda\alpha(\ \ v_n\cos\theta + v_m\sin\theta + (-u_n\cos\theta + u_m\sin\theta)v_s/u_s) & \text{if } \hat{\beta}_n > 0 \wedge \hat{\beta}_m < 0 \wedge \hat{\beta}_s > 0 \\ b_{nms} - \lambda\alpha(\ \ v_n\cos\theta + v_m\sin\theta + (\ \ u_n\cos\theta - u_m\sin\theta)v_s/u_s) & \text{if } \hat{\beta}_n > 0 \wedge \hat{\beta}_m < 0 \wedge \hat{\beta}_s < 0 \\ b_{nms} - \lambda\alpha(-v_n\cos\theta - v_m\sin\theta + (-u_n\cos\theta + u_m\sin\theta)v_s/u_s) & \text{if } \hat{\beta}_n < 0 \wedge \hat{\beta}_m > 0 \wedge \hat{\beta}_s > 0 \\ b_{nms} - \lambda\alpha(-v_n\cos\theta - v_m\sin\theta + (\ \ u_n\cos\theta - u_m\sin\theta)v_s/u_s) & \text{if } \hat{\beta}_n < 0 \wedge \hat{\beta}_m > 0 \wedge \hat{\beta}_s < 0 \\ b_{nms} - \lambda\alpha(-v_n\cos\theta + v_m\sin\theta + (-u_n\cos\theta + u_m\sin\theta)v_s/u_s) & \text{if } \hat{\beta}_n < 0 \wedge \hat{\beta}_m < 0 \wedge \hat{\beta}_s > 0 \\ b_{nms} - \lambda\alpha(-v_n\cos\theta + v_m\sin\theta + (\ \ u_n\cos\theta - u_m\sin\theta)v_s/u_s) & \text{if } \hat{\beta}_n < 0 \wedge \hat{\beta}_m < 0 \wedge \hat{\beta}_s < 0 \\ \text{derivative not defined, skip update} & \end{cases} , \tag{3.22}$$

with

$$\begin{aligned} a_{nms} = {} & \sum_{i=1}^{N} w_i\Big(x_{im}\sin\theta - x_{in}\cos\theta + \frac{x_{is}}{u_s}(u_n\cos\theta - u_m\sin\theta)\Big)^2 + \lambda(1-\alpha)\cdot\Big(v_n\cos^2\theta + v_m\sin^2\theta \\ & + \frac{v_s}{u_s^2}\Big(-u_n\cos\theta + u_m\sin\theta\Big)^2\Big), \end{aligned} \tag{3.23}$$

$$\begin{aligned} b_{nms} = {} & -\sum_{i=1}^{N} w_i\Big(x_{im}\sin\theta - x_{in}\cos\theta + \frac{x_{is}}{u_s}(u_n\cos\theta - u_m\sin\theta)\Big)\cdot\Big(y_i - \beta_0 - \sum_{\substack{j=1 \\ j\neq s,n,m}}^{p} x_{ij}\beta_j - x_{in}t_1 - x_{im}t_2 \\ & - \frac{x_{is}c}{u_s} + \frac{x_{is}}{u_s}\Big(\sum_{\substack{j=1 \\ j\neq s,n,m}}^{p} u_j\beta_j + u_n t_1 + u_m t_2\Big)\Big) - \lambda(1-\alpha)\Big(v_n t_1\cos\theta - v_m t_2\sin\theta \\ & + \frac{v_s}{u_s^2}\Big(c - \sum_{\substack{j=1 \\ j\neq n,m,s}}^{p} u_j\beta_j - u_n t_1 - u_m t_2\Big)\cdot\Big(-u_n\cos\theta + u_m\sin\theta\Big)\Big). \end{aligned} \tag{3.24}$$

The derivation is detailed in the appendix A.9.
By applying the back transformation

$$\begin{pmatrix}\beta_n\\ \beta_m\end{pmatrix} = \begin{pmatrix}\cos\theta & \sin\theta\\ -\sin\theta & \cos\theta\end{pmatrix}\begin{pmatrix}\beta'_n\\ \beta'_m\end{pmatrix} + \begin{pmatrix}t_1\\ t_2\end{pmatrix}, \tag{3.25}$$

the local optimal values for $\hat{\beta}_n$ and $\hat{\beta}_m$ can be obtained. The optimal value $\hat{\beta}_s$ is as above fixed by (3.16). t_1 and t_2 are set in the following to β_n and β_m to simplify the calculations in the transformed search space. This update scheme will be referred to in the following as *rotated update*.

As for the *normal update*, each case has to be evaluated and verified whether the update is allowed. However, since *rotated updates* have twice as many cases and require a back transformation in each case, they are overall more complicated to calculate. This results in *rotated updates* being more demanding to compute than *normal updates*. Therefore, a combination of both will be used.

Another issue is that both updates are not able to set coefficients exactly to zero and coefficients which are almost but not exactly zero ($|\beta_j| < 10^{-8}$) occur. In order to get rid of these artifacts, a small local search can be used, which will be referred to in the following as *polish*. The concept is to set small coefficients to zero by transferring their values to other non-zero coefficients if the cost function is not impaired. This transfer is necessary to prevent a deviation from the zero-sum constraint.

With these coordinate descent update schemes and the *polish* an algorithm can be constructed. In contrast to the coordinate descent for regressions without the zero-sum constraint, the number of possible *normal updates* scales quadratically with the number of features. The number of possible *rotated updates* even scales to the power of three (if the rotation angle θ is neglected).

Therefore, the following adaptive coordinate descent procedure will be applied to use these updates effectively:

1. Start with $\boldsymbol{\beta} = \vec{0}$ or initialize $(\beta_0, \boldsymbol{\beta})$ with values obtained by a previous calculation as *warm start*.
2. Cycle once over all unique combinations of the coefficients k and s ($k \neq s$) using the *normal update*.
3. Cycle once over all combinations of the coefficients k and s where $\beta_k \neq 0 \wedge \beta_s \neq 0$ using the *polish* to remove small non-zero artifacts.
4. Cycle once over all combinations of the coefficients k and s where $\beta_k \neq 0 \wedge \beta_s \neq 0$ using the *normal update* and determine the number of rejected updates.
5. If more than 10% of the *normal updates* were rejected cycle once over all combinations of the coefficients k and s where $\beta_k \neq 0 \wedge \beta_s \neq 0$ using the *rotated updates*. For each update an additional coefficient h ($\beta_h \neq 0 \wedge h \neq k \wedge h \neq s$) and an angle σ_i are randomly selected.
6. Repeat 4. and 5. while the cost function is improving within a defined precision (default 10^{-8}).
7. Cycle once over all combinations of the coefficients k and s where $\beta_k \neq 0 \wedge \beta_s = 0$ using the *normal update* scheme. If a non zero coefficient is set to (almost) zero or a zero coefficient becomes non zero go back to 3. else your done.

This procedure is attempting to use *rotated updates* to navigate out of search space regions, where the *inaccessible issue* occurs.

Another possible issue is the quadratic approximation used in the logistic, multinomial and Cox regression case which may have adverse effects on the optimization. For that reason, this procedure may not result in the global optimum.

3.3 Optimizing the Hyperparameter λ

This section details how the *normal update* can be used for constructing a λ sequence that is tested in cross-validation as described in the previous chapter.

To determine an upper bound of the regularization weight λ, it has to be considered that due to the zero-sum constraint – despite of the trivial solution ($\boldsymbol{\beta} = \vec{0}$) – at least two coefficients have to be non zero. Thus, the solution where λ is chosen a little bit smaller than $\lambda_{\max}$, which results the trivial solution, has to be given by exactly one *normal update*. This specific update has to be applied on the trivial solution $\boldsymbol{\beta} = \vec{0}$ and $c = 0$ and thus the first and the forth case in the case differentiation of the *normal update* (3.18) cannot occur. The reason for this is that in both cases the resulting optimal values are of equal sign and since $u_j > 0$ for all features j, these two cases would violate the zero-sum constraint (3.9). Moreover, it can be seen in the second case that $\hat{\beta}_k$ is only larger than zero if $b_{ks} - \lambda\alpha(v_k + v_s u_k/u_s)$ is greater than zero, since a_{ks} is always greater than zero. The case that a_k is zero in the ridge case ($\alpha = 0$) is avoided as described in section 2.6 by temporary assigning a small value (typically 0.01) to α. Thus, b_{ks} needs to be larger than $\lambda\alpha(v_k + v_s u_k/u_s)$. The same consideration applied on the third case implies that b_{ks} needs to be lower than $-\lambda\alpha(v_k + \frac{v_s u_k}{u_s})$. Consequently, neither of these conditions can occur, if

$$\lambda\alpha(v_k + \frac{v_s u_k}{u_s}) > |b_{ks}| \tag{3.26}$$

$$\lambda\alpha(v_k + \frac{v_s u_k}{u_s}) > \left| \sum_{i=1}^{N} w_i(-x_{ik} + \frac{x_{is} u_k}{u_s})(y_i - \beta_0) \right| . \tag{3.27}$$

Demanding that this should hold for all combinations of k and s, results in the following approximation for $\lambda_{\max}$:

$$\lambda_{\max} \approx \max_{\forall s,k} \frac{\left| \sum_{i=1}^{N} w_i(-x_{ik} + \frac{x_{is} u_k}{u_s})(y_i - \beta_0) \right|}{\alpha(v_k + \frac{v_s u_k}{u_s})} . \tag{3.28}$$

Analogously to section 2.6, a λ sequence can be constructed and a optimal value for λ determined using cross-validation.

3.4 Alternative Optimization Strategies

The coordinate descent algorithm for solving the zero-sum regression problem described in section 3.2 is based on assumptions which can cause that the optimal solution is not found. This is on the one hand due to optimizing the quadratic approximation and not the likelihood itself and on the other hand due to algorithmic issues which might cause that the coordinate descent procedure gets stuck.

The first issue can be resolved by using optimization algorithms which are not based on the partial derivative and therefore can directly operate on the log-likelihood. The second issue can be tackled by using arbitrary search space directions as proposed with the *rotated updates* (see section 3.2). However, such updates result in an indefinite amount of search space directions. For that reason, an algorithm has to be restricted to a specific set of directions in order to solve the problem within a reasonable amount of time. This in turn causes that the issue is only partially resolved.

Another approach for resolving these issues is to use global optimization algorithms that are able to escape from local optima. Therefore, they are also able to escape from regions where every accessible search space direction would lead to a worse configuration. For that reason, the general purpose optimization algorithms simulated annealing and local search are used to optimize the zero-sum regression problem. The advantage of these algorithms is that they do not depend on the derivative and thus can operate on the actual log-likelihood. Both algorithms are based on local coordinate wise alterations like the coordinate descent algorithm. However, simulated annealing does not have the issue of getting stuck, as it also accepts updates to worse configurations. This is not the case for local search, which is thus not immune to this problem.

The main drawback of simulated annealing and to some extent of local search is the much higher computational demand in contrast to the coordinate descent approach. For that reason, simulated annealing is in the following only used to verify the results obtained by the coordinate descent algorithm.

The motivation for the local search approach is that it can also be used to solve the generalized lasso problem described in section 2.10 with the zero-sum constraint within a reasonable time frame.

3.4.1 Simulated Annealing

Simulated annealing was proposed by Kirkpatrick et al. [44, 45] as a general purpose optimization algorithm and is based on Markov chain Monte Carlo methods. The concept is to represent an optimization problem as a thermal equilibrium at a specific temperature. Initially, this temperature has to be chosen high enough to allow all possible solutions of a problem to occur. Afterwards, the temperature is reduced bit by bit, which forces the system into more optimal solutions. If the cooling is performed slow enough the system freezes in its global optimal state [57].

In order to simulate a thermal equilibrium the cost function is used as an energy function and a sequence of solutions is generated according to the Boltzmann distribution with the help of a Markov chain [33]. A Markov chain can be constructed by successively using a transition scheme with a specific probability $P(\mu \to \nu)$ which creates a new state ν from a previous state μ. This probability needs to fulfill the following conditions [59, 64]:

- $P(\mu \to \nu)$ may only depend on the configurations μ and ν and not on previous states.
- All probabilities have to be positive ($P(\mu \to \nu) \geq 0$ for all μ, ν including $\mu = \nu$) and have to add up to one:

$$\sum_{\nu} P(\mu \to \nu) = 1\,. \tag{3.29}$$

Since the probability of every state is non-zero according to the Boltzmann distribution, the Markov chain has to be able to reach any state if the chain is long enough. This property is known as *ergodicity* and can be accomplished with the sufficient condition of detailed balance and the incorporation of the Boltzmann distribution [59]. The condition of detailed balance is

$$p_\mu P(\mu \to \nu) = p_\nu P(\nu \to \mu)\,, \tag{3.30}$$

where p_μ is the probability of the state μ and p_ν corresponds to the probability of the state ν. Since these probabilities should be given by the Boltzmann distribution, the following equation is obtained:

$$\frac{P(\mu \to \nu)}{P(\nu \to \mu)} = \frac{p_\nu}{p_\mu} = e^{-\left(\mathcal{H}(\nu)-\mathcal{H}(\mu)\right)/T}\,. \tag{3.31}$$

The cost function $\mathcal{H}$ of the optimization problem has to be used as an energy function and the Boltzmann constant can be omitted, since the temperature is used as a control parameter and should not represent a real physical temperature. However, the temperature defines an equilibrium which is reached if a sufficient amount of transitions with such probabilities are performed.

In order to generate such transitions computationally, the acceptance-ratio method can be used, which separates the transition probability into two different components [59]:

$$P(\mu \to \nu) = g(\mu \to \nu)A(\mu \to \nu)\,, \tag{3.32}$$

where $g(\mu \to \nu)$ is the selection probability and $A(\mu \to \nu)$ denotes the acceptance probability. Using this method allows an algorithm to generate new configurations in a more or less arbitrary way, which are then accepted or rejected using the acceptance probability to comply with the transition probability.

One of the most known algorithms using this technique is the Metropolis algorithm, which was proposed by Metropolis et al. [56]. The concept of this algorithm is that the selection probability should be symmetric $g(\mu \to \nu) = g(\nu \to \mu)$ [59, 64]. Using this symmetry and the acceptance-ratio (3.32) method, combined with the equation of detailed balance (3.31), yields [59]:

$$\frac{P(\mu \to \nu)}{P(\nu \to \mu)} = \frac{g(\mu \to \nu)A(\mu \to \nu)}{g(\nu \to \mu)A(\nu \to \mu)} = \frac{A(\mu \to \nu)}{A(\nu \to \mu)} = e^{-\left(\mathcal{H}(\nu)-\mathcal{H}(\mu)\right)/T}\,. \tag{3.33}$$

This can be fulfilled by using the Metropolis criterion [56]:

$$A(\mu \to \nu) = \begin{cases} e^{-\left(\mathcal{H}(\nu)-\mathcal{H}(\mu)\right)/T} & \text{if } \mathcal{H}(\nu) - \mathcal{H}(\mu) > 0 \\ 1 & \text{else} \end{cases}\,. \tag{3.34}$$

For each newly selected configuration the change of the cost function $\Delta\mathcal{H} = \mathcal{H}_\nu - \mathcal{H}_\mu$ between the new configuration ν and the current configuration μ has to be calculated to obtain the acceptance probability of this transition. If the new configuration ν is better (lower cost function) it is always accepted, otherwise it is only accepted with the probability $e^{-\Delta\mathcal{H}/T}$. By using this criterion to generate new configurations sufficiently often, a thermal equilibrium is reached.

In order to determine a suitable initial temperature, a *random walk* with the selection procedure $g(\nu \to \mu)$ can be used [69]. Since the system should initially be able to move freely through the search space, a suitable temperature can be estimated by calculating the temperature with which ~90% of the transitions to worse configurations would have been accepted by the Metropolis criterion [69].

After a thermal equilibrium is achieved the temperature has to be reduced and again enough steps have to be performed until a new equilibrium is reached. In theory, simulated annealing is capable of finding the global optimum if the system is cooled down slowly enough [57]. However, such a cooling procedure is beyond practical application as it would require an unfeasible amount of computing time. Hence, other more functional cooling schedules like exponential cooling have been established [69]. For that reason, exponential cooling with

$$T(t) = \alpha^t \cdot T_{\text{Start}}\,, \tag{3.35}$$

is used in the *zeroSum* implementation. The parameter α is usually set to a value between 0.8 and 0.999 and t denotes the number of performed Markov chains [69]. The *zeroSum* implementation uses a value of 0.8, since the search space of the zero-sum regression problem should not exhibit local optima despite the *inaccessible issue*.

In conclusion, the overall structure of simulated annealing is as follows [69]:

1. Determine a suitable temperature with a *random walk*.
2. Use the Metropolis criterion until a thermal equilibrium is reached.
3. Decrease the temperature of the system.

4. Repeat the steps 3. and 4. until the cost function of the system is not improving any more.

The crucial part of simulated annealing is the construction of an efficient transition procedure for the specific problem, which is commonly named *move* [69]. One import aspect of this *move* is that it should be possible to calculate the cost of the next configuration ν by only incorporating the change into the previous calculated cost of the configuration μ. Therefore, the cost function has to be computed only once and the cost only has to be updated after a successful transition. However, updating the cost function may lead to numerical instabilities. Thus, the cost function is recalculated after each temperature reduction.
Similar to the *normal update* of the coordinate descent algorithm, a *move* can be constructed by randomly selecting two coefficients s and k ($k \neq s$) and adjusting their values. However, they are not adjusted to their local optimal value, but randomly altered.
The zero-sum constraint is sustained by using a *move*, which adds a random amount δ to the coefficient β_k and subtracts $\frac{u_k}{u_s}\delta$ from β_s:

$$\beta_k^{\text{new}} = \beta_k^{\text{old}} + \delta\,, \qquad \beta_s^{\text{new}} = \beta_s^{\text{old}} - \frac{u_k}{u_s}\delta\,. \tag{3.36}$$

δ has to be chosen large enough in order to be able to move fast through the search space, but also small enough to find the optimal solution. During the implementation, the interval $[-0.1, 0.1]$ has proven to be an appropriate choice for all encountered data sets. Though, for other data sets different choices or an adaptive procedure may be more efficient. However, simulated annealing is only used in this thesis to verify the coordinate descent optimization strategy and therefore this small interval and more computing time than necessary were used.

3.4.2 Local Search

By setting the temperature of simulated annealing to zero, a local search algorithm is obtained, which only accepts *moves* to better configurations. However, to achieve a reasonable performance, the following extended structure, which also incorporates the active set cycling concept, will be used:

1. Cycle over all possible coefficients using the local search *move*.
2. Cycle over all non-zero coefficients using the local search *move*.
3. Cycle over all non-zero coefficients using the local search *move* but with $\delta = \beta_k$, since sparse solutions are more likely.
4. Repeat the steps 1. to 3. while the cost function is improving within a defined precision (default 10^{-8}).

This procedure also allows to obtain solutions in the generalized lasso case, but it is advisable to first use coordinate descent to solve the problem without the generalized lasso and then to apply local search with the generalized lasso on the obtained solution as a *warm start*. Thereby an adequate performance can be achieved.
More sophisticated approaches would be possible like, for example, a Nelder-Mead based procedure. However, since the generalized lasso is only a special case and since it was possible to obtain solutions within a reasonable time frame, no further efforts have been made to develop a more advanced procedure.

3.5 Implementation Details

3.5.1 Calculation of the Cost Function Using Previous Calculations

In order to implement an efficient *move* and coordinate descent update, it is useful to store the residuals of all samples (without applying the link function) in memory. These residuals are given by:

$$\text{res}_i^{\text{current}} = y_i - \beta_0 - \sum_{j=1}^{p} \beta_j^{\text{current}} x_{ij} \,. \tag{3.37}$$

Using a coordinate descent update or simulated annealing *move* only alters the coefficients β_k and β_s. Thus, the residuals of the next configuration can be calculated using the current residuals and incorporating the change of these two coefficients:

$$\text{res}_i^{\text{next}} = \text{res}_i^{\text{current}} + x_{ik}(\beta_k^{\text{current}} - \beta_k^{\text{next}}) + x_{is}(\beta_s^{\text{current}} - \beta_s^{\text{next}}) \,. \tag{3.38}$$

Analogously, the elastic net and generalized lasso regularization of the next configuration can be determined. This reduces the computing effort of the coordinate descent update scheme as well as the computing effort of the local search/simulated annealing *move*.

3.5.2 Preventing Numerical Instabilities

In some rare cases, the weights of the quadratic approximation $\tilde{w}_i$ can, at the end of the optimization, become very small ($<10^{-6}$) and can range over orders of magnitudes (10^{-15} up to 10^{-6}). Summations using such small weights can lead to numerical instabilities which can distort the accuracy of the approximation to such an extend, that coordinate update steps result in worse configurations. Since this occurs at the end of the optimization procedure, the coordinate descent is extended to also stop if the percentage of weights $\tilde{w}_i$ smaller than 10^{-6} is higher than 70%. This issue has only been observed in artificial data sets and these settings have proven to resolve it without worsening the outcome.

Another issue that can occur when using multinomial or Cox regression is that the largest possible number representable in double precision can be exceeded due to the exponential functions. In particular the logarithm of a sum of exponential functions, which is part of the multinomial log-likelihood function (2.43) and Cox regression log-likelihood (2.66), can be very unstable. This is a common problem and therefore numerical more stable versions have been developed like the *logSumExp* function in *R* [66] or the *logsumexp* function of the *SciPy* python library [41]. Both functions are based on the approach:

$$\log\Big(\sum_i e^{x_i}\Big) = a + \log\Big(\sum_i e^{x_i - a}\Big), \tag{3.39}$$

where a is set to the maximum value of $\boldsymbol{x}$. Thereby, the exponential function is prevented to overflow. This approach is crucial for the numerical stability of multinomial and Cox regression. Therefore, the following extension with weights is used in the *zeroSum* software:

$$\log\Big(\sum_i w_i e^{x_i}\Big) = a + \log\Big(\sum_i w_i e^{x_i - a}\Big). \tag{3.40}$$

3.5.3 zeroSum - the Software

All algorithms presented in this thesis are implemented in the software *zeroSum*, which has three different levels of parallelization to achieve a high performance. First, vectorization is used for leveraging the performance of the cost function, the coordinate descent update schemes and the simulated annealing/local search *move* using AVX, AVX2 and AVX512 compiler intrinsics. These instructions allow to simultaneously perform 4 or 8 (with AVX512) double precision operations and can be applied for most of the calculations. For that reason using AVX(2) or AVX512 directly translates to a 4- or 8-fold performance increase.

The optimization algorithms itself have not been parallelized, although parallel extensions of these algorithms exist like, for example, *shotgun* [8] for coordinate descent and *parallel tempering* [38] for simulated annealing. However, these extensions have not been used, since the cross-validation procedure for the determination of an optimal λ can be considered as an *embarrassingly parallel* problem, which can be more effectively parallelized. This is achieved by using OpenMP, which is not only easy to implement but also allows every fold of the cross-validation to access the same memory. Thus, the data only has to be stored once. This is the second level of parallelization of *zeroSum*.

Another distributed memory parallelism is implemented using MPI (message passing interface) and is used for the generalized lasso regularization. In this case, the search space for the optimal combination of λ and γ translates to a two dimensional grid space. Parallelizing the search along the λ direction only leads to small performance improvements, since solutions obtained from higher λ values can be used as *warm starts* for lower values. However, such *warm starts* are less efficient along the γ direction, which is thus parallelized with MPI. This parallelization allows to distribute the problem among different computers and therefore allows the use of computing clusters. To balance the computing effort of different nodes, the values of γ are randomly distributed among the MPI processes, since different scales of γ cause that the computational challenge of the optimization problems can be different.

Moreover, to allow for a better hardware utilization per node, the γ direction is additionally parallelized with OpenMP. Thus, one MPI process per CPU results in the best performance, since the OpenMP parallelization is capable to access the same memory, which not only reduces the total memory requirement, but also allows for better cache utilization.

Another characteristic of the MPI parallelization is that it is not advisable to distribute every γ value to an individual node, since common computing nodes will not be fully utilized. The reason for this is, that in a typical situation a λ and γ sequence, each containing 100 values, should be evaluated with a 10-fold cross-validation. It is possible to distribute the problem among 100 nodes, however, on each node only 11 cores – 10 for the cross-validation and 1 containing all samples – are used. Since 11 cores per node are not common, all additional cores would not be used and, for instance, a node containing 16 cores would only be utilized to ~69%. If, on the contrary, 10 nodes are used, every node would have to evaluate 10 different values of γ. This causes that 110 different parallelizable tasks have to be processed and a 16 core node would require 7 iterations and would be utilized to 98%. Hence, in this example a tenfold increase of nodes from 10 to 100 only results in a speedup of 7 and a significant deterioration of the utilization.

Another option for obtaining a better utilization would be to use settings, which are more adapted to the conditions of computers. For example a 7 or 15-fold cross-validation could fully utilize a 8 core CPU (one additional core for all samples). However, settings like a 10 fold cross-validation and a λ sequence length of 100 have been established [32, 39] and thus these settings will be used as default and will also be used throughout this thesis.

All algorithms are implemented in a C++ core, which is the common base of a console interface intended for HPC usage and of an R-package for easy use. The source code and build instructions are available at `https://github.com/rehbergT/zeroSum`. For all simulations and data evaluations in this thesis version 1.1.1 has been used.

3.6 Convergence Behavior

In this section the convergence behavior of the presented algorithms and the corresponding extensions is evaluated. Therefore, these different approaches were applied on subsets of data sets, which are presented and analyzed in chapter 5. At first, the microbiome data set of section 5.1 is used for zero-sum linear regression, while the blood plasma metabolomics data of section 5.2 is used for zero-sum logistic regression. To analyze zero-sum multinomial regression the DNA-methylation data set of section 5.4 is used. Subsequently, Cox regression is applied on the gene expression data and the corresponding survival data of section 5.5. These data sets are only used in this section to investigate the convergence behavior of the zero-sum regression algorithms and are thus despite the obtained costs not further investigated. Hence, more precise information about the data sets are not necessary for this section, but are given in the corresponding sections of chapter 5.

In order to evaluate the convergence behavior under different conditions, 1000 subsets containing 50 features for 20 samples were randomly sampled from these data sets. Subsequently, each subset was analyzed using a 10-fold cross-validation regression with $\alpha = 1$ in order to determine a optimal value for λ. This value for λ and $\alpha = 1$ were then used to train linear models for each subset with the following different algorithms and settings:

- Coordinate descent (CD): This procedure only uses *normal updates* (3.18) without *rotated updates* and without *polish*.
- Coordinate descent with *polish* (CD+P): This procedure uses *normal updates* and applies the *polish* procedure.
- Coordinate descent with *rotated updates* (CD+R): This procedure uses *normal updates* and *rotated updates* (3.22). No *polish* is applied on the final result.
- Coordinate descent with *rotated updates* and *polish* (CD+R+P): This procedure uses *normal updates*, *rotated updates* and the additional *polish* procedure.
- Coordinate descent using *warm starts* (CD*): The same as CD but the regularization sequence from λ_{max} down to λ is evaluated using the solution of the higher value of λ as *warm start* for the next lower value. *Warm starts* are the usual case, since the optimal value for λ is typically not known. Moreover, it was mentioned that *warm starts* also increase the stability of the algorithm [26, 70].
- Coordinate descent with *polish* using *warm starts* (CD+P*): The same as CD+P but also using *warm starts*.
- Coordinate descent with *rotated updates* using *warm starts* (CD+R*): The same as CD+R but also using *warm starts*.
- Coordinate descent with *rotated updates* and *polish* using *warm starts* (CD+R+P*): The same as CD+R+P but also using *warm starts*.
- Local search (LS): This procedure uses the local search algorithm as described in section 3.4.2.
- Simulated annealing (SA): This procedure uses simulated annealing as described in section 3.4.1 and is used to determine the optimal solution.
- Normal lasso control (ctrl): This procedure uses the coordinate descent without the zero-sum constraint as detailed in section 2.5, which is additionally implemented in the *zeroSum* software.

- Normal lasso reference (glmnet): This procedure uses the *glmnet* R-package [26] as independent implementation for creating models without the zero-sum constraint.

In order to evaluate to which extend a regression method is capable to reach the optimal solution (identified with simulated annealing) the deviation from the best obtained solution per subset and per method was determined. The distributions of these deviations are shown as box plots in figure 3.1 and the mean cost is shown table 3.1. The deviation from the best obtained solution measures the capability of an algorithm to yield the optimal solution and thus can indicate conceptual problems like the *inaccessible issue*. Therefore, a high mean and standard deviation in the box plots of figure 3.1, implies that the algorithm is insufficient to reach the optimal solution. Vice versa, a good algorithm exhibits a low mean and a low standard deviation.

Note that the costs of regressions with and without the zero-sum constraint cannot be compared even for equal λ, since the zero-sum constraint changes the optimization problem. For that reason, the final costs will always be slightly higher if the zero-sum constraint is enforced. This property can be observed by comparing the costs of the normal lasso (ctrl and glmnet in table 3.1) with the best obtained costs of zero-sum regression (SA).

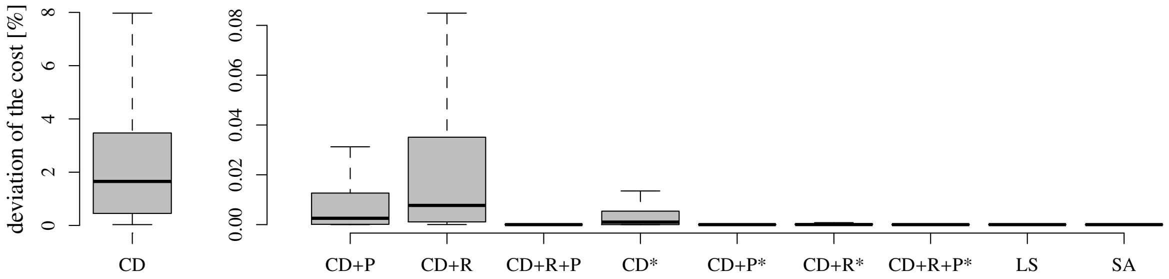
The normal lasso control (ctrl) and the *glmnet* R-package generate the exact same results. This aspect and the fact that all obtained solutions with the zero-sum constraint exhibit a cost which only differs slightly from the costs of the solutions obtained without the zero-sum constraint, indicate that the implementation of the software *zeroSum* is correct.

The mean cost (table 3.1) and the deviation from the best solution (figure 3.1) show that a naive coordinate descent (CD) itself is not sufficient to reach the optimal solution. In extreme cases the obtained solutions are 8% worse. This worst case deviation can be improved to less than 0.1% by using the *polish* (CD+P) or *rotated updates* (CD+R). Combining *rotated updates* and *polish* further improves the solutions to such an extend that they are (almost) equivalent to the best obtained ones. This indicates that the *polish* alone is not capable to reach the optimal solution due to the *inaccessible issue*, which however can be mitigated by the *rotated updates* (CD+R).

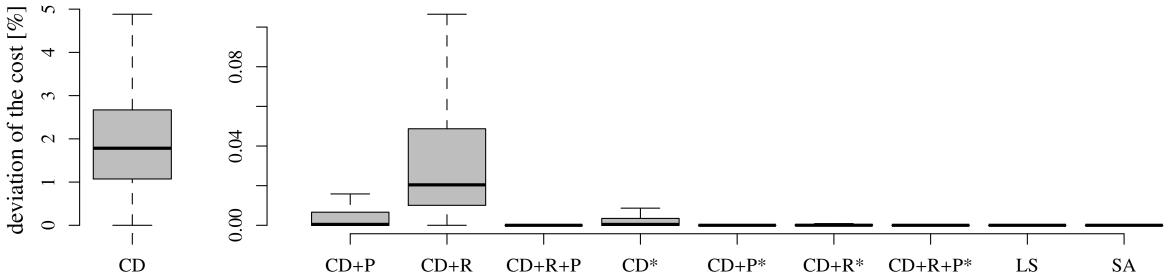
	CD	CD+P	CD+R	CD+R+P	CD*	CD+P*	CD+R*	CD+R+P*	LS	SA	ctrl	glmnet
linear	5.3539	5.1832	5.1855	5.1823	5.1826	5.1823	5.1824	5.1823	5.1823	5.1823	4.9462	4.9462
logistic	0.4752	0.4656	0.4659	0.4656	0.4656	0.4656	0.4656	0.4656	0.4656	0.4656	0.4545	0.4545
multi.	0.5516	0.5397	0.5402	0.5396	0.5383	0.5381	0.5381	0.5380	0.5379	0.5379	0.4973	0.4973
cox	-0.6493	-0.6502	-0.6502	-0.6502	-0.6502	-0.6502	-0.6502	-0.6502	-0.6502	-0.6502	-0.6620	-0.6620

Table 3.1: Shown is the the mean cost of each regression approach for linear regression, logistic regression, multinomial regression and Cox regression. CD denotes a zero-sum coordinate descent algorithm, which only uses *normal updates*. ”+P” denotes that an additional *polish* is applied and ”+R” denotes that *rotated updates* are additionally used. ”*” indicates that *warm starts* where utilized. LS is an abbreviation of local search and SA is an abbreviation of simulated annealing. The lower the cost the better are the obtained solutions of the algorithms.

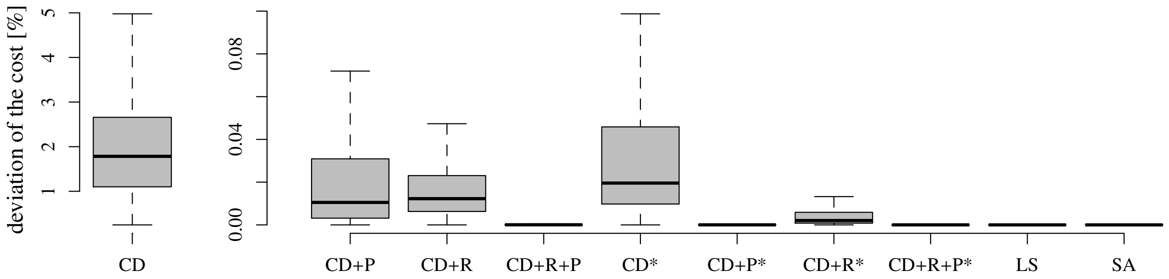
(a) Linear Regression



(b) Logistic Regression



(c) Multinomial Regression



(d) Cox Regression

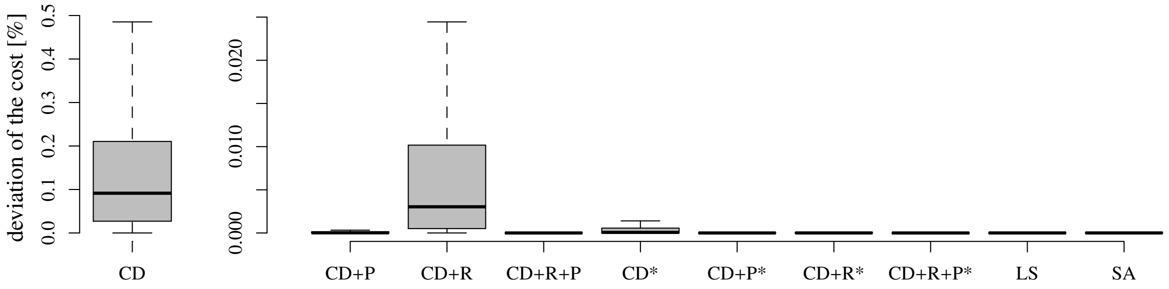


Figure 3.1: Shown is the distribution of deviations from the best found solution per regression approach for linear regression (a), logistic regression (b), multinomial regression (c) and Cox regression (d). CD denotes a zero-sum coordinate descent algorithm, which only uses *normal updates*. "+P" denotes that an additional *polish* is applied and "+R" denotes that *rotated updates* are additionally used. "*" indicates that *warm starts* where used in addition. LS is an abbreviation of local search and SA is an abbreviation of simulated annealing.

The use of *warm starts* improves the convergence behavior. Thus, even in the CD* case the optimal solutions are only slightly missed. *Rotated updates* (CD+R*), *polish* (CD+P*) as well as their combination (CD+R+P*) yield the optimal solutions. Simulated annealing and local search performed equally well. All in all, CD+R+P, CD+P*, CD+R* and CD+R+P* were capable to determine the optimal solutions.
Another important aspect is the required computing time. Therefore, the average run time of each method had been determined and is shown in table 3.2. As it can be seen, the runtime increases, when *rotated updates* are used additionally (CD+R, CD+R*) and is decreased when a *polish* is applied (CD+P, CD+P*). The reason for this is that the *polish* attempts to set coefficients with very small values to zero and thus causes that the number of non-zero coefficients is reduced. Hence, the active set cycling can be performed more efficiently, so that not only the required time of the *polish* is recovered but also the total runtime is decreased. This effect can also be seen if a combination of *rotated updates* and *polish* is used (CD+R+P, CD+R+P*).
As a consequence, when a regression for a specific λ should be determined, the combination of *rotated updates* and *polish* is used (CD+R+P) as default. When *warm starts* are utilized, i.e. in the cross-validation for the λ approximation, only the additional *polish* is applied (CD+P*), since this approach is sufficient to determine the optimal solution and also saves computing time.

	CD	CD+P	CD+R	CD+R+P	CD*	CD+P*	CD+R*	CD+R+P*	LS
linear (in ms)	2.194	1.368	24.331	16.712	75.037	74.966	95.627	89.464	43.904
logistic (in ms)	1.824	0.757	14.770	7.344	58.393	59.663	72.272	64.982	173.147
multi. (in s)	0.418	0.364	1.091	0.944	9.388	8.142	14.416	12.222	44.655
cox (in ms)	11.754	11.203	25.030	22.183	467.872	468.559	510.898	512.688	255.067

Table 3.2: Shown is the average runtime of each regression approach for linear regression (a), logistic regression (b), multinomial regression (c) and Cox regression (d). CD denotes a zero-sum coordinate descent algorithm, which only uses *normal updates*. ”+P” denotes that an additional *polish* is applied and ”+R” denotes that *rotated updates* are additionally used. ”*” indicates that *warm starts* where utilized. LS is an abbreviation of local search and SA is an abbreviation of simulated annealing.

4 Simulations

This chapter illustrates the effects of sample-wise data shifts on generalized linear models using simulated data. For that reason, increasingly larger sample-wise shifts are applied and regressions with and without the zero-sum constraint are performed. Afterwards, the accuracy, the selected features as well as the predictive power of the generated models are investigated. This approach is conducted separately for linear, logistic, multinomial and Cox regression and is detailed in the corresponding sections of this chapter.
The simulations for linear regression are part of [3] and have been repeated with the latest version of the glmnet (2.0-13) and zeroSum (1.1.1) software. Furthermore, the simulations have been extended to logistic, multinomial as well as Cox regression.

4.1 Linear Regression

The quality of a linear model can be assessed by considering three aspects. First, the error of a model on the training data is important to evaluate the capability of describing the underlying data. Second, the selected features are interesting by itself and can be used to gain new insights into the relations between data and response. Third, the predictive power of a model, which has to be assessed using independent data, allows to estimate the generalizability of a model and can be used for detecting overfitting. These aspects are closely linked to each other and involve the bias-variance *trade-off*, which is the central problem of supervised learning algorithms.
In order to analyze the influence of sample-wise data shifts on regressions with and without the zero-sum constraint, four scenarios (a)-(d) are analyzed according to these aspects. For each scenario a data set with 100 samples and 500 features is generated using a normal distribution with mean zero and standard deviation of 0.5. Additionally, a correlation structure is imposed on the first three features using a Cholesky decomposition. For reasons of simplicity, only the first three features should have an effect and were used to generate the responses. All other coefficients are set to zero. The coefficients as well as the imposed correlation structure of the scenarios (a)-(d) are as follows:

(a) In this scenario the first three coefficients have been set to $\beta_1 = 1, \beta_2 = -1, \beta_3 = 3$. These coefficients slightly differ from the zero-sum constraint and allow to analyze to which extend zero-sum regression is capable to compensate slight deviations from zero. Moreover, the first three features were simulated with the correlations $\text{cor}(\mathbf{x}_1, \mathbf{x}_2) = 0.9$, $\text{cor}(\mathbf{x}_1, \mathbf{x}_3) = 0.9$, $\text{cor}(\mathbf{x}_2, \mathbf{x}_3) = 0.8$.

(b) This scenario uses the same coefficients as (a). However, the correlations were set to $\text{cor}(\mathbf{x}_1, \mathbf{x}_2) = -0.9$, $\text{cor}(\mathbf{x}_1, \mathbf{x}_3) = 0.9$, $\text{cor}(\mathbf{x}_2, \mathbf{x}_3) = -0.8$ in order to exhibit anti-correlated features.

(c) This scenario uses the same correlation structure as (a). The coefficients have been set to $\beta_1 = 1$, $\beta_2 = 2, \beta_3 = 3$ and allow to investigate the behavior of zero-sum regression in a scenario which clearly distinguishes itself from the zero-sum constraint.

(d) In this scenario no correlations were imposed and the first coefficients have been set to $\beta_1 = 1$, $\beta_2 = -2, \beta_3 = 1$. This corresponds to the ideal case where the coefficients fulfill the zero-sum

> constraint. Thus, a regression without the zero-sum constraint should also not be affected by sample-wise shifts as long as it is capable of correctly identifying the coefficients.

Additionally, a normally distributed noise with standard deviation of 0.1 was added to the data sets after calculating the responses.
In order to simulate scaling problems, sample-wise shifts γ_i were drawn from a normal distribution with standard deviation σ, which was increased from 0.0 to 5.0 in steps of size 0.5. For each σ, a regression with and a regression without the zero-sum constraint was performed on 20 samples using a 10-fold cross-validation for optimizing λ. In total, this procedure has been repeated a thousand times.
The accuracy of the models on the training data was evaluated using the mean squared error (as defined in (2.8) with weights set to 2/N) and is shown as a function of σ in figure 4.1.
In order to verify whether the selected features correspond to the simulated effects, which will be referred to in the following as *true* coefficients, the *area under the receiver operating characteristic curve* (AUC) is used. An AUC of 1 denotes that a regression was able to identify the *true* coefficients, while an AUC of 0.5 equals random guessing. The resulting AUC distribution for scenario (a)-(d) is shown in figure 4.2 as a function of σ.
To investigate the predictive power, the remaining 80 observations were used as independent test data. Therefore, the accuracy of the predictions on the test data was assessed using the *coefficient of determination* R^2

$$\mathrm{R}^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y})^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2} \tag{4.1}$$

as well as the mean squared error. The higher the R^2, the better the predictions of a linear model. The upper bound of the R^2 is 1 and denotes a perfect model. The R^2 for the scenarios (a)-(d) is shown as a function of σ in figure 4.3 and the mean squared error of the test data is shown in figure 4.4.
In all figures of this chapter a blue dashed line depicts the median performance of the zero-sum models and a solid black line shows the median performance of the normal lasso models. The bright red, red and dark red bands correspond to the 1 to 99%, 5 to 95% and 25 to 75% quantiles of the normal lasso regression. Analogously, the bright blue, blue and dark blue bands denote the quantiles of zero-sum regression. However, zero-sum regression returns results independent from σ and thus the corresponding blue bands are not visible and the median is a horizontal line.
The accuracy of zero-sum as well as the accuracy of normal lasso regression on the training data is almost equivalent (figure 4.1). Though, the 1-99% quantiles of the normal lasso regression (bright red) indicate that in rare cases the accuracy of the normal lasso regression decreases.
In the majority of cases, the *true* coefficients are detected (indicated by the median AUC of 1 in scenarios (b)-(d)). However, in scenario (a) the normal lasso regression performs better than zero-sum regression in terms of the feature selection aspect. The reason for this apparently is, that one of the *true* coefficients is sufficient to completely describe the training data due to the high correlations. This is indicated by a mean squared error of almost zero on the training data (see figure 4.1). Scenario (b) exhibits anti-correlated features, which makes it easier for the lasso regularization to identify the *true* coefficients. Thus, both normal-lasso and zero-sum regression yield an AUC of 1. Since scenario (c) was simulated using only positive effects, the *true* coefficients are not directly accessible by zero-sum regression. Nevertheless, the AUC is only slightly lower than 1 which indicates that the regression selects the first 3 *true* coefficients and compensates the sum by selecting other features with small negative values. Scenario (d) was simulated using coefficients which fulfill the zero-sum constraint and therefore both regressions are able to identify the *true* coefficients.
The predictive power of zero-sum regression measured by the R^2 (figure 4.3) and the mean squared error (figure 4.4) in the scenarios (a) to (c) is for small σ worse than the performance of normal lasso

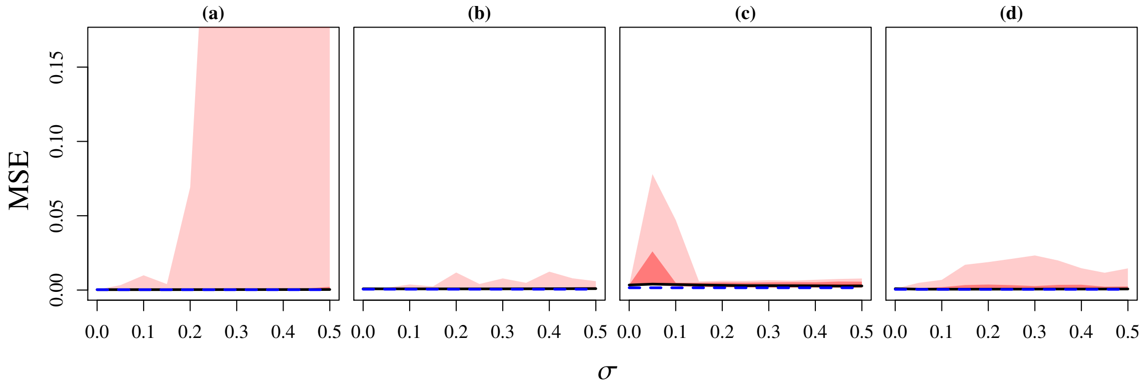


Figure 4.1: Shown is the mean squared error (MSE) on the training data for the scenarios (a)-(d) as a function of σ. A lower MSE corresponds to a better accuracy. The blue dashed line and the solid black line illustrate the median MSE over all repetitions for zero-sum regression and normal lasso regression. The bright red, red and light red bands show the 1 to 99%, 5 to 95% and 25 to 75% quantiles of the MSE distribution of normal linear regression. Zero-sum regression always returned the same results, due to the induced scale invariance. For that reason, the quantiles of zero-sum regressions, which would be indicated as blue band, are not visible.

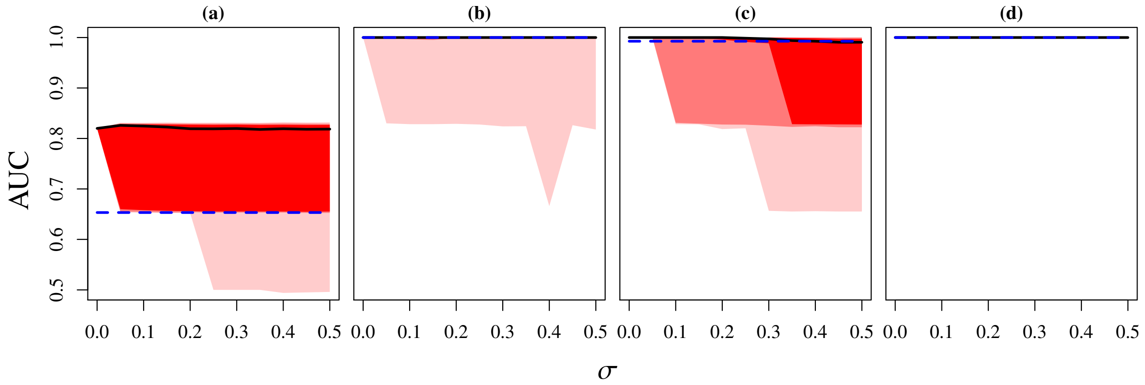


Figure 4.2: Shown is the *area under the receiver operating characteristic curve* (AUC) of the scenarios (a)-(d) as a function of σ. An AUC of 1 implies that the regression is capable of detecting the *true* coefficients and an AUC of 0.5 equals random guessing. The color definitions are analogously to figure 4.1.

regression. However, for larger σ the performance gets better. Moreover, normal lasso regression is in some cases impaired to such an extend that the predictive power completely breaks down ($R^2 \leq 0$). Only in the ideal scenario (d) both methods show a perfect prediction accuracy.

In summary, zero-sum regression is capable of achieving the same accuracy on the training data and is, apart from scenario (a), better suited for detecting the *true* coefficients. Most important however is that sample-wise shifts can worsen the prediction accuracy of normal lasso models, while zero-sum models are immune to these.

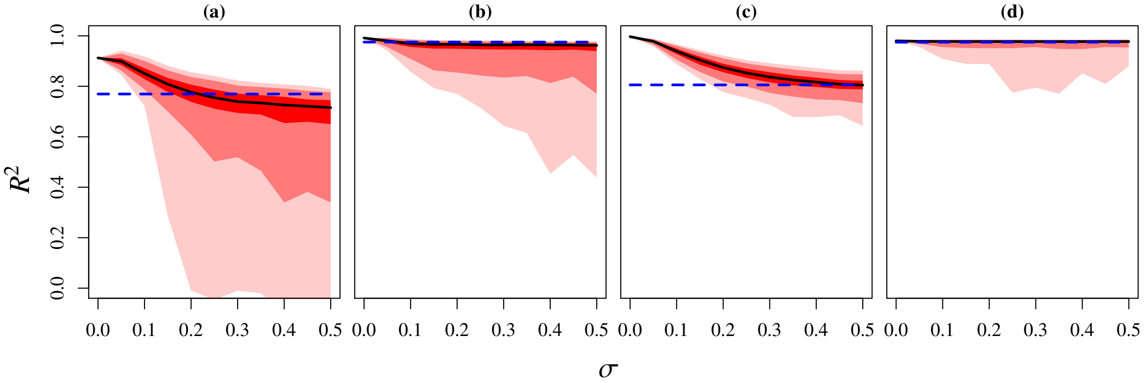


Figure 4.3: Shown is the coefficient of determination R^2 of the predictions on the test data for the simulation scenarios (a)-(d) as a function of the standard deviation σ. The higher the R^2, the better the predictivity of the models. A value of 1 corresponds to perfect predictions. The color definitions are analogously to figure 4.1.

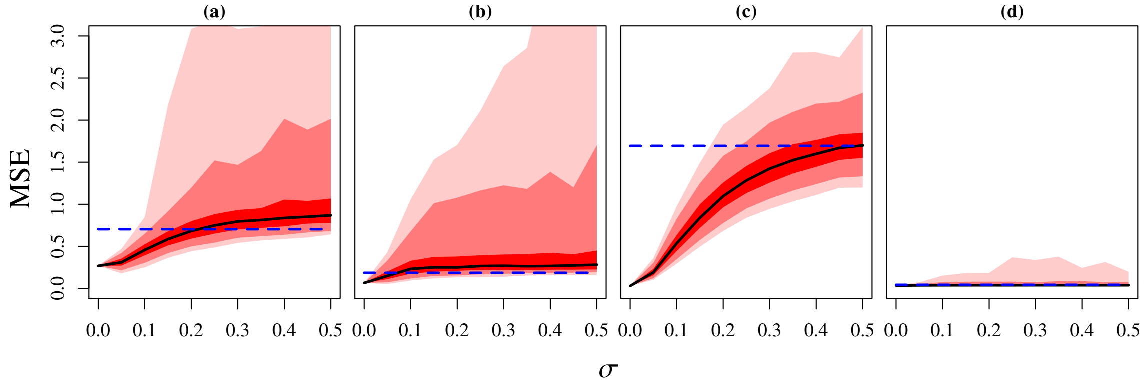


Figure 4.4: Shown is the mean squared error (MSE) of the predictions on the test data for the simulation scenarios (a)-(d) as a function of the standard deviation σ. The lower the MSE, the better the predictivity of the models. The color definitions are analogously to figure 4.1.

4.2 Logistic Regression

In this section the same simulation scenarios as in the last section are used. The only difference is that binary responses are simulated using the logistic function.

To measure the classification quality, *the area under the receiver operating characteristic curve* (AUC) on the training data (shown in figure 4.5), as well as the AUC on the test data (shown in figure 4.7) is used. The capability of identifying the *true* coefficients is also determined using the AUC and is shown in figure 4.6.

Overall, the same behavior as in the last section can be observed: logistic zero-sum regression is robust against sample-wise shifts while normal logistic regression is not. The accuracy of logistic zero-sum models on the test data is without any error (AUC=1), while normal logistic models sometimes correspond to random guessing (AUC=0.5).

Logistic zero-sum regression is capable of determining the *true* coefficients at least as good as normal logistic regression. Moreover, for high σ logistic regression breaks down and corresponds to random guessing. Both the low AUC of the normal logistic regressions on the training data as well as the corresponding low AUC of the feature selection indicate that the internal cross-validation is not able to

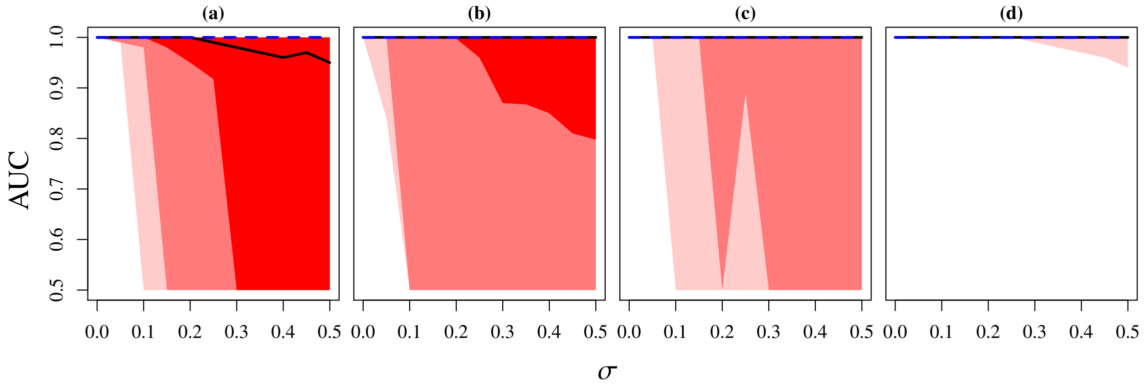


Figure 4.5: In this figure the *area under the receiver operating characteristic curve* (AUC) on the training data set is shown for the scenarios (a)-(d) as a function of σ. An AUC of 1 corresponds to a perfect classification rate, while an AUC of 0.5 equals random guessing. The color definitions are analogously to figure 4.1.

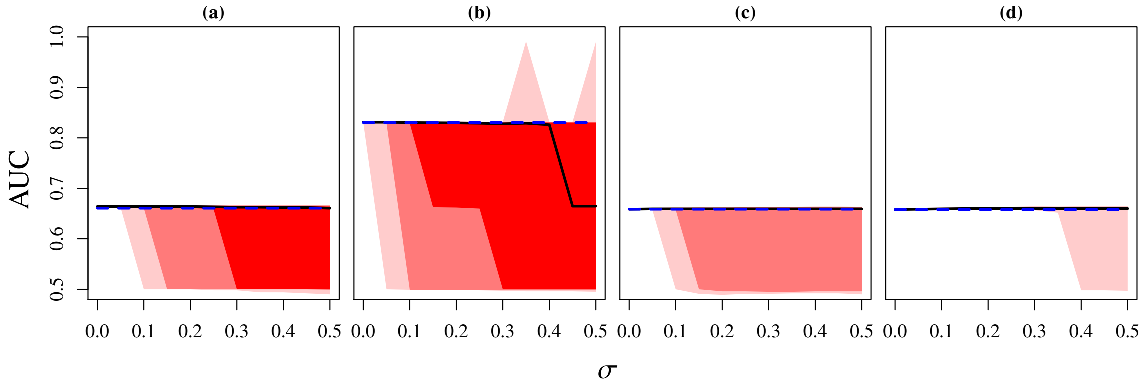


Figure 4.6: This figure shows the AUC of the scenarios (a)-(d) as a function of σ and indicates the capability of the regression to determine the *true* coefficients. An AUC of 1 implies that the regression is capable to identify the *true* coefficients, while an AUC of 0.5 equals random guessing. The color definitions are analogously to figure 4.1.

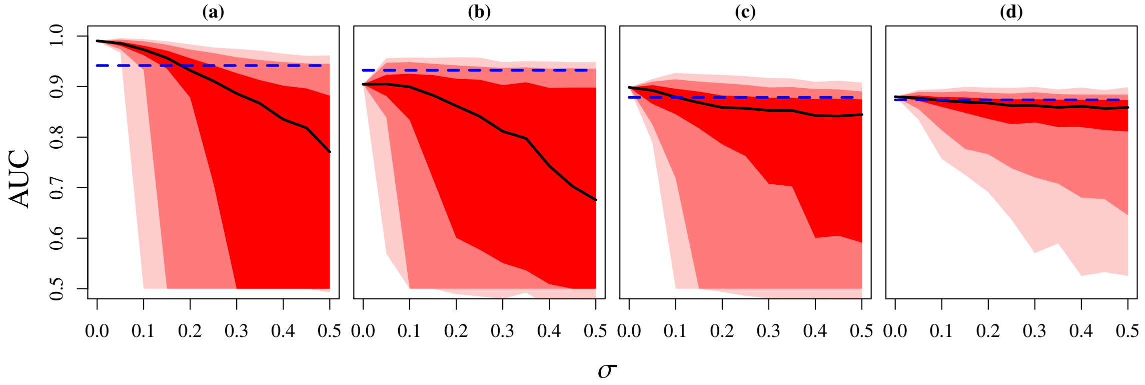


Figure 4.7: In this figure the AUC on the test data set is shown for the scenarios (a)-(d) as a function of σ. An AUC of 1 corresponds to a perfect classification rate, while an AUC of 0.5 equals random guessing. The color definitions are analogously to figure 4.1.

identify a optimal λ due to the data shifts. Thus, a model only consisting of an intercept is returned.
The predictive power of logistic zero-sum regression shows almost the same pattern as in the linear regression case: for low σ the normal logistic regression is better, while for high σ logistic zero-sum regression is better. In scenario (b) the median AUC of logistic zero-sum regression is overall higher than the AUC of normal logistic regression.
As linear zero-sum regression, logistic zero-sum regression is insensitive to sample-wise shifts and therefore maintains its predictivity, while normal logistic regression loses predictive power.

4.3 Multinomial Regression

As shown in section 2.8, multinomial regression uses a separate set of coefficients for the prediction of the probabilities of each class. To extend the simulation scenarios, described in the last two sections, three different classes were simulated using the coefficient matrices $\boldsymbol{\beta}^{(a)}$ for scenario (a), $\boldsymbol{\beta}^{(b)}$ for scenario (b) and $\boldsymbol{\beta}^{(d)}$ for scenario (d), which were defined as

$$\boldsymbol{\beta}^{(a)} = \boldsymbol{\beta}^{(b)} = \begin{array}{c} \begin{array}{ccc} \boldsymbol{\beta}_1 & \boldsymbol{\beta}_2 & \boldsymbol{\beta}_3 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 3 \\ 0 & 0 & 0 \\ \vdots & \vdots & \vdots \end{pmatrix} \end{array} \qquad \boldsymbol{\beta}^{(d)} = \begin{array}{c} \begin{array}{ccc} \boldsymbol{\beta}_1 & \boldsymbol{\beta}_2 & \boldsymbol{\beta}_3 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ \vdots & \vdots & \vdots \end{pmatrix} \end{array}. \tag{4.2}$$

Scenario (c) of the previous sections is not reasonable in the multinomial case, since the zero-sum constraint relaxes to an equal sum constraint. Therefore, scenario (c) is omitted.
As in the last section, the coefficients in scenario (a) and (b) do not fulfill the equal sum constraint, while scenario (d) does. Moreover, scenario (a) is simulated using correlated features, while scenario (b) is simulated using anti-correlated features. Scenario (d) has no enforced correlations. The coefficients and the correlation structures are summarized in (4.2) and table 4.1.

Scenario	cor($\mathbf{x}_1, \mathbf{x}_2$)	cor($\mathbf{x}_1, \mathbf{x}_3$)	cor($\mathbf{x}_2, \mathbf{x}_3$)
(a)	0.9	0.9	0.8
(b)	-0.9	0.9	-0.8
(d)	-	-	-

Table 4.1: This table summaries the correlation structure of the first three features for the simulation scenarios (a), (b) and (d).

Furthermore, the number of training samples was increased to 30 and the number of test samples was increased to 120.
In order to measure the accuracy and the predictivity of the obtained models, the multinomial AUC, as proposed by Hand and Till [29], was computed using the implementation in the *HandTill2001* R-package [17]. The multinomial AUC is shown in figure 4.8 for the training data and in figure 4.10 for the test data. As in the previous sections, the *normal* AUC is used for assessing the capability of the regressions to determine the *true* coefficients and is shown in figure 4.9.
The overall pattern is the same as the one observable in the linear and logistic regression simulations: the capability of the normal multinomial regression to describe the data of the scenarios (a) and (b) is affected by the sample-wise shifts and sometimes breaks down completely (AUC=0.5). In all scenarios zero-sum multinomial regression is is able to identify the true coefficients at least as good as normal multinomial

regression. The low AUC on the training data as well as the low AUC of the feature selection indicate as in the last section that the internal cross-validation for assessing an optimal λ returns the *intercept only model*.

The predictive power of multinomial zero-sum regression in scenario (a) and (b) is for small σ worse than the predictive power of the normal multinomial regression (see figure 4.10). This changes for higher values of σ, since the normal multinomial regression is affected by the sample-wise shifts.

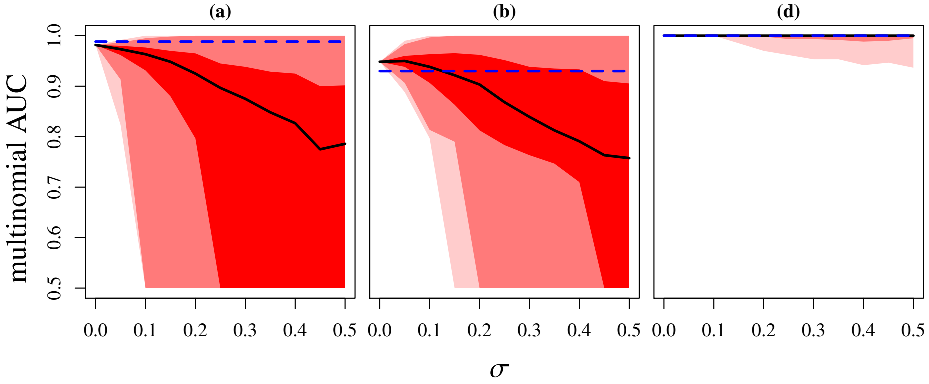


Figure 4.8: In this figure the multinomial *area under the receiver operating characteristic curve* (AUC) on the training data set is shown for the scenario (a), (b) and (d) (scenarios (c) was omitted in the multinomial case) as a function of σ. An AUC of 1 corresponds to a perfect classification rate, while an AUC of 0.5 equals random guessing. The color definitions are analogously to figure 4.1.

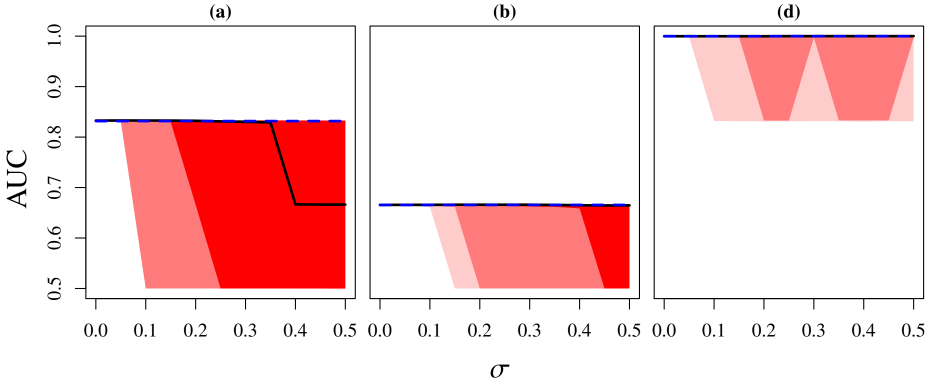


Figure 4.9: This figure shows the AUC of the scenarios (a), (b) and (d) (scenario (c) was omitted in the multinomial case) as a function of σ. An AUC of 1 implies that the regression is capable to identify the *true* coefficients and an AUC of 0.5 equals random guessing. The color definitions are analogously to figure 4.1.

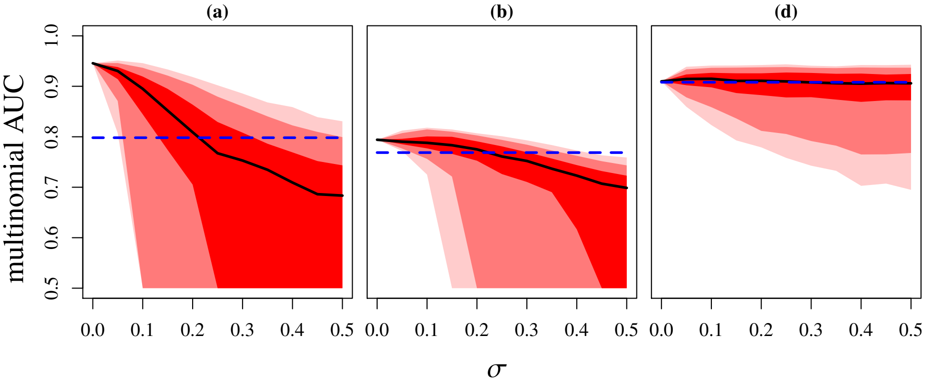


Figure 4.10: In this figure the multinomial AUC on the test data set is shown for the scenarios (a), (b) and (d) (scenario (c) was omitted in the multinomial case) as a function of σ. An AUC of 1 corresponds to a perfect classification rate, while an AUC of 0.5 equals random guessing. The color definitions are analogously to figure 4.1.

Scenario (d) corresponds to the optimal case for zero-sum regression. However, normal multinomial regression also identifies the scale invariant model and hence performed almost equally well.
These scenarios show that the equal-sum constraint for multinomial regressions results in scale invariant models.

4.4 Cox Proportional Hazard Regression

In this section, the same scenarios as in section 4.1 are used for analyzing the influence of sample-wise shifts on Cox models. However, the amount of training samples has been increased to 30.
In order to simulate a response, the hazard function (2.63) was used with a shared baseline hazard $h_0(t) = 1$. Censoring was omitted. This kind of simulation is insufficient for simulating the characteristics of survival data. Therefore, more sophisticated methods, like proposed by Austin [6], would be necessary to imitate the distribution of survival data. However, Cox regression only relies on the ordering of the events and the intention of this simulation is only to demonstrate the difference between normal and zero-sum Cox regression.
Cox regression is not designed for making predictions and is mainly used for identifying survival associated features. The important aspects of Cox models are thus the accuracy of the model and the feature selection property. This is however closely linked to the generalizability of the obtained results. For that reason, the partial log-likelihood (2.66) is evaluated on the training data as well as on test data and is shown in figure 4.11 and 4.13. Moreover, the AUC of the feature selection is computed as in the last sections and is shown in figure 4.12.

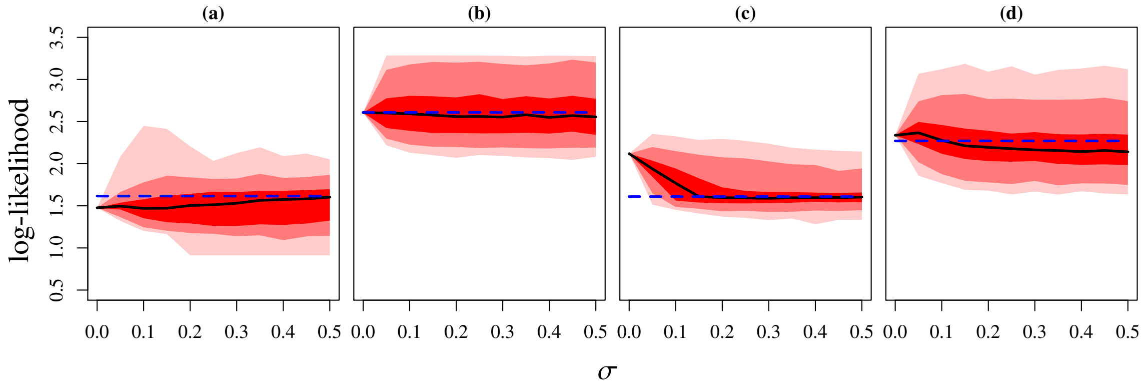


Figure 4.11: This figure shows the partial log-likelihood on the training data of the scenarios (a)-(d) as a function of σ. The color definitions are analogously to figure 4.1.

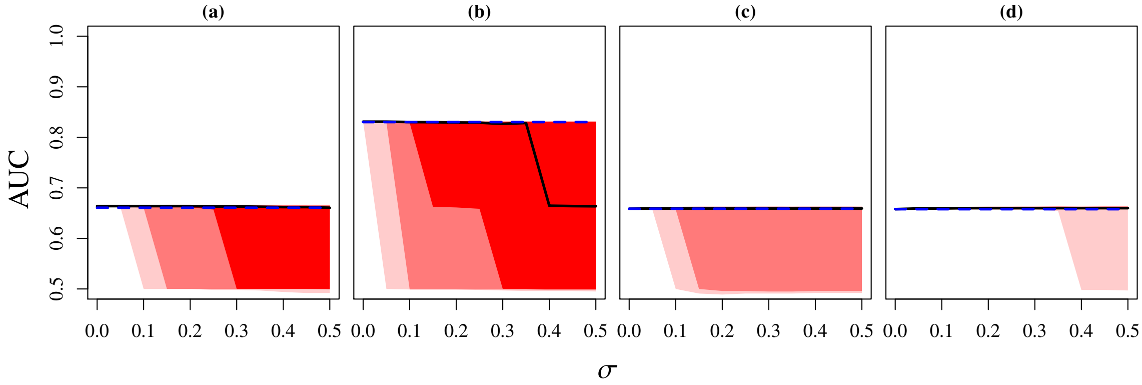


Figure 4.12: This figure shows the *area under the receiver operating characteristic curve* (AUC) of the scenarios (a)-(d) as a function of σ. An AUC of 1 implies that the regression is capable to identify the *true* coefficients and an AUC of 0.5 equals random guessing. The color definitions are analogously to figure 4.1.

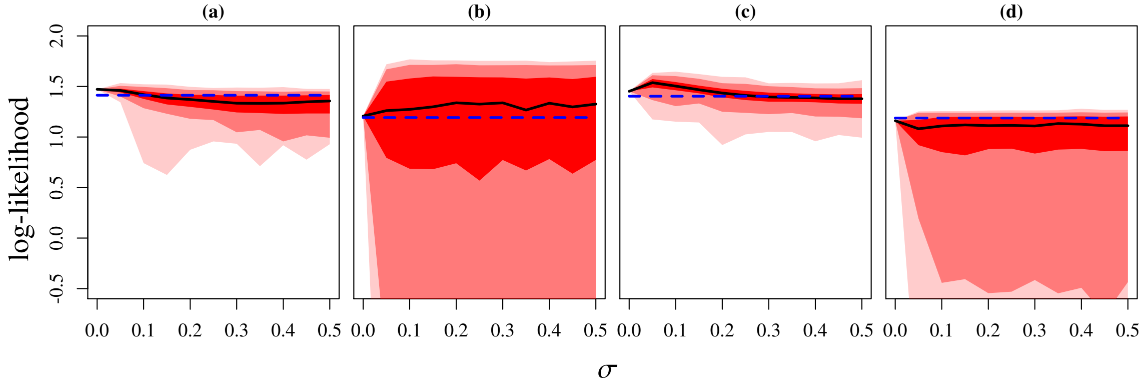


Figure 4.13: This figure shows the partial log-likelihood on the test data of the scenarios (a)-(d) as a function of σ. The color definitions are analogously to figure 4.1.

As expected, zero-sum Cox regression yields consistent results independent of the size of the sample-wise shifts. The median log-likelihood on the training data of zero-sum Cox regression is – despite for low σ – overall higher (better) than the normal Cox regression.

The capability of identifying the *true* coefficients of normal Cox regression is impaired by the data shifts. Only scenario (d) is unaffected. The partial log-likelihood on the training and test data of normal Cox regression is affected by the data shifts. However, normal Cox regression sometimes accomplishes to perform better than zero-sum Cox regression. This is clearly visible in scenario (b).

Overall, zero-sum Cox regression yields consistent results and is better suited for detecting the *true* coefficients when data shifts are present.

5 Application of Zero-Sum Regression on Omics Data

In this chapter linear, logistic, multinomial and Cox regression in combination with the lasso and the zero-sum constraint are applied on omics data sets and are discussed in the corresponding sections. Moreover, the effects of the additional generalized lasso regularization on zero-sum regression is shown in an additional section.
At first, a microbiome data set is analyzed using linear regression in order to identify indole producing bacteria. Afterwards, logistic regressions are performed on NMR metabolite spectra to distinguish between two different disease types. Subsequently, the same data set is analyzed using logistic regression in combination with the generalized lasso regularization. This allows to incorporate knowledge about the ordering of the features to obtain more robust models. Next, a methylation data set is analyzed using zero-sum multinomial regression to differentiate between tumor and metastatic cells using additional methylation data of the surrounding tissue to mitigate the effects of tissue background contamination. The last data set covers gene expression data and survival data of lymphoma patients and is evaluated using Cox regression.

5.1 Application of Zero-Sum Linear Regression on Microbiome Data

In this section linear zero-sum regression is applied to an intestinal microbiome data set to identify bacterial communities, which are associated with the *indole* concentration in patients. Although this is not an omics data set per se, the data is generated using the same high-throughput techniques that are used in omics research and is therefore prone to the same scaling issues. This application was already published in [3] and has been repeated for this thesis with the latest version (1.1.1) of the *zeroSum* software.
The data was generated from patients which have received a bone marrow transplantation and was provided by D. Weber, E. Holler of the Department of Hematology (University Hospital Regensburg), A. Hiergeist, A. Gessner of the Institute of Clinical Microbiology and Hygiene (University Hospital Regensburg) and K. Dettmer, P. J. Oefner of the Institute of Functional Genomics (University of Regensburg). The preprocessing was performed by F. Stämmler of the Institute of Functional Genomics (University of Regensburg).
Patients undergoing a bone marrow transplantation are at risk of developing an *acute graft versus host disease*, where the transplanted immune cells attack the cells of the recipient [24]. Since this complication is associated with the intestinal microbiome [35, 75] and especially with the absence of *indole* producing bacteria [83], the concentration of the related metabolite *3-indoxyl sulfate* (3-IS) in the urine of the patients has been additionally determined using liquid chromatography/tandem mass spectrometry. The metabolite data was normalized using the concentration of creatine as a reference [81]. The microbiome composition has been measured by sequencing the hypervariable V3 region of the 16S ribosomal RNA gene and mapping the sequences to operational taxonomic units (OTUs). Furthermore, three exogenous bacteria (*Salinibacter ruber*, *Rhizobium radiobacter*, and *Alicyclobacillus acidiphilus*) were spiked into the samples to create an external reference.

The reasoning behind this experimental design is to identify the *indole* producing bacteria in order to treat such patients only with antibiotics which have no effect on these bacteria. For this purpose, zero-sum regression is especially suited, since only the relative number of bacteria is measurable, but the absolute number of bacteria is affecting the 3-IS concentration and therefore the *indole* concentration. Moreover, the scale of the microbiome data as well as the scale of the 3-IS metabolite concentration can be affected by diet and treatments. Therefore, linear zero-sum regression is applied to predict the 3-IS levels using the microbiome composition. Furthermore, the feature selection property of the lasso regularization is used to identify the bacteria which are associated with the *indole* concentration.

The data details the composition of 160 different bacterial genera for 37 patients and the corresponding 3-IS levels. In order to demonstrate the properties of zero-sum regression, the data was normalized in two different ways. First, the data set was mean-centered assuming that the total amount of bacteria should be equal. This procedure corresponds to the standard approach used for microbiome data. Second, the data was normalized so that the mean of the external references is equal. This approach is known as spike-in calibration and is an attempt to determine the absolute abundances of the bacteria [71].

Both data sets were subsequently $\log_2$ transformed and were analyzed with normal linear regression and zero-sum linear regression. The regularization parameter λ was optimized using a 10-fold cross-validation. The corresponding cross-validation mean squared error (CV MSE) is shown in figure 5.1.

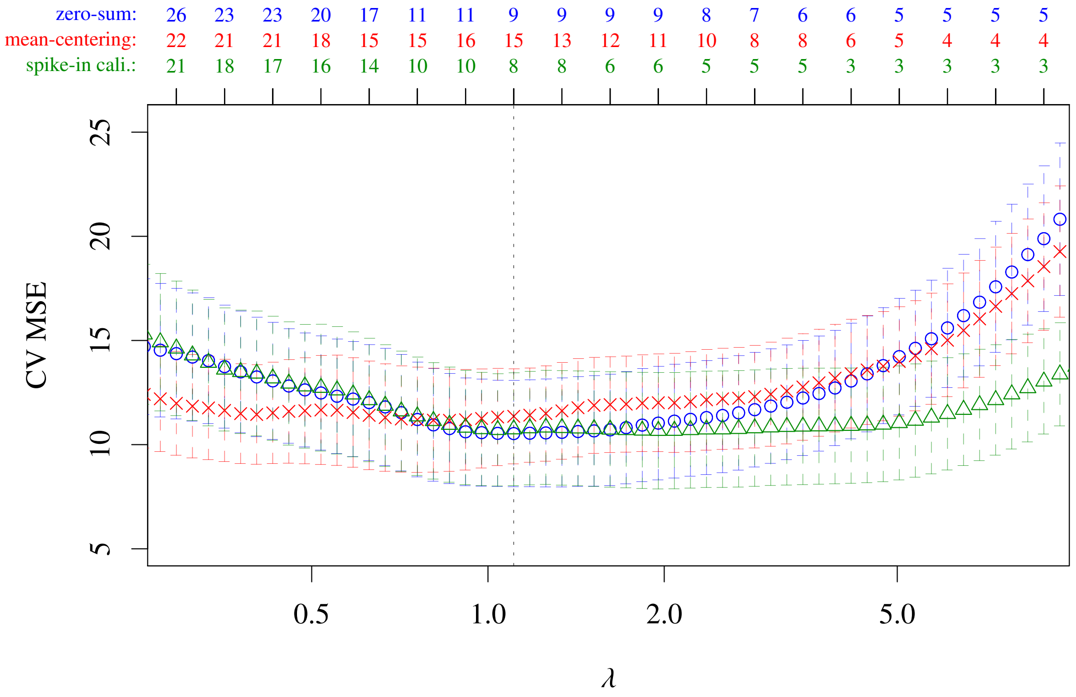


Figure 5.1: Shown is the cross-validation mean squared error (CV MSE) as a function of the regularization parameter λ. Green denotes the CV MSE of normal linear regression applied on the spike-in normalized data, while red denotes for CV MSE of normal linear regression applied on the mean-centered data. Since zero-sum regression is scale invariant, the same CV MSE error is obtained for both normalizations and is shown in blue. The dotted vertical line indicates the minimum of the cross-validation error obtained by zero-sum regression. At the top, the number of non-zero coefficients is shown.

	spike-in cali.	mean-centering	zero-sum
CV MSE for λ_{1SE}	13.38	13.58	13.07
CV MSE for λ_{min}	10.67	11.17	10.53

Table 5.1: This table details the cross-validation mean squared error (CV MSE) for λ_{1SE} and λ_{min} of the normal linear regressions applied to the spike-in calibrated and mean-centered data as well as the CV MSE of zero-sum regression.

As it can be seen in both, figure 5.1 and table 5.1, the lowest CV MSE is obtained by zero-sum regression. However, this cross-validation was used for optimizing the parameter λ and thus an additional *outer* cross-validation has to be performed to determine the predictive power. Each fold of this *outer* cross-validation performs an additional *inner* cross-validation to determine a suitable λ. Subsequently, this value of λ is used to learn a model on all samples of this fold, which is then applied to determine the mean squared error of the left-out samples. This procedure has to be iterated over all folds and will be referred to in the following as nested cross-validation (NCV).

In order to estimate the predictive power a NCV, which uses 10 folds for predicting the performance as well as 10 folds for determining the optimal value for λ, has been applied. However, the NCV depends on the randomly selected folds and has been repeated 10 times. The NCV errors of each repetition and the corresponding mean are shown in table 5.2.

repetition	NCV for spike-in cali.	NCV for mean-centering	NCV for zero-sum
1	11.98	15.58	11.53
2	12.85	15.09	11.65
3	13.28	14.15	11.42
4	12.03	14.76	12.07
5	12.70	14.16	11.35
6	12.09	14.42	11.41
7	12.13	14.38	11.62
8	12.34	15.20	11.84
9	12.69	14.82	11.22
10	12.35	15.83	12.04
mean	12.44	14.84	11.61

Table 5.2: This table details the obtained nested cross-validation (NCV) errors and the corresponding mean of the 10 repetitions for the normal linear regressions on the spike-in calibrated and mean-centered data as well as for zero-sum regression.

It can be seen that zero-sum regression has almost always a lower NCV error for each repetition than each of the normal linear regressions. Consequently, the mean NCV error of the zero-sum regressions is also lower than the mean NCV error of the normal linear regressions. A paired t-test comparing the NCV errors of the normal linear regressions with the NCV errors of zero-sum regression yields p-values lower than 0.01.

Note that the NCV is only based on one data set and therefore every observation may be affected by the same systematic scale deviation. Hence, the real benefit of zero-sum models cannot be seen to such an extend and would only be revealed by applying the models to completely independent data. Nevertheless, zero-sum regression performs best on this microbiome data set.

In order to demonstrate the scale invariance, normal linear regression as well as zero-sum regression have been additionally applied on only $\log_2$ transformed data (without normalization) and the coefficients have been determined. The scale invariance property of zero-sum regression can be seen in the Venn diagrams (figure 5.2) as well as in the actual coefficients (table 5.3).

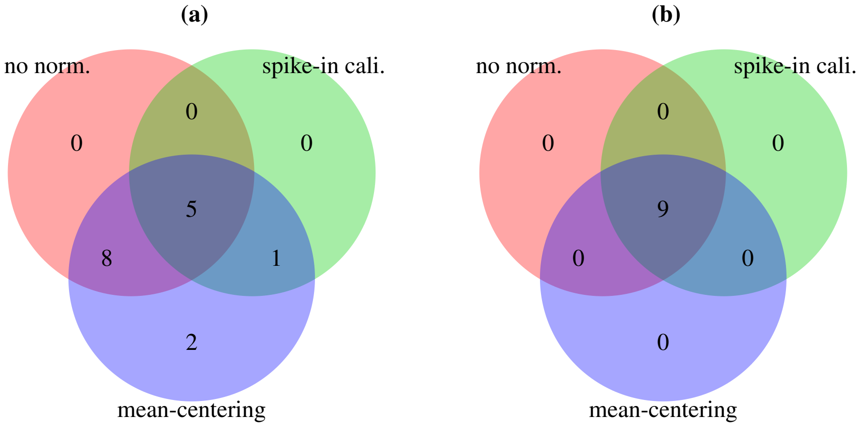


Figure 5.2: Shown is the coefficient overlap of the regressions applied on the raw data and on the two differently normalized data sets. (a) shows the coefficient overlap obtained using normal linear regressions, while (b) shows the coefficient overlap obtained using zero-sum regression.

Due to the scale invariance, zero-sum regression chooses the exact same coefficients, while normal linear regression selects different features. An interesting property of zero-sum regression is, that it selects one of the external spiked-in bacteria *Alicyclobacillus* with a high negative value. It can be assumed that in this way the external reference is automatically detected by zero-sum regression and used as a counterbalance.

bacteria communities	normal no norm.	normal spike-in cali.	normal mean-centering	zero-sum no norm.	zero-sum spike-in cali.	zero-sum mean-centering
Intercept	-9.8274	-7.1387	-11.0776	-1.4704	-1.4704	-1.4704
Alicyclobacillaceae: Alicyclobacillus	0.0000	0.0000	0.0000	-0.7346	-0.7346	-0.7346
Actinomycetaceae: Actinomyces	0.3068	0.2001	0.3263	0.1954	0.1954	0.1954
Bifidobacteriaceae: Bifidobacterium	0.0343	0.0000	0.0205	0.0000	0.0000	0.0000
Coriobacteriaceae: Paraeggerthella	0.0000	0.0000	0.0000	-0.0058	-0.0058	-0.0058
Bacteroidaceae: Bacteroides	0.2957	0.1653	0.3159	0.1893	0.1893	0.1893
Staphylococcaceae: Staphylococcus	-0.0923	-0.0516	-0.1902	-0.3225	-0.3225	-0.3225
Enterococcaceae: Enterococcus	0.0787	0.0000	0.0691	0.0000	0.0000	0.0000
Lactobacillaceae: Lactobacillus	0.0556	0.0000	0.0930	0.0639	0.0639	0.0639
Streptococcaceae: Streptococcus	0.0000	0.0000	0.0164	0.0000	0.0000	0.0000
Lachnospiraceae: Anaerostipes	0.1678	0.0000	0.1958	0.0000	0.0000	0.0000
Lachnospiraceae: Coprococcus	0.2365	0.0000	0.3858	0.0000	0.0000	0.0000
Lachnospiraceae: Incertae Sedis	0.3435	0.3864	0.3647	0.3275	0.3275	0.3275
Lachnospiraceae: uncultured	0.0460	0.0000	0.0683	0.0000	0.0000	0.0000
Ruminococcaceae: Faecalibacterium	0.0000	0.0452	0.0287	0.0656	0.0656	0.0656
Ruminococcaceae: Incertae Sedis	0.0000	0.0000	0.0065	0.0000	0.0000	0.0000
Ruminococcaceae: Subdoligranulum	0.2272	0.2266	0.2233	0.2212	0.2212	0.2212
Ruminococcaceae: uncultured	0.0972	0.0000	0.1826	0.0000	0.0000	0.0000
Verrucomicrobiaceae: Akkermansia	0.0345	0.0000	0.0848	0.0000	0.0000	0.0000

Table 5.3: Shown are the non-zero coefficients of the bacteria communities selected by the normal linear regression and zero-sum regression applied on the raw, spike-in calibrated and mean-centered data.

Concluding, it can be stated that zero-sum regression improves the feature selection as well as the predictive performance and is independent of the applied normalization.

5.2 Application of Zero-Sum Logistic Regression on NMR Metabolomics Data

In this section, the advantages of zero-sum logistic regression are demonstrated on two metabolomics data sets. These data sets have been generated using ^{1}H-NMR spectroscopy of urine samples and of blood plasma samples. The samples have been collected from patients of the University Clinic of Erlangen 24 h after a cardiac surgery with a cardiopulmonary bypass and have been analyzed in [86, 87]. Measuring and preprocessing has been performed by H. U. Zacharias, M. Altenbuchinger, W. Gronwald of the Institute of Functional Genomics (University of Regensburg). Moreover, this data set has been used for demonstrating zero-sum logistic regression in [88]. Parts of this publication have been repeated for this thesis using the latest version of the *zeroSum* software (1.1.1) and are presented in this section.

The urine data consists of 106 patients, while the blood plasma data consists of 85 patients and is a subset of the former one. The aim of this experiment is to identify a biomarker which detects an acute kidney injury (AKI), which is a possible complication of the surgery, as early as possible. Such a complication has been diagnosed 48 h after the surgery on 34 patients of the urine data set and on 33 patients of the blood plasma data set. The NMR spectra measured using the urine samples will be referred to in the following as urinary AKI data, while the NMR spectra measured using the blood plasma samples will be referred to as plasma AKI data.

In order to make the data accessible by generalized linear models, the NMR spectra have been partitioned into 712 small sections called *bins*. The spectral integrals of the *bins* are then used as features. However, normalization methods have to be applied to make the data comparable. For that reason, the urinary AKI data set has been provided normalized in four different ways.

The first normalization uses creatine as a reference metabolite, the second uses the total spectral area as reference, the third uses the NRM reference compound trimethylsilylpropanoic (TSP) and the last normalization is a method called probabilistic quotient normalization (PQN) [22]. The creatine normalization is only reasonable for spectra of urine samples and has been omitted for the plasma AKI data.

Each data set has been $\log_2$ transformed and analyzed using logistic regression as well as zero-sum logistic regression. Additionally, the lasso regularization has been applied. The resulting cross-validation (CV) log-likelihood is shown as a function of the regularization parameter λ in figure 5.3 for the urinary AKI data and in figure 5.4 for the plasma AKI data. The highest obtained CV log-likelihood for each data set and normalization is detailed in table 5.4 for the urinary AKI data and in table 5.5 for the plasma AKI data.

Furthermore, the cross-validation *area under the receiver operating characteristic curve* (AUC) has been computed for $\lambda_{1\text{SE}}$ and $\lambda_{\min}$ and is shown in table 5.6 for the urinary AKI data and in table 5.7 for the plasma AKI data.

It can be seen that both, logistic and zero-sum logistic regression, performed equally well on the urinary AKI data. Only logistic regression on the creatine and the TSP normalized data performed slightly worse. On the plasma AKI data set, zero-sum logistic regression yields sightly better models than the logistic regression.

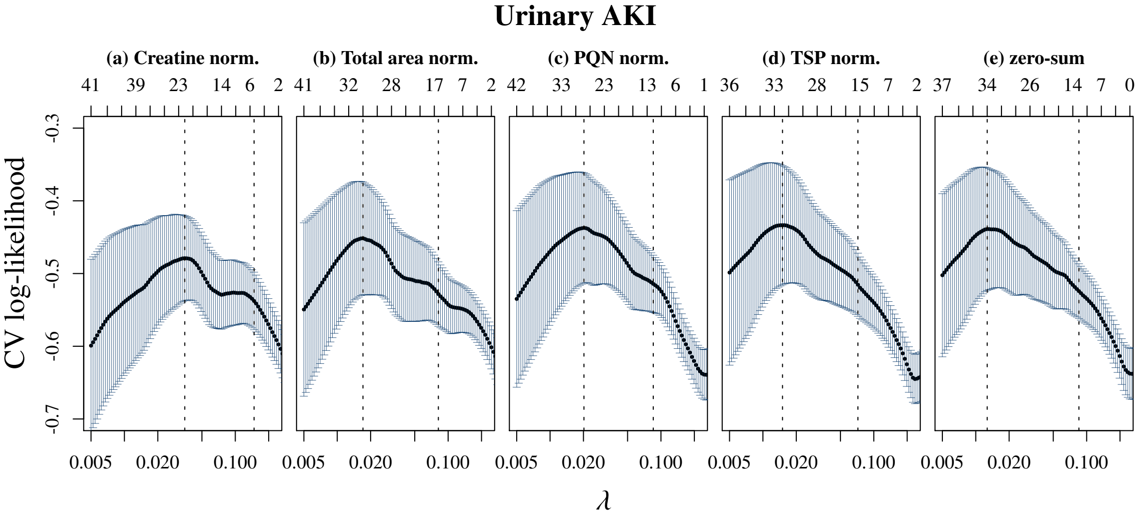


Figure 5.3: Shown is the cross-validation (CV) log-likelihood of logistic and zero-sum logistic regression applied on the urinary AKI data set as a function of λ. (a) shows the CV log-likelihood of logistic regression for the creatinine normalized data set, (b) for the total area normalized data set, (c) for the PQN normalized data set and (d) for the TSP normalized data set. (e) shows the CV log-likelihood of zero-sum logistic regression. Since zero-sum logistic regression resulted in the same CV log-likelihood for each normalization only one figure is shown. At the top, the number of non-zero coefficients is shown.

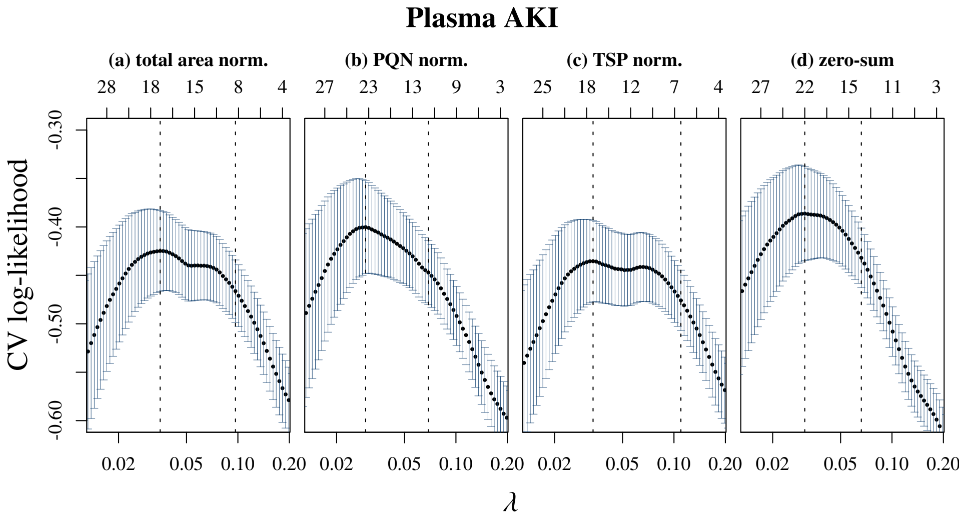


Figure 5.4: Shown is the cross-validation (CV) log-likelihood of logistic and zero-sum logistic regression applied on the plasma AKI data set as a function of λ. (a) shows the CV log-likelihood of logistic regression for the total area normalized data set, (b) for the PQN normalized data set, (c) for the TSP normalized data set. (d) shows the CV log-likelihood of zero-sum logistic regression. Since zero-sum logistic regression resulted in the same CV log-likelihood for each normalization only one figure is shown. At the top, the number of non-zero coefficients is shown.

CV log-likelihood – urinary AKI

selected λ	creatine norm.	total area norm.	PQN norm.	TSP norm.	zero-sum
λ_{1SE}	-0.5377	-0.5282	-0.5142	-0.5154	-0.5243
$\lambda_{\min}$	-0.4789	-0.4515	-0.4372	-0.4335	-0.4388

Table 5.4: Shown is the cross-validation (CV) log-likelihood of logistic and zero-sum logistic regression applied on the urinary AKI data set for λ_{1SE} and $\lambda_{\min}$. Since zero-sum logistic regression resulted in the same CV log-likelihood for each normalization only one value is shown.

CV log-likelihood – plasma AKI

selected λ	total area norm.	PQN norm.	TSP norm.	zero-sum
λ_{1SE}	-0.4663	-0.4471	-0.4759	-0.4319
$\lambda_{\min}$	-0.4248	-0.4002	-0.4354	-0.3862

Table 5.5: Shown is the cross-validation (CV) log-likelihood of logistic and zero-sum logistic regression applied on the plasma AKI data set for λ_{1SE} and $\lambda_{\min}$. Since zero-sum logistic regression resulted in the same CV log-likelihood for each normalization only one value is shown.

CV AUC – urinary AKI

selected λ	creatine norm.	total area norm.	PQN norm.	TSP norm.	zero-sum
λ_{1SE}	0.7717	0.7672	0.7904	0.7778	0.7717
$\lambda_{\min}$	0.8268	0.8595	0.8533	0.8636	0.8644

Table 5.6: Shown is the obtained cross-validation (CV) *area under the receiver operating characteristic curve* (AUC) for λ_{1SE} and $\lambda_{\min}$ of logistic and zero-sum logistic regression applied on the urinary AKI data set. Since zero-sum logistic regression resulted in the same CV AUC for each normalization only one value is shown.

CV AUC – plasma AKI

selected λ	total area norm.	PQN norm.	TSP norm.	zero-sum
λ_{1SE}	0.8706	0.8683	0.8601	0.8922
$\lambda_{\min}$	0.8811	0.8922	0.8776	0.8998

Table 5.7: Shown is the obtained cross-validation (CV) *area under the receiver operating characteristic curve* (AUC) for λ_{1SE} and $\lambda_{\min}$ of logistic and zero-sum logistic regression applied on the plasma AKI data set. Since zero-sum logistic regression resulted in the same CV AUC for each normalization only one value is shown.

In order to assess the predictive power of the different approaches, a nested cross-validation (NCV), as described in the previous section, has been performed. The obtained NCV AUCs of each repetition and the corresponding means are shown in table 5.8 for the urinary AKI data set and in table 5.9 for the plasma AKI data set.

It can be seen that, the predictive power of zero-sum logistic regression is slightly exceeded by logistic regression applied on the PQN normalized data. A paired t-test comparing PQN with all other results yields p-values lower than 0.001. Hence, it can be stated that the PQN normalization seems to be well suited for this data set.

On the plasma AKI data, zero-sum logistic regression performs better than all other methods and yields the highest NCV AUC. A paired t-test resulted in a p-value lower than 0.001.

Nested cross-validation AUC – urinary AKI

repetition	creatine norm.	total area norm.	PQN norm.	TSP norm.	zero-sum
1	0.7806	0.7900	0.8288	0.7635	0.8076
2	0.7594	0.7839	0.7925	0.7998	0.7749
3	0.8051	0.7549	0.8268	0.7574	0.7704
4	0.7753	0.7606	0.8141	0.7610	0.7753
5	0.7631	0.7892	0.8125	0.7435	0.7688
6	0.7896	0.7953	0.8284	0.7672	0.7884
7	0.7480	0.7655	0.7700	0.7537	0.7574
8	0.7827	0.7712	0.8015	0.7059	0.7831
9	0.8043	0.7896	0.8039	0.7316	0.7962
10	0.7761	0.7475	0.7798	0.6638	0.7717
mean	0.7784	0.7748	0.8058	0.7447	0.7794

Table 5.8: This table details the nested cross-validation (NCV) *area under the receiver operating characteristic curve* (AUC) on the urinary AKI data.

Nested cross-validation AUC – plasma AKI

repetition	total area norm.	PQN norm.	TSP norm.	zero-sum
1	0.8462	0.8642	0.8094	0.8881
2	0.8747	0.8840	0.8596	0.8928
3	0.8584	0.8561	0.6754	0.8753
4	0.8269	0.8252	0.7069	0.8473
5	0.8508	0.8642	0.7756	0.8776
6	0.8677	0.8823	0.7797	0.8887
7	0.8706	0.8508	0.7797	0.8770
8	0.8607	0.8735	0.7314	0.8928
9	0.8275	0.8561	0.7488	0.8794
10	0.8555	0.8438	0.7908	0.8619
mean	0.8539	0.8600	0.7657	0.8781

Table 5.9: This table details the nested cross-validation (NCV) *area under the receiver operating characteristic curve* (AUC) on the plasma AKI data.

As it can be seen in the Venn diagrams in figure 5.5 for urinary AKI data set and in figure 5.6 for the plasma AKI data set, zero-sum logistic regression selects the same coefficients, while logistic regression depends on the applied normalization. The coefficients are shown in the appendix B.1 for the urinary AKI and in the appendix B.2 for plasma AKI.

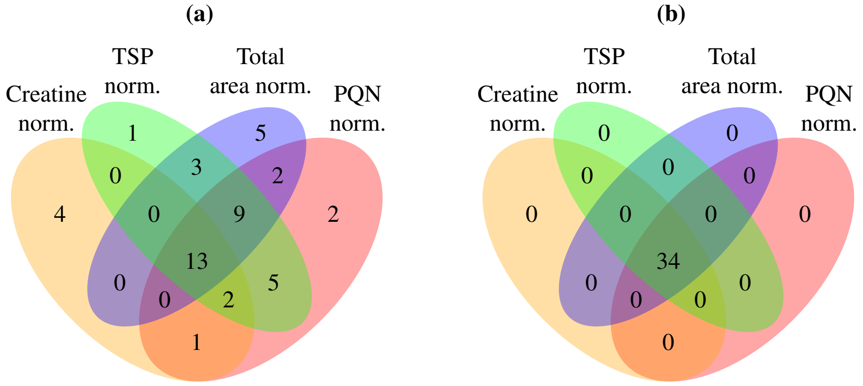


Figure 5.5: Shown is the overlap of the coefficients of logistic and zero-sum logistic regression on the differently normalized urinary AKI data. (a) shows the overlap of the coefficients obtained using logistic regressions, while (b) shows the overlap of the coefficients obtained using zero-sum logistic regression.

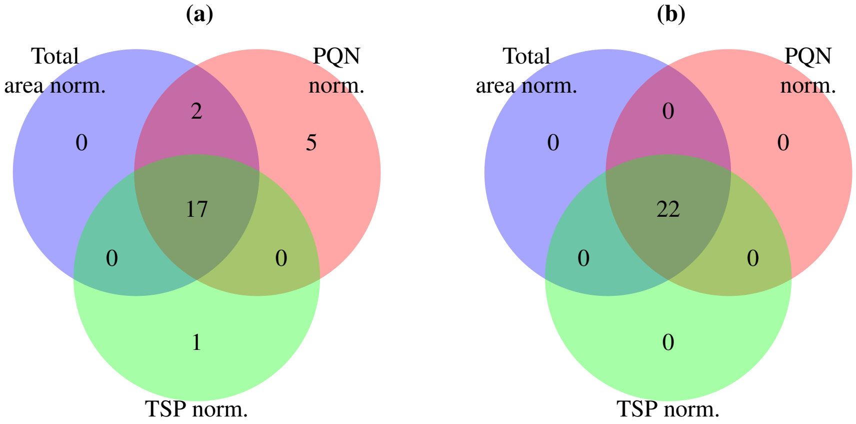


Figure 5.6: Shown is the overlap of the coefficients of logistic and zero-sum logistic regression on the differently normalized plasma AKI data. (a) shows the overlap of the coefficients obtained using logistic regressions, while (b) shows the overlap of the coefficients obtained using zero-sum logistic regression.

In summary, zero-sum regression only yielded an increased predictivity on the plasma AKI data. Moreover, it was shown that the feature selection property of zero-sum regression does not depend on the applied normalization.

5.3 Application of Zero-Sum Fused Logistic Regression on NMR Metabolomics Data

In this section the generalized lasso regularization is applied to mitigate the problem of displaced signals in NMR spectra. These displacements can be caused by varying pH levels, salt concentration or temperature differences and also depend on specific metabolites. Hence, different signals in the same spectra can be differently shifted.

The common approach to compensate for these shifts is to choose a large enough *bin* size. However, the optimal *bin* size is not known *a priori* and depends on the data. Thus, equally-sized *bins* cause that metabolites with small displacements are included into too large *bins*, meaning that signals can get superimposed by surrounding signals. This issue can be resolved not only by using adaptive *binning* procedures [5, 21], but also by using the generalized lasso [85].

The fused lasso is a special case of the generalized lasso and is achieved by using a regularization matrix F of the form

$$F = \begin{pmatrix} 1 & -1 & 0 & 0 & \dots & 0 \\ 0 & 1 & -1 & 0 & \dots & 0 \\ 0 & 0 & 1 & -1 & \dots & 0 \\ & & \vdots & & & \\ 0 & 0 & 0 & \dots & 1 & -1 \end{pmatrix}. \tag{5.1}$$

This regularization forces adjacent coefficients to be equal.

To demonstrate the adaptive *binning* property, the plasma AKI data set of the last section is used but with a ten times smaller *bin* size. In total, the data consists of 7130 *bins* for 85 samples. During the preprocessing, specific noisy regions of the spectra have been removed. The corresponding rows of F have been omitted. This data was then $\log_2$ transformed and analyzed using zero-sum fused logistic regression with the lasso regularization.

Note that, despite using the same fold partitioning of the cross-validation as used in the previous section, the results are not fully comparable. The reason for this is that the logarithm of a wide *bin* is not the same as the sum of the logarithms of the corresponding *sub-bins*. Moreover, this approach uses the local search implementation, which is more likely to result in only near optimal solutions.

The cost function of the fused logistic regression consists of two hyper-parameters γ and λ, which cause that the search-space is two dimensional. A suitable γ sequence has been determined by manually identifying a value of γ which is high enough to only obtain the trivial solution (every coefficient except of the intercept equals zero). Starting from this value, a logarithmic declining sequence has been calculated. The λ sequence has been determined as detailed in section 3.3. Subsequently, a basic grid search is performed. The resulting cross-validation error is shown as a heatmap in figure 5.7. Additionally, a logistic zero-sum regression (without fusion) has been applied. The resulting cross-validation log-likelihood is shown in the bottom row of the heatmap. The highest CV log-likelihood, which corresponds to the best result, is marked in orange.

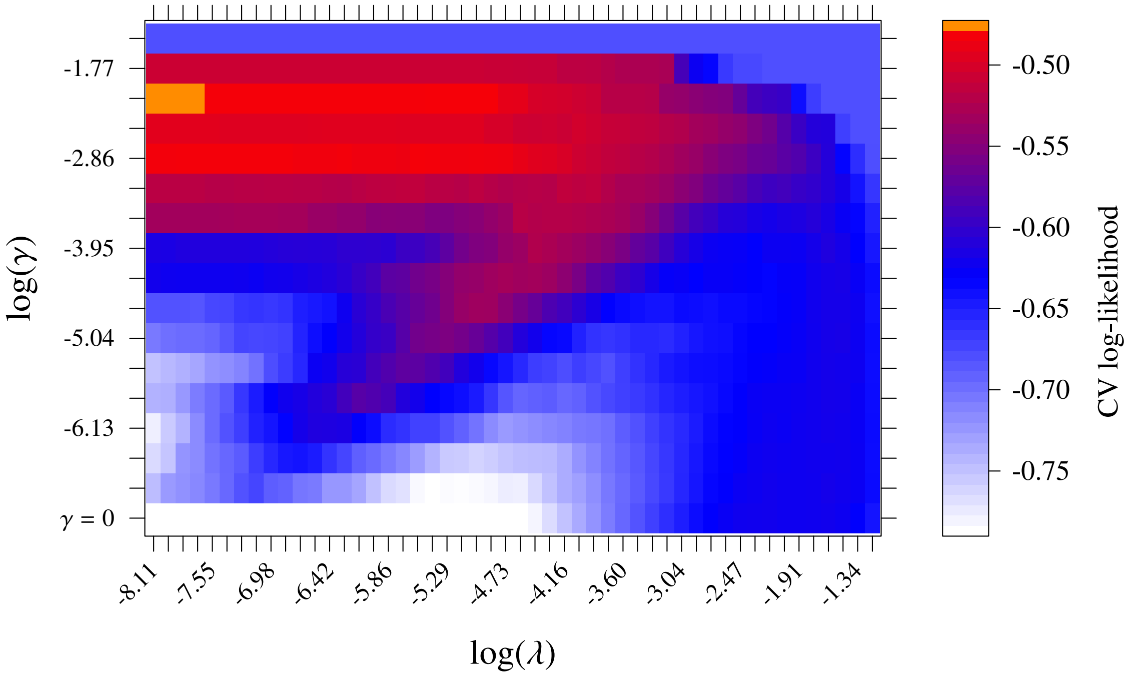


Figure 5.7: This heatmap shows the obtained cross-validation (CV) log-likelihood of zero-sum fused logistic regression applied on the plasma AKI data set as a function of λ and γ. The lowest row shows the CV log-likelihood obtained by a zero-sum logistic regression without the fused lasso regularization ($\gamma = 0$). The highest obtained CV log likelihood is shown in orange.

It can be seen that logistic regression without fusion applied on the smaller *bins* performs worse (CV log-likelihood of ≈-0.6, AUC ≈0.69) than logistic regression applied on the wider *bins* used in the last section (CV log-likelihood of ≈-0.4, AUC of ≈0.90). By adding the fused lasso regularization the accuracy of the predictions significantly increases (CV log-likelihood ≈-0.47, AUC of ≈0.86).
The coefficients of the best solution are almost always non-zero (6715 of 7130). Hence, further lowering λ does not further increase the cross-validation log-likelihood, since the lasso regularization is to small to have any effect. The coefficients, which have a larger absolute value than 0.001, are shown in the appendix B.3. It can be seen that adjacent coefficients get assigned the same value, which corresponds to locally wider bins.
The main drawback, however, is, that the local search algorithm for this analysis required a runtime of roughly a week on a 28 core workstation (for hardware informations see *rhskl3* in the appendix B.5). In contrast, the runtime of the logistic regression using coordinate descent (shown in the bottom row of the heatmap) required less then 10 minutes. Note that the local search implementation is only a proof of concept and that more sophisticated approaches could reduce the required computing time.
In summary, it can be stated that zero-sum regression in combination with the generalized lasso is capable of using additional information about the characteristics of the data to improve the models. This advantage, however, comes in exchange for an increased computational effort.

5.4 Application of Zero-Sum Multinomial Regression on DNA Methylation Data

This section shows how zero-sum multinomial regression can be used to mitigate the effects of a contamination of biopsies with surrounding tissue on classifiers. Such contaminations can distort molecular profiles and therefore can mislead machine learning algorithms by using features which are specific for the surrounding tissue. To mitigate this problem, data of the surrounding tissue is used as additional classes in a multinomial regression with adjusted weights and a modified cross-validation to enforce a feature selection which is more specific for tumor and metastatic tissue.

To exemplify this approach, DNA methylation data of primary tumors, metastasis and the surrounding lung and breast tissue will be used. This data was provided by the DFG research group FOR2127 (Coordinator Prof. Dr. C. Klein, Experimental Medicine and Therapy Research, University of Regensburg) and was generated using BALB-NeuT mice. These mice have been genetically altered to develop a cancer which is equivalent to HER2-positive human breast cancer and which is therefore used as a model for this cancer type [37]. From these mice, tissue samples of tumors, metastases and surrounding breast and lung tissue have been extracted. Subsequently, DNA methylation profiles of these samples have been measured by the group of Prof. Dr. M. Rehli of the Department of Internal Medicine III (University Hospital Regensburg). The preprocessing and normalization to an artificially fully methylated reference has been performed by M. Schwarz of the Institute of Functional Genomics (University of Regensburg).

In total, the data consists of methylation profiles of 40 primary tumors, 40 metastases, 2 profiles of lung tissue (tissue background of the metastases) and 2 profiles of breast tissue (tissue background of the primary tumors).

In order to identify features which are more specific for tumor and metastatic cells, all four tissue types are used as separate classes in a zero-sum multinomial regression. However, the number of background samples is smaller than the number of primary tumors and metastases. Thus, the weights of the background samples have been used to equalize the contribution to the multinomial log-likelihood (2.43). Moreover, the cross-validation has been modified so that each fold contains all background samples with adjusted weights. This causes that the cross-validation log-likelihood is only determined for the differentiation between primary tumor and metastasis.

In order to compare this approach, a zero-sum logistic regression is performed using the same fold partitioning as the multinomial regression (without the samples of the surrounding tissue). For both methods, the lasso regularization has been applied and λ optimized using cross validation. The cross-validation log-likelihood is shown in figure 5.8. As x-axis the number of performed λ reduction steps is used, since the different structure of these regression problems causes that the value of λ differs and cannot be compared. Moreover, the cross-validation probabilities determined for $\lambda_{\min}$ are shown as box plots in figure 5.9. It can be seen that the cross-validation log-likelihood of the multinomial regression (-0.40) is higher, i.e. better, than the log-likelihood of the logistic regression (-0.46). Hence, the predicted probabilities of the multinomial regression models are also more accurate than the probabilities of the logistic regression models. Moreover, the AUC is increased from 0.86 to 0.91.

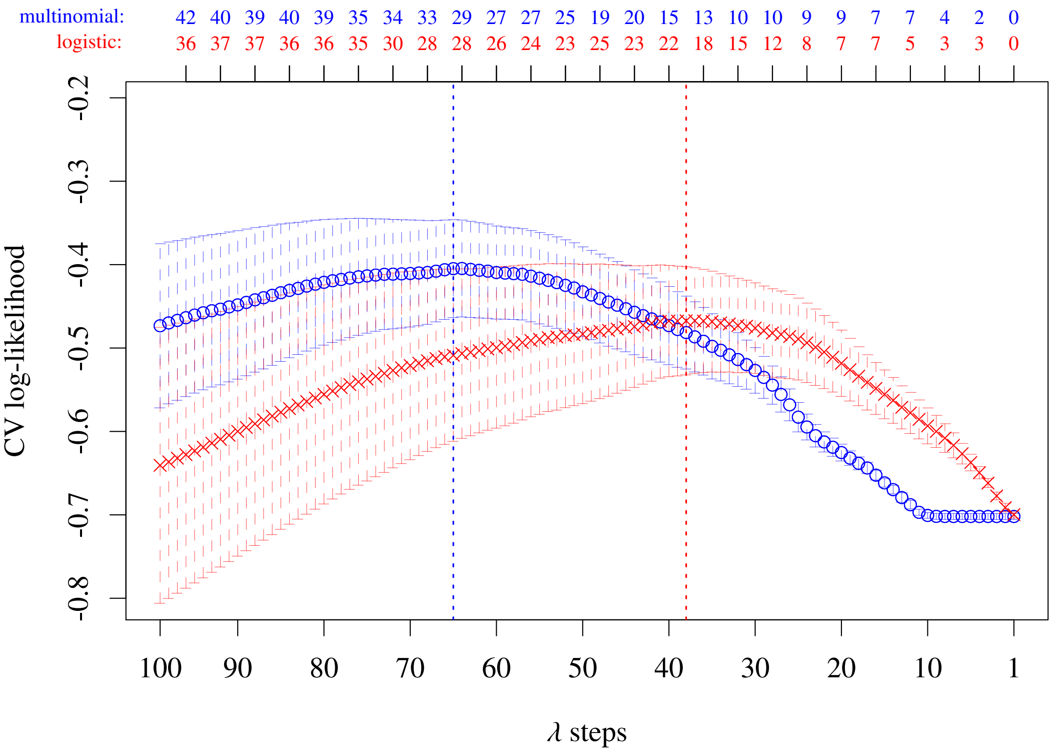


Figure 5.8: This figure shows the cross-validation log-likelihood as a function of the regularization parameter λ. The log-likelihood of the multinomial regression is shown in blue, while the log-likelihood of the logistic regression is shown in red. The x-axis denotes the number of reduction steps from $\lambda_{\max}$. At the top, the number of non-zero coefficients is shown.

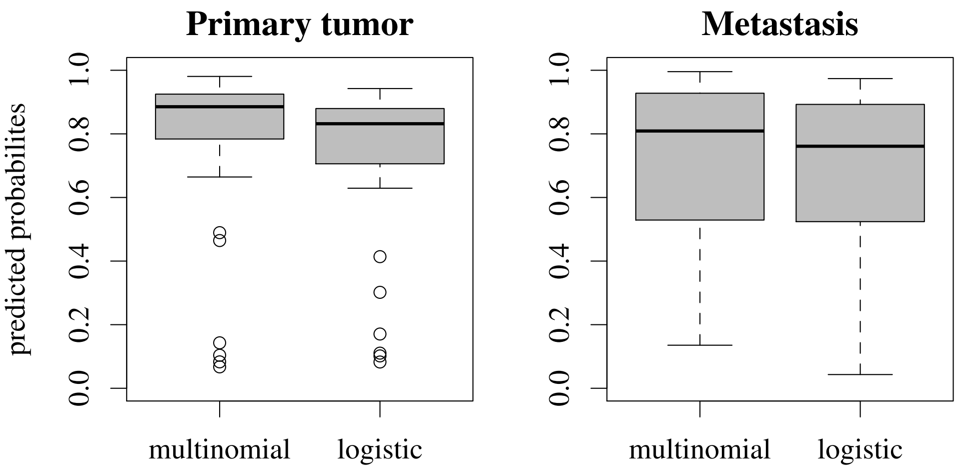


Figure 5.9: Shown is the distribution of the obtained cross-validation probabilities for primary tumor and metastasis determined using multinomial and logistic regression.

This increased prediction accuracy is unexpected at first, since the regression should be less capable to use tissue background information to distinguish between the classes. Therefore, this approach may even loose accuracy. Nevertheless, the multinomial regression seems to be able to identify more characteristic and reliable features due to the additional tissue background information. The reason for this probably is that logistic regression is more prone to overfitting and has to select a more sparse model (19 selected features), while multinomial regression is able to incorporate more features before overfitting occurs (29 selected feature). The selected features are shown in table 5.10.

chromosome–position	gene	multi. reg. metastasis	multi. reg. primary tumor	multi. reg. breast	multi. reg. lung	logistic reg. primary tumor
Intercept	–	1.3280	-0.4855	0.8321	0.2006	-1.3739
2 – 32741401	Sh2d3c	-2.1370	0.0000	0.0000	0.0000	0.8988
2 – 74698101	Hoxd9	1.6217	0.0000	0.0000	0.0000	-1.9365
2 – 105126801	Wt1	0.5677	0.0000	0.0000	0.0000	-0.3989
3 – 34650301	Sox2	0.0000	-0.5293	0.0000	0.0000	0.0000
3 – 55780501	Mab21l1	0.0000	4.9407	0.0000	0.0000	0.0000
3 – 55782901	Mab21l1	0.0000	0.0000	0.0000	0.0000	0.0746
4 – 99656601	Foxd3	0.3976	0.0000	0.0000	0.0000	-0.9856
4 – 99656701	Foxd3	1.0028	0.0000	0.0000	0.0000	0.0000
4 – 107683701	Glis1	0.0000	-0.1468	0.0000	0.0000	0.0000
6 – 39206801	Tbxas1	0.0000	0.0000	0.9705	0.0000	0.0000
6 – 52177201	5730596B20Rik	-0.8840	0.0000	0.0000	0.0000	0.1193
6 – 52177601	5730596B20Rik	-0.2459	0.0000	0.0000	0.0000	0.0000
7 – 19507301	Trappc6a	0.0000	-0.4284	0.0000	0.0000	-1.6949
7 – 19507401	Trappc6a	0.0000	-0.7309	0.0000	0.0000	0.0000
7 – 45638801	Rasip1	0.0000	0.0000	2.2852	0.0000	0.0000
7 – 83882501	Mesd	0.0000	0.0000	0.0000	0.0000	0.1454
7 – 83882601	Mesd	-0.5770	0.0000	0.0000	0.0000	0.0299
7 – 97400101	Ndufc2	0.0000	0.0000	0.0000	0.0000	0.5247
7 – 127824401	Stx4a	0.1786	0.0000	0.0000	0.0000	0.0000
8 – 71511601	Gtpbp3	0.0000	0.0000	0.0000	0.0000	0.5500
8 – 105268401	Fbxl8	0.0000	0.0000	0.0000	-0.3439	0.0000
8 – 105268501	Fbxl8	0.0000	0.0000	0.0000	0.0000	1.7981
8 – 105268601	Fbxl8	0.0000	0.5684	0.0000	0.0000	0.6496
9 – 121839201	Klhl40	0.0174	0.0000	0.0000	0.0000	-0.6315
9 – 121839301	Klhl40	0.0000	-0.6630	0.0000	0.0000	0.0000
10 – 3740601	Plekhg1	0.0000	0.0000	0.0000	3.7551	0.3946
10 – 75859901	Derl3	0.0000	-1.2742	0.0000	0.0000	0.0000
11 – 69560301	Trp53	1.3113	0.0000	0.0000	0.0000	0.0000
11 – 75795901	Ywhae	0.0000	0.0000	0.0000	0.0000	-0.1418
12 – 44221801	Pnpla8	0.0000	-0.1778	0.0000	0.0000	0.0000
13 – 55210101	Nsd1	0.0000	0.0000	0.0000	-3.4112	0.0000
14 – 67231801	Ebf2	0.0000	-0.8649	0.0000	0.0000	-0.4694
15 – 76090301	K230010J24Rik	0.0000	0.0000	0.0000	0.0000	1.1289
15 – 101054201	Scn8a	0.0000	0.0000	-3.2557	0.0000	0.0000
17 – 27555401	Hmga1	0.0000	-0.6939	0.0000	0.0000	0.0000
19 – 47015001	Ina	0.0000	0.0000	0.0000	0.0000	-0.0556
19 – 59345801	Rps12-ps3	-1.2531	0.0000	0.0000	0.0000	0.0000

Table 5.10: Shown are the coefficients of the genomic regions selected by the zero-sum multinomial and logistic regression. The first four columns show the coefficients of the multinomial regression for metastasis, primary tumor, breast and lung tissue. The last column shows the coefficients of logistic regression for predicting the probability of primary tumor.

Directly seeing the filtering effect of this multinomial regression approach is difficult and would require a detailed analysis of the genomic regions, which could involve further experiments.
Besides improving the predictivity, these models can also be used to determine highly contaminated samples. Therefore, all four tissue types are predicted. The samples with the highest contamination should be classified as background tissue. For that reason, the probabilities of the four samples with the highest probability of being normal lung or breast tissue and the four samples with the highest probability of being metastasis or primary tumor are shown in table 5.11 and 5.12. The probabilities for all sample are shown in the appendix B.4.

sample	metastasis	primary tumor	breast	lung
`(a) X3699_PT`	0.005	0.061	0.814	0.120
`(b) X3732_PT`	0.025	0.086	0.876	0.013
`(c) X3703_PT`	0.040	0.263	0.686	0.011
`(d) X3500_PT`	0.070	0.428	0.123	0.379

Table 5.11: Shown are the cross-validation probabilities of the samples with the highest probabilities of being breast or lung tissue.

sample	metastasis	primary tumor	breast	lung
`(e) X5423_MET_LUNG_2`	0.994	0.004	0.001	0.000
`(f) X3794_MET_LUNG_1`	0.954	0.045	0.001	0.000
`(g) X3627_MET_LUNG_1`	0.905	0.093	0.001	0.001
`(h) X5423_MET_LUNG_1`	0.974	0.024	0.001	0.000

Table 5.12: Shown are the cross-validation probabilities of the samples with the highest probabilities of being metastasis or primary tumor.

Whether these probabilities can be used for assessing the background contamination, can be verified by using *comparative genomic hybridization* (CGH) data, which has been additionally measured of the same biopsies. CGH data is used to detect copy-number variations (CNVs), which for instance can be duplications and depletion of genetic regions. CNVs are common in cancer and thus should occur in the samples with the lowest contamination of surrounding tissue and should be barely detectable in samples with a high contamination. The CGH data for the samples with the highest probability of being background tissue are shown in figure 5.10, while figure 5.11 shows the CGH data of the samples with the highest probability of being metastatic or tumor tissue.

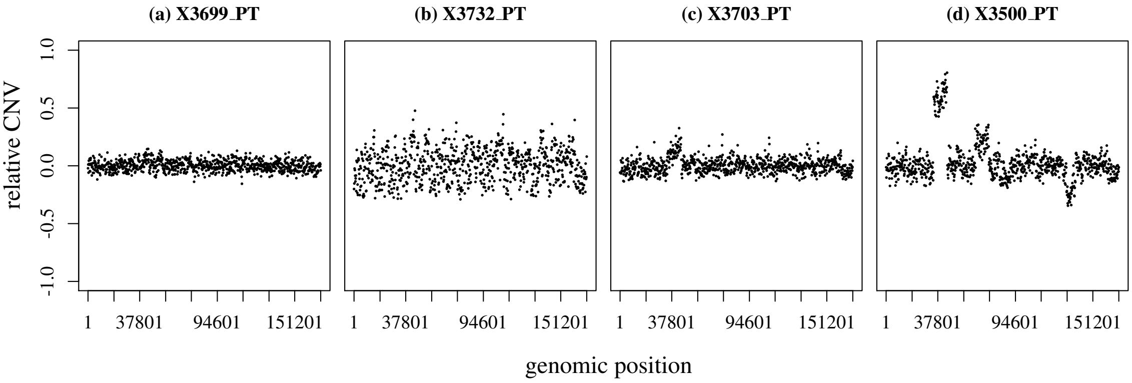


Figure 5.10: This figure shows the relative copy-number alterations as a function of the genomic position for the samples with the highest probabilities of being background tissue.

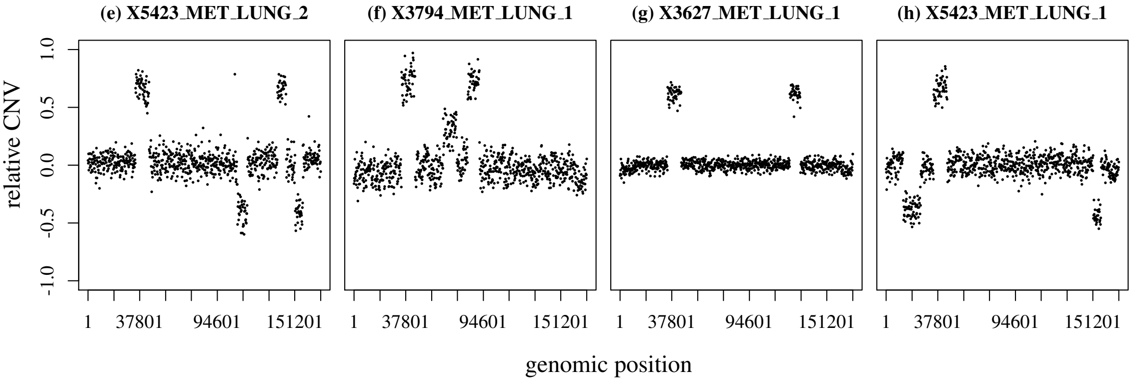


Figure 5.11: This figure shows the relative copy-number alterations as a function of the genomic position for the samples with the highest probabilities of being metastatic or tumor tissue.

The samples with the highest probability of being background tissue (a), (b), (c) in figure 5.10 hardly show any CNVs. Sample (d) shows CNVs, but the model would already classify it as primary tumor. The opposite can be seen for the samples, which have been clearly predicted to be tumor or metastatic tissue. These samples exhibit broad amplified and depleted genomic regions, which are not present in the samples predicted to be background tissue.
In conclusion it can be stated, that this approach shows that zero-sum multinomial regression in combination with the modified cross-validation is capable to improve not only the prediction accuracy, but also is capable to serve as a quality assessment tool. Another advantage of this approach is, that it only requires additional profiles of the surrounding tissue, which in *in vivo* experiments are available without requiring additional animals.

5.5 Application of Zero-Sum Cox Proportional Hazard Regression on Gene Expression Data

This section shows the application of zero-sum Cox regression on gene expression data and corresponding survival data of patients with diffuse large-B-cell lymphomas (DLBCL). This cancer type consists of two biological and clinical different subtypes: the germinal-center B-cell-like (GCB) DLBCL and the activated B-cell-like (ABC) DLBCL. These two types can be distinguished by their gene expression profiles and have an significant influence on the survival rate of the patients [2, 51, 67]. The 5-year survival rate of patients diagnosed with GCB DLBCL is 60%, while the survival rate of patients with ABC DLBCL is 30 % [51, 84].

In the following, the data set provided by Lenz et al. [51] is used. Preprocessing using VSN-normalization has been performed by C. W. Kohler of the Institute of Functional Genomics (University of Regensburg). The data consists of 414 patients with newly diagnosed DLBCL, which have been treated with the chemotherapeutic drugs CHOP (181 patients) and R-CHOP (233 patients). However, only the data of the patients which have been treated with R-CHOP is used, since this drug has been proven to be more effective and is therefore used as a standard treatment.

This data set has been analyzed using Cox regression and zero-sum Cox regression. The resulting cross-validation partial log-likelihood is shown in figure 5.12 and the obtained coefficients are shown in table 5.13.

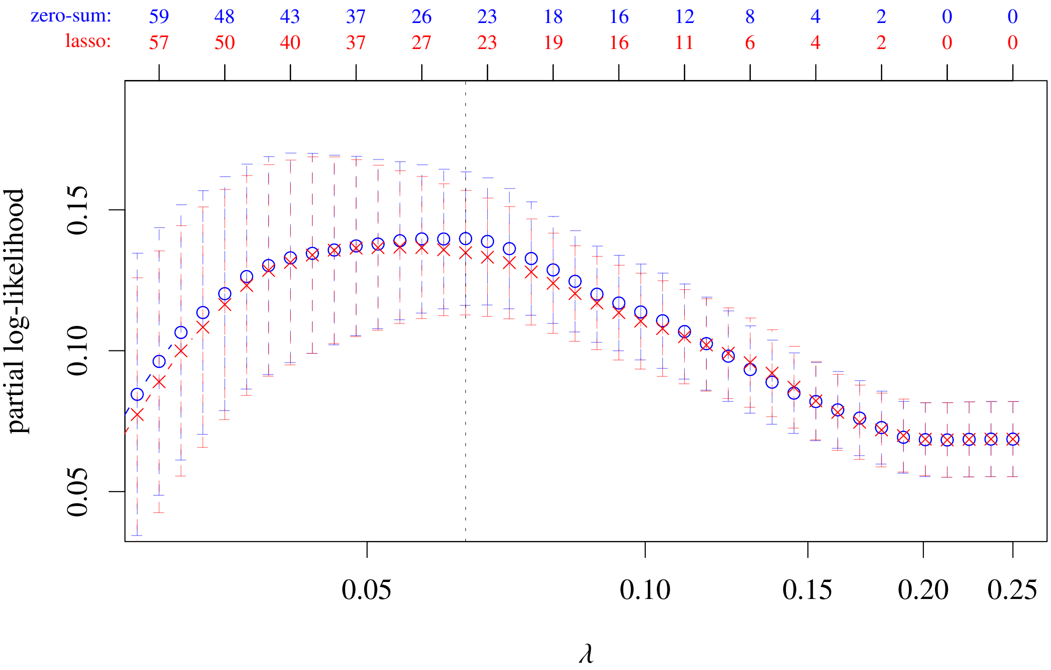


Figure 5.12: This figure shows the cross-validation partial log-likelihood as a function of the regularization parameter λ. The log-likelihood of normal Cox regression is shown in red, while the log-likelihood of zero-sum Cox regression is shown in blue. At the top, the number of non-zero coefficients is shown.

Zero-sum regression is capable to achieve a slightly better result (partial log-likelihood 0.3270 vs. 0.3241), but this difference is far within the error range.

The reason for the almost identical accuracy is that the sum of the coefficients obtained by normal Cox regression is also almost zero ($\sum_j \beta_j \approx 0.18$).

gene symbol	normal Cox regression	zero-sum Cox regression
NLRP11	-0.1957	-0.1647
FCRL3	-0.0569	-0.0093
MS4A4A	0.3691	0.3309
HBB	0.0068	0.0000
TIMP1	-0.1336	-0.1186
RPS4Y1	0.0794	0.0646
RCAN2	-0.0162	0.0000
CXCL9	-0.0439	-0.0153
FABP4	0.1122	0.1310
LMO2	-0.0799	-0.0665
MMP12	-0.0206	-0.0077
VSIG4	0.0664	0.0490
PHF16	0.0546	0.0739
BCL2A1	-0.2074	-0.1808
SPINK2	-0.0024	-0.0095
XK	0.1566	0.1306
CXCR4	0.0493	0.0008
ADRA2A	-0.0068	0.0000
CCL18	0.0924	0.0767
SULF1	-0.0350	-0.0132
CD3D	-0.0735	-0.0535
C3	-0.0646	-0.0815
NGFRAP1	-0.0669	-0.0300
STXBP6	-0.0366	0.0000
C15orf48	-0.0888	-0.0916
BEX2	-0.0762	-0.0705
GTSF1	0.0804	0.0720
AMICA1	-0.0099	0.0000
EOMES	-0.0346	-0.0166

Table 5.13: Shown are the non-zero coefficients of the genes selected by Cox regression and zero-sum Cox regression.

In summary, zero-sum Cox is not able to achieve better results than Cox regression for this data-set. The reason for that is, that Cox regression is able to identify an (almost) zero-sum model. Thus, the advantage of zero-sum Cox regression is that this solution is enforced.

6 Summary and Outlook

This thesis showed that a scale invariance in generalized linear models for log-transformed omics data can be achieved by using the zero-sum constraint. Therefore, the approach proposed by Friedman et al. [26], Simon et al. [70] to reduce the cost functions of linear, logistic, multinomial and Cox proportional hazard regression onto the same quadratic form using the quadratic approximation, has been followed. This approach allows to solve these regression problems with the same coordinate descent algorithm and therefore has been extended in this thesis to also incorporate the zero-sum constraint.

At first, it has been shown how the zero-sum constraint can be incorporated into the unified cost function and how an extended coordinate descent algorithm can be developed. Moreover, it has been shown how convergence issues of a naive coordinate descent approach can be resolved by using search space rotations, a concluding local search procedure and *warm starts*. Afterwards, the convergence behavior of the developed algorithm has been evaluated by comparing the results with general purpose optimization algorithms. Subsequently, the scale invariance caused by the zero-sum constraint has not only been demonstrated in simulations, but also in applications on omics data sets. It has been shown, that zero-sum regression yields the same linear models independent of the applied normalization. Furthermore, it has been demonstrated that the additional constraint does not impair the predicivity of models trained on omics data, but in most cases increases it.

A publicly available implementation of the algorithms is provided as the R-package *zeroSum* and as a command-line application. Both are based on the same C++ core, which has been optimized to utilize modern CPUs effectively by using AVX, AVX2 and AVX512 vector instructions and OpenMP/MPI parallelism. The software *zeroSum* is already used by the scientific community and, for instance, has been applied to better distinguish between subtypes of diffuse large B cell lymphoma using gene expression data [10, 62, 73]. Another important application of *zeroSum* is the transfer of molecular biomarkers from one measurement platform to another [4]. Moreover, one of the authors of the approach presented in [26, 70], which zero-sum regression is based on, recently also suggested the use of the zero-sum constraint [7].

All in all, it can be stated that zero-sum regression resolves scaling and normalization issues and thereby contributes to a better understanding of omics data.

However, there is still room left for software improvements. In particular, the performance of *zeroSum* may become a limiting factor due to the rapid increase in size of omics data. One of the most promising approaches to prepare *zeroSum* for such data is to utilize GPUs, by parallelizing the coordinate descent algorithm itself. Therefore, the first step of the algorithm, which updates all combinations of features and identifies the active set, has to be parallelized. This could be achieved by separating the active set search and the updating of the coefficients by using the update scheme to only identify important features without updating the coefficients. Thereby, the sequential nature of the coordinate descent approach can be circumvented and the active set search performed in parallel.

Zero-sum regression itself can also be further improved by allowing that only subsets of the coefficients add up to zero. Thereby, different omics data sets can be combined while maintaining the scale invariance of each data set. Furthermore, the local search algorithm for solving the generalized lasso is only a proof of concept and an efficient optimization strategy remains open for further improvements.

Another future development could also be the incorporation of the zero-sum concept into other machine learning techniques like artificial neural networks, where an equivalent scale invariance could be achieved by demanding that the weights of a neuron of the input layer add up to zero.

The future of zero-sum regression still remains challenging, since not only adaptations to new hardware and general improvements are possible, but also the transfer of the scale invariance to other methods offers plenty of room for improvements.

A Calculations

A.1 Coordinate Descent Update Scheme for Elastic Net Regularized Regression

The local update scheme needed for a coordinate descent algorithm can be obtained by first calculating the partial derivative of (2.10) with respect to the coefficient β_k:

$$\frac{\partial \mathcal{H}_\lambda(\beta_0, \boldsymbol{\beta})}{\partial \beta_k} = -\sum_{i=1}^{N} w_i x_{ik}\Big(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_i\Big) + \lambda(1-\alpha)v_k\beta_k + \lambda v_k \begin{cases} (-\alpha) & \text{if } \beta_k < 0 \\ \alpha & \text{if } \beta_k > 0 \\ \text{derivative not defined} & \text{if } \beta_k = 0 \end{cases} . \tag{A.1.1}$$

By setting the partial derivative to zero

$$\frac{\partial \mathcal{H}_\lambda(\beta_0, \boldsymbol{\beta})}{\partial \beta_k} \overset{!}{=} 0 \tag{A.1.2}$$

and solving for β_k using $v_j \geq 0\ \forall j$ and $w_i > 0\ \forall i$ one obtains the follow update scheme for determining the optimal value $\hat{\beta}_k$ for k coefficient:

$$\hat{\beta}_k \leftarrow \frac{1}{\sum_{i=1}^{N} w_i x_{ik}^2 + \lambda(1-\alpha)v_k} \cdot \begin{cases} \Big(\sum_{i=1}^{N} w_i x_{ik}\Big(y_i - \beta_0 - \sum\limits_{\substack{j=1\\ j\neq k}}^{p} x_{ij}\beta_j\Big) + \lambda\alpha v_k\Big) & \text{if } f_k < 0 \wedge g_k < |f_k| \\ \Big(\sum_{i=1}^{N} w_i x_{ik}\Big(y_i - \beta_0 - \sum\limits_{\substack{j=1\\ j\neq k}}^{p} x_{ij}\beta_j\Big) - \lambda\alpha v_k\Big) & \text{if } f_k > 0 \wedge g_k < |f_k| \\ 0 & \text{if } g_k \geq |f_k| \end{cases} \tag{A.1.3}$$

$$\text{with} \quad f_k := \sum_{i=1}^{N} w_i x_{ik}\Big(y_i - \beta_0 - \sum_{\substack{j=1\\ j\neq k}}^{p} x_{ij}\beta_j\Big) \quad \text{and} \quad g_k := \lambda\alpha v_k\,. \tag{A.1.4}$$

An update scheme for the intercept is obtained analogously:

$$\frac{\partial \mathcal{H}_\lambda(\beta_0, \boldsymbol{\beta})}{\partial \beta_0} = -\sum_{i=1}^{N} w_i\Big(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_i\Big) \overset{!}{=} 0 \tag{A.1.5}$$

$$\Rightarrow \beta_0 = \frac{\sum_{i=1}^{N} w_i\Big(y_i - \sum_{j=1}^{p} x_{ij}\beta_i\Big)}{\sum_{i=1}^{N} w_i}\,. \tag{A.1.6}$$

A.2 Intermediate Steps of Transforming the Logistic Regression Log-Likelihood

Derivation of (2.26) from (2.25):

$$
\begin{aligned}
\mathcal{L}(\beta_0, \boldsymbol{\beta}) &= \sum_{i=1}^{N} \Big\{ y_i \log p(\boldsymbol{x}_i) + (1 - y_i) \log(1 - p(\boldsymbol{x}_i)) \Big\} && \text{(A.2.1)} \\
&= \sum_{i=1}^{N} \Big\{ y_i \log \left(\frac{e^{\beta_0 + \boldsymbol{x}_i^T \boldsymbol{\beta}}}{1 + e^{\beta_0 + \boldsymbol{x}_i^T \boldsymbol{\beta}}} \right) + (1 - y_i) \log(1 - \frac{e^{\beta_0 + \boldsymbol{x}_i^T \boldsymbol{\beta}}}{1 + e^{\beta_0 + \boldsymbol{x}_i^T \boldsymbol{\beta}}}) \Big\} && \text{(A.2.2)} \\
&= \sum_{i=1}^{N} \Big\{ y_i \left(\beta_0 + \boldsymbol{x}_i^T \boldsymbol{\beta} - \log \left(1 + e^{\beta_0 + \boldsymbol{x}_i^T \boldsymbol{\beta}} \right) \right) + (1 - y_i) \log \left(\frac{1}{1 + e^{\beta_0 + \boldsymbol{x}_i^T \boldsymbol{\beta}}} \right) \Big\} && \text{(A.2.3)} \\
&= \sum_{i=1}^{N} \Big\{ y_i \left(\beta_0 + \boldsymbol{x}_i^T \boldsymbol{\beta} \right) - y_i \log \left(1 + e^{\beta_0 + \boldsymbol{x}_i^T \boldsymbol{\beta}} \right) - (1 - y_i) \log \left(1 + e^{\beta_0 + \boldsymbol{x}_i^T \boldsymbol{\beta}} \right) \Big\} && \text{(A.2.4)} \\
&= \sum_{i=1}^{N} \Big\{ y_i (\beta_0 + \boldsymbol{x}_i^T \boldsymbol{\beta}) - \log(1 + e^{\beta_0 + \boldsymbol{x}_i^T \boldsymbol{\beta}}) \Big\} && \text{(A.2.5)}
\end{aligned}
$$

.

A.3 Quadratic Approximation

The second order multidimensional Taylor expansion $\mathrm{Tf}_{\boldsymbol{a}}(\boldsymbol{x})$ of a function $f(\boldsymbol{x})$ centered at $\boldsymbol{a}$ can be transformed as follows to be of equivalent form as the weighted residual sum of squares (2.8) [26, 70]. This is based on the assumption that the Hessian matrix $\boldsymbol{H}_f$ of the function f is symmetric, which is the case if the Schwarz theorem is satisfied for f:

$$
\begin{aligned}
\mathrm{Tf}_{\boldsymbol{a}}(\boldsymbol{x}) &= f(\boldsymbol{a}) + (\boldsymbol{x} - \boldsymbol{a})^T \nabla f(\boldsymbol{a}) + \frac{1}{2} (\boldsymbol{x} - \boldsymbol{a})^T \boldsymbol{H}_f(\boldsymbol{a}) (\boldsymbol{x} - \boldsymbol{a}) && \text{(A.3.6)} \\
&= \frac{1}{2} \Big[(\boldsymbol{a} - \boldsymbol{x})^T \boldsymbol{H}_f(\boldsymbol{a}) (\boldsymbol{a} - \boldsymbol{x}) - 2 (\boldsymbol{a} - \boldsymbol{x})^T \boldsymbol{H}_f(\boldsymbol{a}) \boldsymbol{H}_f^{-1}(\boldsymbol{a}) \nabla f(\boldsymbol{a}) \Big] + f(\boldsymbol{a}) && \text{(A.3.7)} \\
&= \frac{1}{2} \Big[(\boldsymbol{a} - \boldsymbol{x})^T \boldsymbol{H}_f(\boldsymbol{a}) (\boldsymbol{a} - \boldsymbol{x}) - (\boldsymbol{a} - \boldsymbol{x})^T \boldsymbol{H}_f(\boldsymbol{a}) \boldsymbol{H}_f^{-1}(\boldsymbol{a}) \nabla f(\boldsymbol{a}) \\
&\qquad - \left(\boldsymbol{H}_f^{-1}(\boldsymbol{a}) \nabla f(\boldsymbol{a}) \right)^T \left((\boldsymbol{a} - \boldsymbol{x})^T \boldsymbol{H}_f(\boldsymbol{a}) \right)^T \Big] + f(\boldsymbol{a}) && \text{(A.3.8)} \\
&= \frac{1}{2} \Big[(\boldsymbol{a} - \boldsymbol{x})^T \boldsymbol{H}_f(\boldsymbol{a}) (\boldsymbol{a} - \boldsymbol{x}) - (\boldsymbol{a} - \boldsymbol{x})^T \boldsymbol{H}_f(\boldsymbol{a}) \boldsymbol{H}_f^{-1}(\boldsymbol{a}) \nabla f(\boldsymbol{a}) - \left(\boldsymbol{H}_f^{-1}(\boldsymbol{a}) \nabla f(\boldsymbol{a}) \right)^T \boldsymbol{H}_f(\boldsymbol{a}) (\boldsymbol{a} - \boldsymbol{x}) \\
&\qquad + \left(\boldsymbol{H}_f^{-1}(\boldsymbol{a}) \nabla f(\boldsymbol{a}) \right)^T \boldsymbol{H}_f(\boldsymbol{a}) \left(\boldsymbol{H}_f^{-1}(\boldsymbol{a}) \nabla f(\boldsymbol{a}) \right) \Big] \underbrace{- \frac{1}{2} \left(\boldsymbol{H}_f^{-1}(\boldsymbol{a}) \nabla f(\boldsymbol{a}) \right)^T \boldsymbol{H}_f(\boldsymbol{a}) \left(\boldsymbol{H}_f^{-1}(\boldsymbol{a}) \nabla f(\boldsymbol{a}) \right) + f(\boldsymbol{a})}_{C(\boldsymbol{a})} \\
& && \text{(A.3.9)} \\
&= \frac{1}{2} \Big(\underbrace{\boldsymbol{a} - \boldsymbol{H}_f^{-1}(\boldsymbol{a}) \nabla f(\boldsymbol{a})}_{\tilde{z}(\boldsymbol{a})} - \boldsymbol{x} \Big)^T \boldsymbol{H}_f(\boldsymbol{a}) \Big(\boldsymbol{a} - \boldsymbol{H}_f^{-1}(\boldsymbol{a}) \nabla f(\boldsymbol{a}) - \boldsymbol{x} \Big) + C(\boldsymbol{a}) && \text{(A.3.10)} \\
&= \frac{1}{2} (\tilde{z}(\boldsymbol{a}) - \boldsymbol{x})^T \boldsymbol{H}_f(\boldsymbol{a}) (\tilde{z}(\boldsymbol{a}) - \boldsymbol{x}) + C(\boldsymbol{a}) \,. && \text{(A.3.11)}
\end{aligned}
$$

$C(\boldsymbol{a})$ only depends on $\boldsymbol{a}$ and not on $\boldsymbol{x}$ and therefore vanishes in the calculation of the partial derivatives. Since the calculation of the Hessian matrix $\boldsymbol{H}_f$ is computationally demanding, it is replaced by an approximation, which only consists of the diagonal elements of $\boldsymbol{H}_f$ with the reasoning that non-diagonal elements are small in comparison to the diagonal elements [31, 70]. In the case of the logistic and multinomial regression the non-diagonal elements are even zero.

In order to obtain the quadratic form, the negative values of the diagonal elements are composed to a vector $\tilde{\boldsymbol{w}}$:

$$\mathrm{Tf}_{\boldsymbol{a}}(\boldsymbol{x}) = -\frac{1}{2}\sum_i \tilde{w}_i(\boldsymbol{a})\Big(\tilde{z}_i(\boldsymbol{a}) - x\Big)^2 + C(\boldsymbol{a}) \tag{A.3.12}$$

with

$$\tilde{w}_i(\boldsymbol{a}) = -\frac{\partial^2 f}{\partial x_i^2}(\boldsymbol{a}) \tag{A.3.13}$$

$$\tilde{z}_i(\boldsymbol{a}) = a_i - \left(\frac{\partial^2 f}{\partial x_i^2}(\boldsymbol{a})\right)^{-1} \frac{\partial f}{\partial x_i}(\boldsymbol{a})\,. \tag{A.3.14}$$

A.4 Intermediate Steps of Transforming the Multinomial Regression Log-Likelihood

The intermediate steps of transforming the multinomial regression log-likelihood from (2.42) to (2.43) are:

$$\mathcal{L}(\{\beta_{0h}, \boldsymbol{\beta_h}\}_1^K) = \sum_{i=1}^{N} w_i \sum_{l=1}^{K} y_{il} \log p_l(\boldsymbol{x}_i) \tag{A.4.1}$$

$$= \sum_{i=1}^{N} w_i \sum_{l=1}^{K} y_{il} \log\Big(\frac{e^{\beta_{0l}+\boldsymbol{x}_i^T\boldsymbol{\beta}_l}}{\sum_{k=1}^{K} e^{\beta_{0k}+\boldsymbol{x}_i^T\boldsymbol{\beta}_k}}\Big) \tag{A.4.2}$$

$$= \sum_{i=1}^{N} w_i \Big[\sum_{l=1}^{K} y_{il}(\beta_{0l} + \boldsymbol{x}_i^T\boldsymbol{\beta}_l) - \log\Big(\sum_{k=1}^{K} e^{\beta_{0k}+\boldsymbol{x}_i^T\boldsymbol{\beta}_k}\Big) \underbrace{\sum_{l=1}^{K} y_{il}}_{=1}\Big] \tag{A.4.3}$$

$$= \sum_{i=1}^{N} w_i \Big[\sum_{l=1}^{K} y_{il}(\beta_{0l} + \boldsymbol{x}_i^T\boldsymbol{\beta}_l) - \log\Big(\sum_{k=1}^{K} e^{\beta_{0k}+\boldsymbol{x}_i^T\boldsymbol{\beta}_k}\Big)\Big]\,. \tag{A.4.4}$$

A.5 Quadratic Approximation of the Multinomial Regression Log-Likelihood

In order to apply the Taylor approximation (2.29) to the multinomial regression cost function (2.45) the first and second derivatives with respect to $(\beta_{0l} + \boldsymbol{x}_i^T \boldsymbol{\beta}_l)$ have to be calculated:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial(\beta_{0l} + \boldsymbol{x}_i^T \boldsymbol{\beta}_l)}(\{\beta_{0h}, \boldsymbol{\beta}_h\}_1^K) &= w_i y_{il} - \frac{w_i e^{\beta_{0l} + \boldsymbol{x}_i^T \boldsymbol{\beta}_l}}{\sum_{k=1}^K e^{\beta_{0k} + \boldsymbol{x}_i^T \boldsymbol{\beta}_k}} && \text{(A.5.1)} \\ &= w_i (y_{il} - p_l(\boldsymbol{x}_i)) && \text{(A.5.2)} \\ \frac{\partial^2 \mathcal{L}}{\partial(\beta_{0l} + \boldsymbol{x}_i^T \boldsymbol{\beta}_l)^2}(\{\beta_{0h}, \boldsymbol{\beta}_h\}_1^K) &= -w_i p_l(\boldsymbol{x}_i) \frac{\sum_{k=1}^K e^{\beta_{0k} + \boldsymbol{x}_i^T \boldsymbol{\beta}_k} - e^{\beta_{0l} + \boldsymbol{x}_i^T \boldsymbol{\beta}_l}}{\sum_{k=1}^K e^{\beta_{0k} + \boldsymbol{x}_i^T \boldsymbol{\beta}_k}} && \text{(A.5.3)} \\ &= -w_i p_l(\boldsymbol{x}_i)(1 - p_l(\boldsymbol{x}_i))\,. && \text{(A.5.4)} \end{aligned}$$

The parameters of the approximation $\tilde{w}_{il}$ and $\tilde{z}_{il}$ centered at $(\tilde{\beta}_0, \tilde{\boldsymbol{\beta}})$ are:

$$\begin{aligned} \tilde{w}_{il} &= w_i \tilde{p}_l(\boldsymbol{x}_i)(1 - \tilde{p}_l(\boldsymbol{x}_i)) && \text{(A.5.5)} \\ \tilde{z}_{il} &= \beta_{0l} + \boldsymbol{\beta}_l + \frac{y_i - \tilde{p}_l(\boldsymbol{x}_i)}{\tilde{p}_l(\boldsymbol{x}_i)(1 - \tilde{p}_l(\boldsymbol{x}_i))}\,. && \text{(A.5.6)} \end{aligned}$$

$\tilde{p}_l(\boldsymbol{x}_i)$ denotes the probability for sample i being class l given by (2.39) evaluated with $(\tilde{\beta}_{0l}, \tilde{\boldsymbol{\beta}}_l)$.
The resulting log-likelihood is:

$$\mathcal{L}(\{\beta_{0h}, \boldsymbol{\beta}_h\}_1^K) = -\frac{1}{2} \sum_{i=1}^N \tilde{w}_{il} \left(\tilde{z}_{il} - \beta_{0l} - \boldsymbol{x}_i^T \boldsymbol{\beta}_l\right)^2 + C_l\,. \tag{A.5.7}$$

As above C_l is irrelevant for the further calculation, since it disappears in the partial derivative.

A.6 Parameter Ambiguity Problem

The parameter ambiguity problem defined by the cost function (2.61) can be solved for $v_j \neq 0$ by calculating the derivative with respect to δ_k [26]:

$$\frac{\partial R(\boldsymbol{\delta})}{\partial \delta_k} = \sum_{l=1}^K v_k \left(-(1-\alpha)(\beta_{kl} - \delta_k) + \alpha \begin{cases} -1 & \text{if } \beta_{kl} - \delta_k > 0 \\ 1 & \text{if } \beta_{kl} - \delta_k < 0 \\ \text{derivative not defined} & \end{cases} \right) \stackrel{!}{=} 0 \tag{A.6.1}$$

$$\Rightarrow K(1-\alpha)\delta_k + \sum_{l=1}^K \left(-(1-\alpha)\beta_{kl} + \alpha \begin{cases} -1 & \text{if } \beta_{kl} - \delta_k > 0 \\ 1 & \text{if } \beta_{kl} - \delta_k < 0 \\ \text{derivative not defined} & \end{cases} \right) = 0\,. \tag{A.6.2}$$

(A.6.3)

Therefore the optimal value for δ_k is given by [26]:

$$\delta_k \leftarrow \frac{1}{K} \sum_{l=1}^K \beta_{kl} - \frac{1}{K} \frac{\alpha}{1-\alpha} \sum_{l=1}^K \begin{cases} -1 & \text{if } \beta_{kl} - \delta_k > 0 \\ 1 & \text{if } \beta_{kl} - \delta_k < 0 \\ \text{derivative not defined} & \end{cases}\,. \tag{A.6.4}$$

A.7 Quadratic Approximation of the Cox Proportional Hazard Regression Log-Likelihood

The partial log-likelihood (2.66) can be approximated with the second order Taylor expansion given by (2.29) centered at $\tilde{\beta}$ by calculating the derivatives with respect to $\boldsymbol{x}_i^T\boldsymbol{\beta}$:

$$\frac{\partial\mathcal{L}}{\partial(\boldsymbol{x}_i^T\boldsymbol{\beta})}(\boldsymbol{\beta}) = w_i\delta_i - \sum_{k\in C_i}\left[\frac{d_k w_i e^{\boldsymbol{x}_i^T\boldsymbol{\beta}}}{\sum_{j\in R_k} w_j e^{\boldsymbol{x}_j^T\boldsymbol{\beta}}}\right], \tag{A.7.1}$$

$$\frac{\partial^2\mathcal{L}}{\partial(\boldsymbol{x}_i^T\boldsymbol{\beta})^2}(\boldsymbol{\beta}) = -\sum_{k\in C_i} d_k \frac{w_i e^{\boldsymbol{x}_i^T\boldsymbol{\beta}}\sum_{j\in R_k} w_j e^{\boldsymbol{x}_j^T\boldsymbol{\beta}} - w_i^2 e^{2\boldsymbol{x}_i^T\boldsymbol{\beta}}}{\left(\sum_{j\in R_k} w_j e^{\boldsymbol{x}_j^T\boldsymbol{\beta}}\right)^2}. \tag{A.7.2}$$

where C_i denotes the set of sets D_j where the observation time y_j is less than or equal to the observation time of the set D_i ($y_j \leq y_i$ for all elements of C_i). The parameters $\tilde{w}_i$ and $\tilde{z}_i$ centered at $\tilde{\boldsymbol{\beta}}$ are:

$$\tilde{w}_i = \sum_{k\in C_i} d_k \frac{w_i e^{\boldsymbol{x}_i^T\boldsymbol{\beta}}\sum_{j\in R_k} w_j e^{\boldsymbol{x}_j^T\boldsymbol{\beta}} - w_i^2 e^{2\boldsymbol{x}_i^T\boldsymbol{\beta}}}{\left(\sum_{j\in R_k} w_j e^{\boldsymbol{x}_j^T\boldsymbol{\beta}}\right)^2}, \tag{A.7.3}$$

$$\tilde{z}_i = \boldsymbol{x}_i^T\tilde{\boldsymbol{\beta}}_l + \frac{1}{\tilde{w}_i}\left[w_i\delta_i - \sum_{k\in C_i}\frac{d_k w_i e^{\boldsymbol{x}_i^T\boldsymbol{\beta}}}{\sum_{j\in R_k} w_j e^{\boldsymbol{x}_j^T\boldsymbol{\beta}}}\right]. \tag{A.7.4}$$

The resulting log-likelihood is:

$$\mathcal{L}(\boldsymbol{\beta}) = -\frac{1}{2}\sum_{i=1}^{N} w_i\left(z_i - \boldsymbol{x}_i^T\boldsymbol{\beta}\right)^2 + C(\tilde{\boldsymbol{\beta}}). \tag{A.7.5}$$

Note that a notation conflict occurs due to different conventions: C_i denotes the sets of samples, while $C(\tilde{\boldsymbol{\beta}})$ derives from the quadratic approximation and is as above irrelevant for the further calculations.
In this case the off diagonal elements are not zero and could be an issue for the approximation. However, as shown in [31] and mentioned in [70, 77] the off-diagonal elements can be neglected.

A.8 Construction of the Normal Zero-Sum Coordinate Descent Update Scheme

The partial derivative of (3.17) with respect to β_k can be used to determine the local optimal value for β_k:

$$\begin{aligned}\frac{\partial\mathcal{H}_\lambda(\beta_0,\boldsymbol{\beta})}{\partial\beta_k} &= \sum_{i=1}^{N} w_i\left(-x_{ik} + \frac{x_{is}u_k}{u_s}\right)\left(y_i - \beta_0 - \sum_{\substack{j=1\\ j\neq s,k}}^{p} x_{ij}\beta_j - \frac{x_{is}}{u_s}\Big(c - \sum_{\substack{j=1\\ j\neq s,k}}^{p} u_j\beta_j\Big)\right) - \frac{\lambda v_s(1-\alpha)u_k}{u_s^2}\Big(c - \sum_{\substack{j=1\\ j\neq s,k}}^{p} u_j\beta_j\Big)\\ &+\beta_k\left(\sum_{i=1}^{N} w_i\Big(-x_{ik} + \frac{x_{is}u_k}{u_s}\Big)^2 + \lambda v_k(1-\alpha) + \frac{\lambda v_s(1-\alpha)u_k^2}{u_s^2}\right)\\ &+\lambda\alpha\begin{cases}(\ \ v_k - \frac{v_s u_k}{u_s}) & \text{if} \quad \beta_k > 0 \ \wedge \ \beta_s > 0\\ (\ \ v_k + \frac{v_s u_k}{u_s}) & \text{if} \quad \beta_k > 0 \ \wedge \ \beta_s < 0\\ (-v_k - \frac{v_s u_k}{u_s}) & \text{if} \quad \beta_k < 0 \ \wedge \ \beta_s > 0\\ (-v_k + \frac{v_s u_k}{u_s}) & \text{if} \quad \beta_k < 0 \ \wedge \ \beta_s < 0\\ \text{derivative not defined} & \end{cases}.\end{aligned} \tag{A.8.1}$$

The last condition of the case differentiation uses that $c - \sum_{j=1, j\neq s}^{p} u_j\beta_j = u_s\beta_s$ and $u_j > 0\ \forall j$.
By setting the partial derivative to zero and by solving for β_k the following update scheme is obtained:

$$\hat{\beta}_k \leftarrow \frac{1}{a_{ks}} \cdot \begin{cases} \Big(b_{ks} - \lambda\alpha(v_k - \frac{v_s u_k}{u_s})\Big) & \text{if} \quad \hat{\beta}_k > 0 \ \wedge \ \hat{\beta}_s > 0 \\ \Big(b_{ks} - \lambda\alpha(v_k + \frac{v_s u_k}{u_s})\Big) & \text{if} \quad \hat{\beta}_k > 0 \ \wedge \ \hat{\beta}_s < 0 \\ \Big(b_{ks} + \lambda\alpha(v_k + \frac{v_s u_k}{u_s})\Big) & \text{if} \quad \hat{\beta}_k < 0 \ \wedge \ \hat{\beta}_s > 0 \\ \Big(b_{ks} + \lambda\alpha(v_k - \frac{v_s u_k}{u_s})\Big) & \text{if} \quad \hat{\beta}_k < 0 \ \wedge \ \hat{\beta}_s < 0 \\ \text{derivative not defined, skip update} \end{cases}, \tag{A.8.2}$$

with

$$a_{ks} = \sum_{i=1}^{N} w_i(-x_{ik} + \frac{x_{is}u_k}{u_s})^2 + \lambda v_k(1-\alpha) + \lambda(1-\alpha)\frac{v_s u_k^2}{u_s^2}\,, \tag{A.8.3}$$

$$b_{ks} = -\sum_{i=1}^{N} w_i(-x_{ik} + \frac{x_{is}u_k}{u_s})\Bigg(y_i - \beta_0 - \sum_{\substack{j=1\\ j\neq s,k}}^{p} x_{ij}\beta_j - \frac{x_{is}}{u_s}\Big(c - \sum_{\substack{j=1\\ j\neq s,k}}^{p} u_j\beta_j\Big)\Bigg) + \frac{\lambda v_s(1-\alpha)u_k}{u_s^2}\Big(c - \sum_{\substack{j=1\\ j\neq s,k}}^{p} u_j\beta_j\Big)\,. \tag{A.8.4}$$

A.9 Construction of the Rotated Zero-Sum Coordinate Descent Update Scheme

Rotating and translating the cost function (3.17) with (3.21) yields:

$$\begin{aligned} \mathcal{H}_\lambda(\beta_0, \boldsymbol{\beta}') =& \frac{1}{2}\sum_{i=1}^{N} w_i\Bigg(y_i - \beta_0 - \sum_{\substack{j=1\\ j\neq n,m,s}}^{p} x_{ij}\beta_j - x_{in}\,(\beta'_n\cos\theta + \beta'_m\sin\theta + t_1) - x_{im}\,(-\beta'_n\sin\theta + \beta'_m\cos\theta + t_2) - \frac{x_{is}c}{u_s} \\ &+ \frac{x_{is}}{u_s}\Big(\sum_{\substack{j=1\\ j\neq n,m,s}}^{p} u_j\beta_j + u_n\beta'_n\cos\theta + u_n\beta'_m\sin\theta + u_n t_1 - u_m\beta'_n\sin\theta + u_m\beta'_m\cos\theta + u_m t_2\Big)\Bigg)^2 \\ &+ \frac{\lambda(1-\alpha)}{2}\Bigg(\sum_{\substack{j=1\\ j\neq n,m,s}}^{p} v_j\beta_j^2 + v_n\,(\beta'_n\cos\theta + \beta'_m\sin\theta + t_1)^2 + v_m\,(-\beta'_n\sin\theta + \beta'_m\cos\theta + t_2)^2\Bigg) \\ &+ \frac{\lambda v_s(1-\alpha)}{2u_s^2}\Bigg(c - \sum_{\substack{j=1\\ j\neq n,m,s}}^{p} u_j\beta_j - u_n\beta'_n\cos\theta - u_n\beta'_m\sin\theta - u_n t_1 + u_m\beta'_n\sin\theta - u_m\beta'_m\cos\theta - u_m t_2\Bigg)^2 \\ &+ \lambda\alpha\Bigg(\sum_{\substack{j=1\\ j\neq n,m,s}}^{p} v_j|\beta_j| + v_n\left|\beta'_n\cos\theta + \beta'_m\sin\theta + t_1\right| + v_m\left|-\beta'_n\sin\theta + \beta'_m\cos\theta + t_2\right|\Bigg) \\ &+ \lambda\alpha\left|\frac{v_s}{u_s}\Bigg(c - \sum_{\substack{j=1\\ j\neq n,m,s}}^{p} u_j\beta_j - u_n\beta'_n\cos\theta - u_n\beta'_m\sin\theta - u_n t_1 + u_m\beta'_n\sin\theta - u_m\beta'_m\cos\theta - u_m t_2\Bigg)\right|. \end{aligned} \tag{A.9.1}$$

In the following it is already assumed that β_m' is zero, since t_1 and t_2 will be set to β_n and β_m. The partial derivative with respect to β_n' is thus given by:

$$
\begin{aligned}
\frac{\partial \mathcal{H}_\lambda(\beta_0,\boldsymbol{\beta}')}{\partial \beta_n'} &= \sum_{i=1}^{N} w_i\Big(x_{im}\sin\theta - x_{in}\cos\theta + \frac{x_{is}}{u_s}(u_n\cos\theta - u_m\sin\theta)\Big) \\
&\cdot\Bigg(y_i - \beta_0 - \sum_{\substack{j=1\\ j\neq s,n,m}}^{p} x_{ij}\beta_j - x_{in}t_1 - x_{im}t_2 - \frac{x_{is}c}{u_s} + \frac{x_{is}}{u_s}\Big(\sum_{\substack{j=1\\ j\neq s,n,m}}^{p} u_j\beta_j + u_n t_1 + u_m t_2\Big)\Bigg) \\
&+ \beta_n' \sum_{i=1}^{N} w_i\Big(x_{im}\sin\theta - x_{in}\cos\theta + \frac{x_{is}}{u_s}(u_n\cos\theta - u_m\sin\theta)\Big)^2 \\
&+ \lambda(1-\alpha)\Big(v_n\,(\beta_n'\cos\theta + t_1)\cos\theta - v_m\,(-\beta_n'\sin\theta + t_2)\sin\theta\Big) \\
&+ \frac{\lambda v_s(1-\alpha)}{u_s^2}\Big(c - \sum_{\substack{j=1\\ j\neq n,m,s}}^{p} u_j\beta_j - u_n\beta_n'\cos\theta - u_n t_1 + u_m\beta_n'\sin\theta - u_m t_2\Big)\cdot\Big(-u_n\cos\theta + u_m\sin\theta\Big) \\
&+ \lambda\alpha\cdot\begin{cases}
v_n\cos\theta - v_m\sin\theta + (-u_n\cos\theta + u_m\sin\theta)v_s/u_s & \text{if } \hat{\beta}_n > 0 \wedge \hat{\beta}_m > 0 \wedge \hat{\beta}_s > 0 \\
v_n\cos\theta - v_m\sin\theta + (\;\;u_n\cos\theta - u_m\sin\theta)v_s/u_s & \text{if } \hat{\beta}_n > 0 \wedge \hat{\beta}_m > 0 \wedge \hat{\beta}_s < 0 \\
v_n\cos\theta + v_m\sin\theta + (-u_n\cos\theta + u_m\sin\theta)v_s/u_s & \text{if } \hat{\beta}_n > 0 \wedge \hat{\beta}_m < 0 \wedge \hat{\beta}_s > 0 \\
v_n\cos\theta + v_m\sin\theta + (\;\;u_n\cos\theta - u_m\sin\theta)v_s/u_s & \text{if } \hat{\beta}_n > 0 \wedge \hat{\beta}_m < 0 \wedge \hat{\beta}_s < 0 \\
-v_n\cos\theta - v_m\sin\theta + (-u_n\cos\theta + u_m\sin\theta)v_s/u_s & \text{if } \hat{\beta}_n < 0 \wedge \hat{\beta}_m > 0 \wedge \hat{\beta}_s > 0 \\
-v_n\cos\theta - v_m\sin\theta + (\;\;u_n\cos\theta - u_m\sin\theta)v_s/u_s & \text{if } \hat{\beta}_n < 0 \wedge \hat{\beta}_m > 0 \wedge \hat{\beta}_s < 0 \\
-v_n\cos\theta + v_m\sin\theta + (-u_n\cos\theta + u_m\sin\theta)v_s/u_s & \text{if } \hat{\beta}_n < 0 \wedge \hat{\beta}_m < 0 \wedge \hat{\beta}_s > 0 \\
-v_n\cos\theta + v_m\sin\theta + (\;\;u_n\cos\theta - u_m\sin\theta)v_s/u_s & \text{if } \hat{\beta}_n < 0 \wedge \hat{\beta}_m < 0 \wedge \hat{\beta}_s < 0 \\
\text{derivative not defined} &
\end{cases}.
\end{aligned}
\tag{A.9.2}
$$

Solving for β_n' by $\frac{\partial \mathcal{H}}{\partial \beta_{n'}} \overset{!}{=} 0$ yields the following update scheme:

$$
\hat{\beta}_n' \leftarrow \frac{1}{a_{nms}}\begin{cases}
b_{nms} - \lambda\alpha(\;\;\;v_n\cos\theta - v_m\sin\theta + (-u_n\cos\theta + u_m\sin\theta)v_s/u_s) & \text{if } \hat{\beta}_n > 0 \wedge \hat{\beta}_m > 0 \wedge \hat{\beta}_s > 0 \\
b_{nms} - \lambda\alpha(\;\;\;v_n\cos\theta - v_m\sin\theta + (\;\;u_n\cos\theta - u_m\sin\theta)v_s/u_s) & \text{if } \hat{\beta}_n > 0 \wedge \hat{\beta}_m > 0 \wedge \hat{\beta}_s < 0 \\
b_{nms} - \lambda\alpha(\;\;\;v_n\cos\theta + v_m\sin\theta + (-u_n\cos\theta + u_m\sin\theta)v_s/u_s) & \text{if } \hat{\beta}_n > 0 \wedge \hat{\beta}_m < 0 \wedge \hat{\beta}_s > 0 \\
b_{nms} - \lambda\alpha(\;\;\;v_n\cos\theta + v_m\sin\theta + (\;\;u_n\cos\theta - u_m\sin\theta)v_s/u_s) & \text{if } \hat{\beta}_n > 0 \wedge \hat{\beta}_m < 0 \wedge \hat{\beta}_s < 0 \\
b_{nms} - \lambda\alpha(-v_n\cos\theta - v_m\sin\theta + (-u_n\cos\theta + u_m\sin\theta)v_s/u_s) & \text{if } \hat{\beta}_n < 0 \wedge \hat{\beta}_m > 0 \wedge \hat{\beta}_s > 0 \\
b_{nms} - \lambda\alpha(-v_n\cos\theta - v_m\sin\theta + (\;\;u_n\cos\theta - u_m\sin\theta)v_s/u_s) & \text{if } \hat{\beta}_n < 0 \wedge \hat{\beta}_m > 0 \wedge \hat{\beta}_s < 0 \\
b_{nms} - \lambda\alpha(-v_n\cos\theta + v_m\sin\theta + (-u_n\cos\theta + u_m\sin\theta)v_s/u_s) & \text{if } \hat{\beta}_n < 0 \wedge \hat{\beta}_m < 0 \wedge \hat{\beta}_s > 0 \\
b_{nms} - \lambda\alpha(-v_n\cos\theta + v_m\sin\theta + (\;\;u_n\cos\theta - u_m\sin\theta)v_s/u_s) & \text{if } \hat{\beta}_n < 0 \wedge \hat{\beta}_m < 0 \wedge \hat{\beta}_s < 0 \\
\text{derivative not defined, skip update} &
\end{cases},
\tag{A.9.3}
$$

with

$$\begin{aligned} a_{nms} &= \sum_{i=1}^{N} w_i\Big(x_{im}\sin\theta - x_{in}\cos\theta + \frac{x_{is}}{u_s}(u_n\cos\theta - u_m\sin\theta)\Big)^2 + \lambda(1-\alpha)\cdot\Big(v_n\cos^2\theta + v_m\sin^2\theta \\ &\quad + \frac{v_s}{u_s^2}\big(-u_n\cos\theta + u_m\sin\theta\big)^2\Big), \end{aligned} \tag{A.9.4}$$

$$\begin{aligned} b_{nms} &= -\sum_{i=1}^{N} w_i\Big(x_{im}\sin\theta - x_{in}\cos\theta + \frac{x_{is}}{u_s}(u_n\cos\theta - u_m\sin\theta)\Big)\cdot\Big(y_i - \beta_0 - \sum_{\substack{j=1\\ j\neq s,n,m}}^{p} x_{ij}\beta_j - x_{in}t_1 - x_{im}t_2 \\ &\quad - \frac{x_{is}c}{u_s} + \frac{x_{is}}{u_s}\Big(\sum_{\substack{j=1\\ j\neq s,n,m}}^{p} u_j\beta_j + u_n t_1 + u_m t_2\Big)\Big) - \lambda(1-\alpha)\Big(v_n t_1\cos\theta - v_m t_2\sin\theta \\ &\quad + \frac{v_s}{u_s^2}\Big(c - \sum_{\substack{j=1\\ j\neq n,m,s}}^{p} u_j\beta_j - u_n t_1 - u_m t_2\Big)\cdot\big(-u_n\cos\theta + u_m\sin\theta\big)\Big). \end{aligned} \tag{A.9.5}$$

By transformation the updated values back in the original space and by using the zero-sum constraint the optimal values for $\hat{\beta}_n$, $\hat{\beta}_m$ and $\hat{\beta}_s$ are obtained.

B Supplementary Tables

B.1 Coefficients of the Regression of Section 5.2 on the Urinary AKI Metabolomics Data

spectral region	normal creatine	normal total area	normal PQN	normal TSP	zero-sum creatine	zero-sum total area	zero-sum PQN	zero-sum TSP
Intercept	0.4019	-21.8138	10.9045	-18.2190	-14.2749	-14.2749	-14.2749	-14.2749
9.405 ppm	0.0000	0.0000	-0.0733	0.0000	0.0000	0.0000	0.0000	0.0000
9.325 ppm	0.0000	-0.2913	0.0000	-0.2437	-0.2611	-0.2611	-0.2611	-0.2611
9.115 ppm	-0.0434	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
9.015 ppm	0.0000	-0.1503	0.0000	-0.0806	0.0000	0.0000	0.0000	0.0000
8.985 ppm	0.0000	-0.0097	-0.1523	-0.2945	-0.2144	-0.2144	-0.2144	-0.2144
8.835 ppm	0.0000	0.0000	0.0000	-0.1911	0.0000	0.0000	0.0000	0.0000
8.825 ppm	-0.2542	-0.5009	-0.4422	-0.2968	-0.4975	-0.4975	-0.4975	-0.4975
8.695 ppm	-0.2241	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
8.575 ppm	0.0686	0.1929	0.1837	0.4525	0.3458	0.3458	0.3458	0.3458
8.375 ppm	0.2866	0.5465	0.5179	0.3926	0.7017	0.7017	0.7017	0.7017
8.365 ppm	0.0000	-0.1798	-0.1694	-0.5016	-0.6069	-0.6069	-0.6069	-0.6069
8.325 ppm	-0.6660	-0.9680	-0.9804	-1.2027	-1.1897	-1.1897	-1.1897	-1.1897
8.105 ppm	0.0000	0.0059	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
7.795 ppm	-0.0943	-0.4767	-0.4083	-0.5208	-0.5794	-0.5794	-0.5794	-0.5794
7.715 ppm	0.1952	0.3897	0.4611	0.3572	0.4519	0.4519	0.4519	0.4519
7.615 ppm	0.0000	0.1197	0.0058	0.1247	0.0202	0.0202	0.0202	0.0202
7.285 ppm	0.0000	0.1228	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
7.165 ppm	0.0000	-0.4996	-0.2823	-0.5507	-0.3388	-0.3388	-0.3388	-0.3388
7.155 ppm	-0.0551	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
6.985 ppm	0.0696	0.0000	0.0545	0.0021	0.0190	0.0190	0.0190	0.0190
6.965 ppm	0.0000	0.0000	0.0349	0.0218	0.0226	0.0226	0.0226	0.0226
6.875 ppm	-0.0859	-0.1239	-0.2166	-0.2417	-0.3453	-0.3453	-0.3453	-0.3453
6.855 ppm	0.0000	-0.3313	0.0000	0.0000	-0.2416	-0.2416	-0.2416	-0.2416
6.655 ppm	0.3395	0.3930	0.3570	0.3634	0.3959	0.3959	0.3959	0.3959
6.605 ppm	0.0000	-0.0580	-0.0634	-0.0211	-0.1599	-0.1599	-0.1599	-0.1599
6.515 ppm	0.2387	0.2460	0.4160	0.1952	0.2546	0.2546	0.2546	0.2546
4.155 ppm	0.0000	-0.0595	-0.0091	0.0000	-0.0658	-0.0658	-0.0658	-0.0658
4.135 ppm	0.0762	0.0000	0.0438	0.0000	0.0000	0.0000	0.0000	0.0000
3.855 ppm	0.0000	0.0000	0.0330	0.0290	0.0411	0.0411	0.0411	0.0411
3.795 ppm	0.0000	0.0000	0.0766	0.0115	0.0000	0.0000	0.0000	0.0000
3.715 ppm	0.2778	0.1008	0.3583	0.2750	0.3170	0.3170	0.3170	0.3170
3.355 ppm	0.0000	0.0861	0.0132	0.0000	0.0229	0.0229	0.0229	0.0229
3.285 ppm	0.8498	1.0796	1.2930	1.5067	1.7809	1.7809	1.7809	1.7809
3.235 ppm	-0.3537	-0.5807	-0.0211	-0.4026	-0.3586	-0.3586	-0.3586	-0.3586
3.225 ppm	0.0000	0.0000	-0.1603	-0.0260	-0.0474	-0.0474	-0.0474	-0.0474
3.145 ppm	0.2767	0.5487	0.6439	0.5969	0.8695	0.8695	0.8695	0.8695

Table B.1: Shown is the first part of the non-zero coefficients of the spectral regions selected by the normal and zero-sum logistic regression applied on the total area, PQN, and TSP normalized urinary AKI data.

spectral region	normal creatine	normal total area	normal PQN	normal TSP	zero-sum creatine	zero-sum total area	zero-sum PQN	zero-sum TSP
2.975 ppm	0.0000	0.0000	0.0347	0.1646	0.1507	0.1507	0.1507	0.1507
2.935 ppm	0.0000	-0.0011	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
2.695 ppm	0.0000	0.0901	0.0499	0.1713	0.1162	0.1162	0.1162	0.1162
2.345 ppm	0.0000	-0.1569	-0.0454	-0.4034	-0.4530	-0.4530	-0.4530	-0.4530
2.325 ppm	0.0000	-0.1569	-0.0422	-0.2893	-0.1217	-0.1217	-0.1217	-0.1217
2.225 ppm	0.0000	-0.2482	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
2.065 ppm	0.0000	-0.3790	0.0000	-0.2992	-0.3421	-0.3421	-0.3421	-0.3421
1.155 ppm	0.0466	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
1.135 ppm	0.1601	0.0000	0.3310	0.2108	0.2456	0.2456	0.2456	0.2456
0.625 ppm	0.0000	0.0000	0.0239	0.0000	0.0439	0.0439	0.0439	0.0439
0.535 ppm	0.0000	0.0823	0.0144	0.0003	0.0235	0.0235	0.0235	0.0235

Table B.2: Shown is the second part of the non-zero coefficients of the spectral regions selected by the normal and zero-sum logistic regression applied on the total area, PQN, and TSP normalized urinary AKI data. (part2, part1 is on the previous page)

B.2 Coefficients of the Regression of Section 5.2 on the Plasma AKI Metabolomics Data

spectral region	normal total area	normal PQN	normal TSP	zero-sum total area	zero-sum PQN	zero-sum TSP
Intercept	19.7034	-2.2943	16.1284	-5.7743	-5.7743	-5.7743
9.165 ppm	0.0000	-0.1601	0.0000	-0.2262	-0.2262	-0.2262
8.935 ppm	0.0000	-0.0038	0.0000	-0.0430	-0.0430	-0.0430
8.745 ppm	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
8.235 ppm	-0.3271	-0.2854	-0.3922	-0.2636	-0.2636	-0.2636
8.185 ppm	-0.4638	-0.8395	-0.3456	-0.8639	-0.8639	-0.8639
7.835 ppm	0.5880	0.6777	0.6606	0.6423	0.6423	0.6423
7.805 ppm	-0.2099	-0.3192	-0.2318	-0.3318	-0.3318	-0.3318
7.785 ppm	-0.0851	-0.1916	-0.1039	-0.2144	-0.2144	-0.2144
7.715 ppm	0.1145	0.0739	0.1319	0.0419	0.0419	0.0419
7.575 ppm	0.3176	0.2922	0.3343	0.2721	0.2721	0.2721
7.505 ppm	-0.0669	-0.0418	-0.1026	-0.0236	-0.0236	-0.0236
7.285 ppm	0.8195	0.8162	0.8818	0.7666	0.7666	0.7666
7.145 ppm	-0.0741	-0.1154	-0.1086	-0.1152	-0.1152	-0.1152
7.075 ppm	-0.2136	-0.2171	-0.0835	-0.1871	-0.1871	-0.1871
6.965 ppm	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
4.565 ppm	0.0280	0.0031	0.0038	0.0000	0.0000	0.0000
4.455 ppm	-0.0025	-0.0415	0.0000	-0.0581	-0.0581	-0.0581
4.445 ppm	0.0000	-0.0457	0.0000	-0.0640	-0.0640	-0.0640
4.375 ppm	-0.0560	-0.2278	-0.1388	-0.2312	-0.2312	-0.2312
4.355 ppm	0.0000	0.0000	0.0047	0.0000	0.0000	0.0000
4.305 ppm	0.8979	0.8953	0.8404	0.8548	0.8548	0.8548
4.225 ppm	0.1607	0.0791	0.1218	0.0533	0.0533	0.0533
4.045 ppm	0.0612	0.0282	0.0227	0.0255	0.0255	0.0255
3.885 ppm	0.0000	0.0135	0.0000	0.0332	0.0332	0.0332
3.695 ppm	0.0010	0.0442	0.0000	0.0000	0.0000	0.0000
1.925 ppm	0.0000	-0.0814	0.0000	-0.0943	-0.0943	-0.0943
1.195 ppm	0.0816	0.0499	0.0885	0.0267	0.0267	0.0267

Table B.3: Shown are the non-zero coefficients of the spectral regions selected by the normal and zero-sum logistic regression applied on the total area, PQN, and TSP normalized plasma AKI data.

B.3 Coefficients of the Regression of Section 5.3 on the Plasma AKI Metabolomics Data

spectral region	coef.	spectral region	coef.	spectral region	coef.	spectral region	coef.
Intercept	-0.87431	7.834 ppm	0.00377	7.413 ppm	0.00115	6.807 ppm	-0.00168
8.625 ppm	-0.00360	7.833 ppm	0.00377	7.412 ppm	0.00115	6.713 ppm	0.00138
8.624 ppm	-0.00360	7.832 ppm	0.00377	7.411 ppm	0.00115	6.712 ppm	0.00138
8.623 ppm	-0.00361	7.831 ppm	0.00377	7.410 ppm	0.00115	6.711 ppm	0.00138
8.622 ppm	-0.00361	7.830 ppm	0.00377	7.409 ppm	0.00115	6.710 ppm	0.00138
8.621 ppm	-0.00361	7.829 ppm	0.00377	7.408 ppm	0.00115	6.709 ppm	0.03126
8.238 ppm	-0.00230	7.828 ppm	0.00377	7.407 ppm	0.00115	6.708 ppm	0.03126
8.237 ppm	-0.00230	7.827 ppm	0.00377	7.406 ppm	0.00115	6.707 ppm	0.03126
8.236 ppm	-0.00230	7.807 ppm	-0.00313	7.405 ppm	0.00115	6.706 ppm	0.03126
8.235 ppm	-0.00230	7.806 ppm	-0.00313	7.404 ppm	0.00115	6.705 ppm	0.03126
8.234 ppm	-0.00230	7.805 ppm	-0.00313	7.403 ppm	0.00115	6.704 ppm	0.03126
8.233 ppm	-0.00230	7.804 ppm	-0.00313	7.402 ppm	0.00115	6.703 ppm	0.03126
8.232 ppm	-0.00230	7.803 ppm	-0.00313	7.401 ppm	0.00115	6.702 ppm	0.03126
8.195 ppm	-0.00107	7.802 ppm	-0.00313	7.400 ppm	0.00115	6.701 ppm	0.03126
8.194 ppm	-0.00107	7.801 ppm	-0.00313	7.399 ppm	0.00115	6.700 ppm	0.03126
8.193 ppm	-0.00107	7.800 ppm	-0.00313	7.398 ppm	0.00115	6.519 ppm	0.00114
8.192 ppm	-0.00107	7.799 ppm	-0.00313	7.397 ppm	0.00115	6.518 ppm	0.00114
8.191 ppm	-0.00107	7.798 ppm	-0.00313	7.396 ppm	0.00115	6.517 ppm	0.00114
8.190 ppm	-0.00107	7.797 ppm	-0.00315	7.289 ppm	0.00107	6.516 ppm	0.00114
8.189 ppm	-0.00108	7.796 ppm	-0.00323	7.288 ppm	0.00107	6.515 ppm	0.00114
8.188 ppm	-0.00108	7.795 ppm	-0.00323	7.287 ppm	0.00107	6.514 ppm	0.00114
8.187 ppm	-0.00108	7.794 ppm	-0.00323	7.286 ppm	0.00107	4.473 ppm	-0.00146
8.186 ppm	-0.00108	7.793 ppm	-0.00346	7.285 ppm	0.00107	4.472 ppm	-0.00146
8.185 ppm	-0.00108	7.792 ppm	-0.00364	7.284 ppm	0.00107	4.471 ppm	-0.00146
8.184 ppm	-0.00108	7.791 ppm	-0.00364	7.283 ppm	0.00107	4.470 ppm	-0.00146
8.183 ppm	-0.00108	7.790 ppm	-0.00364	7.282 ppm	0.00107	4.469 ppm	-0.00146
8.182 ppm	-0.00108	7.789 ppm	-0.00364	7.281 ppm	0.00107	4.468 ppm	-0.00149
8.181 ppm	-0.00108	7.788 ppm	-0.03075	7.280 ppm	0.00107	4.467 ppm	-0.00149
8.180 ppm	-0.00108	7.787 ppm	-0.03075	7.279 ppm	0.00107	4.466 ppm	-0.00149
7.847 ppm	0.00101	7.786 ppm	-0.03075	7.278 ppm	0.00107	4.465 ppm	-0.00149
7.846 ppm	0.00101	7.785 ppm	-0.03075	7.277 ppm	0.00107	4.464 ppm	-0.00149
7.845 ppm	0.00101	7.784 ppm	-0.03075	7.276 ppm	0.00107	4.463 ppm	-0.00149
7.844 ppm	0.00101	7.783 ppm	-0.03075	7.275 ppm	0.00107	4.462 ppm	-0.00149
7.843 ppm	0.00101	7.782 ppm	-0.03075	7.274 ppm	0.00107	4.461 ppm	-0.00149
7.842 ppm	0.00101	7.781 ppm	-0.03075	7.273 ppm	0.00107	4.460 ppm	-0.00149
7.841 ppm	0.00101	7.780 ppm	-0.03075	7.272 ppm	0.00107	4.459 ppm	-0.00225
7.840 ppm	0.00101	7.779 ppm	-0.03075	7.271 ppm	0.00107	4.458 ppm	-0.00225
7.839 ppm	0.00377	7.778 ppm	-0.03035	7.270 ppm	0.00107	4.457 ppm	-0.00225
7.838 ppm	0.00377	7.777 ppm	-0.03035	7.269 ppm	0.00107	4.456 ppm	-0.00225
7.837 ppm	0.00377	7.416 ppm	0.00115	6.810 ppm	-0.00211	4.455 ppm	-0.00225
7.836 ppm	0.00377	7.415 ppm	0.00115	6.809 ppm	-0.00211	4.454 ppm	-0.00225
7.835 ppm	0.00377	7.414 ppm	0.00115	6.808 ppm	-0.00211	4.453 ppm	-0.00320

Table B.4: Shown is the first part of the non-zero coefficients, which have an larger absolute value than 0.001, selected by the zero-sum fused logistic regression applied on the plasma AKI data.

spectral region	coef.	spectral region	coef.	spectral region	coef.
4.452 ppm	-0.00320	4.330 ppm	0.00673	4.288 ppm	0.00119
4.451 ppm	-0.00320	4.329 ppm	0.00673	4.287 ppm	0.00119
4.450 ppm	-0.00320	4.328 ppm	0.00673	4.286 ppm	0.00119
4.449 ppm	-0.00320	4.327 ppm	0.00673	4.233 ppm	0.00261
4.448 ppm	-0.00320	4.326 ppm	0.00673	4.232 ppm	0.00261
4.447 ppm	-0.00320	4.325 ppm	0.00673	4.231 ppm	0.00261
4.446 ppm	-0.00320	4.324 ppm	0.00673	4.230 ppm	0.00272
4.445 ppm	-0.00320	4.323 ppm	0.00673	4.229 ppm	0.00272
4.444 ppm	-0.00320	4.322 ppm	0.00673	4.228 ppm	0.00272
4.443 ppm	-0.00320	4.321 ppm	0.00673	4.227 ppm	0.00188
4.442 ppm	-0.00320	4.320 ppm	0.00673	4.226 ppm	0.00188
4.378 ppm	-0.02011	4.319 ppm	0.00673	4.225 ppm	0.00182
4.377 ppm	-0.02011	4.318 ppm	0.00673	4.224 ppm	0.00182
4.376 ppm	-0.02011	4.317 ppm	0.00673	4.223 ppm	0.00182
4.375 ppm	-0.02011	4.316 ppm	0.00673	4.222 ppm	0.00182
4.374 ppm	-0.02011	4.315 ppm	0.00673	4.221 ppm	0.00182
4.373 ppm	-0.02011	4.314 ppm	0.00673	4.220 ppm	0.00182
4.372 ppm	-0.02011	4.313 ppm	0.00673	4.219 ppm	0.00182
4.371 ppm	-0.02011	4.312 ppm	0.00673	4.218 ppm	0.00182
4.370 ppm	-0.02011	4.311 ppm	0.00673	4.217 ppm	0.00182
4.369 ppm	-0.02011	4.310 ppm	0.00673	4.216 ppm	0.00182
4.368 ppm	-0.01613	4.309 ppm	0.00673	4.215 ppm	0.00182
4.367 ppm	-0.01613	4.308 ppm	0.00673	4.214 ppm	0.00182
4.366 ppm	-0.01613	4.307 ppm	0.00673	4.213 ppm	0.00182
4.348 ppm	0.00195	4.306 ppm	0.00673	4.212 ppm	0.00182
4.347 ppm	0.00195	4.305 ppm	0.00673	4.211 ppm	0.00182
4.346 ppm	0.00195	4.304 ppm	0.00673	4.210 ppm	0.00174
4.345 ppm	0.00195	4.303 ppm	0.00673	4.209 ppm	0.00174
4.344 ppm	0.00673	4.302 ppm	0.00673	4.208 ppm	0.00174
4.343 ppm	0.00673	4.301 ppm	0.00673	4.207 ppm	0.00174
4.342 ppm	0.00673	4.300 ppm	0.00673	4.206 ppm	0.00174
4.341 ppm	0.00673	4.299 ppm	0.00673	4.205 ppm	0.00174
4.340 ppm	0.00673	4.298 ppm	0.00673	4.204 ppm	0.00174
4.339 ppm	0.00673	4.297 ppm	0.00673	4.203 ppm	0.00174
4.338 ppm	0.00673	4.296 ppm	0.00673	4.202 ppm	0.00174
4.337 ppm	0.00673	4.295 ppm	0.00673	4.201 ppm	0.00174
4.336 ppm	0.00673	4.294 ppm	0.00673	4.200 ppm	0.00174
4.335 ppm	0.00673	4.293 ppm	0.00654	4.199 ppm	0.00174
4.334 ppm	0.00673	4.292 ppm	0.00119	4.198 ppm	0.00174
4.333 ppm	0.00673	4.291 ppm	0.00119	4.197 ppm	0.00174
4.332 ppm	0.00673	4.290 ppm	0.00119		
4.331 ppm	0.00673	4.289 ppm	0.00119		

Table B.5: Shown is the second part of the non-zero coefficients, which have an larger absolute value than 0.001, selected by the zero-sum fused logistic regression applied on the plasma AKI data.

B.4 Predicted Probabilities of the Multinomial Regression of Section 5.4 on the DNA Methylation Data

sample	metastasis	primary tumor	breast	lung
`X3145_MET_LUNG_2`	0.933	0.061	0.002	0.004
`X3769_MET_LUNG_2`	0.883	0.113	0.003	0.002
`X3769_MET_LUNG_3`	0.764	0.229	0.004	0.003
`X3770_MET_LUNG_2`	0.129	0.824	0.027	0.020
`X3770_MET_LUNG_3`	0.258	0.723	0.010	0.009
`X3620_MET_LUNG_2`	0.369	0.619	0.008	0.004
`X3620_MET_LUNG_3`	0.448	0.525	0.019	0.007
`X3315_MET_LUNG_2`	0.964	0.033	0.002	0.000
`X3315_MET_LUNG_3`	0.596	0.332	0.045	0.028
`X3522_MET_LUNG_2`	0.698	0.242	0.017	0.043
`X3644_MET_LUNG_2`	0.187	0.801	0.009	0.003
`X3794_MET_LUNG_2`	0.899	0.093	0.006	0.002
`X5419_PT`	0.076	0.918	0.004	0.001
`X5419_MET_LUNG_1`	0.857	0.137	0.004	0.001
`X5423_PT`	0.034	0.955	0.009	0.002
`X5423_MET_LUNG_1`	0.974	0.024	0.001	0.000
`X5423_MET_LUNG_2`	0.994	0.004	0.001	0.000
`X5423_MET_LUNG_3`	0.986	0.012	0.001	0.001
`X3524_PT`	0.474	0.455	0.018	0.053
`X3524_MET_LUNG_1`	0.531	0.454	0.008	0.007
`X3525_PT`	0.886	0.103	0.002	0.009
`X3525_MET_LUNG_1`	0.611	0.358	0.011	0.020
`X3525_MET_LUNG_2`	0.401	0.498	0.013	0.088
`X3525_MET_LUNG_3`	0.861	0.134	0.004	0.002
`X3569_PT`	0.215	0.773	0.009	0.004
`X3569_MET_LUNG_1`	0.669	0.317	0.008	0.005
`X3576_PT`	0.082	0.910	0.006	0.002
`X3576_MET_LUNG_1`	0.936	0.060	0.003	0.001
`X3639_PT`	0.213	0.778	0.007	0.001
`X3639_MET_LUNG_1`	0.782	0.211	0.006	0.002
`X3703_PT`	0.040	0.263	0.686	0.011
`X3703_MET_LUNG_1`	0.959	0.040	0.001	0.000
`X3743_PT`	0.055	0.936	0.009	0.001
`X3743_MET_LUNG_1`	0.227	0.210	0.233	0.330
`X3890_PT`	0.045	0.951	0.004	0.001
`X3890_MET_LUNG_1`	0.968	0.030	0.002	0.000
`X3519_PT`	0.120	0.824	0.014	0.042
`X3555_PT`	0.118	0.872	0.008	0.002
`X3599_PT`	0.087	0.907	0.004	0.001
`X3643_PT`	0.205	0.783	0.010	0.002

Table B.6: Shown is the first part of the obtained cross-validation probabilities for each sample determined using multinomial zero-sum regression.

sample	metastasis	primary tumor	breast	lung
X3640_PT	0.075	0.920	0.004	0.001
X3651_PT	0.092	0.901	0.005	0.001
X3692_PT	0.059	0.926	0.014	0.001
X3676_PT	0.333	0.660	0.005	0.001
X3719_PT	0.076	0.916	0.004	0.004
X3612_PT	0.020	0.976	0.004	0.001
X3615_PT	0.284	0.711	0.003	0.001
X3500_PT	0.070	0.428	0.123	0.379
X3732_PT	0.025	0.086	0.876	0.013
X3744_PT	0.084	0.907	0.008	0.002
X3697_PT	0.080	0.900	0.018	0.002
X3637_PT	0.029	0.948	0.022	0.001
X3145_PT	0.841	0.076	0.048	0.035
X3145_MET_LUNG_1	0.921	0.075	0.003	0.001
X3769_PT	0.851	0.142	0.004	0.003
X3769_MET_LUNG_1	0.865	0.131	0.002	0.002
X3770_PT	0.108	0.863	0.021	0.009
X3770_MET_LUNG_1	0.545	0.439	0.011	0.005
X3620_PT	0.118	0.874	0.006	0.002
X3620_MET_LUNG_1	0.468	0.522	0.005	0.005
X3315_PT	0.078	0.898	0.021	0.003
X3315_MET_LUNG_1	0.807	0.165	0.018	0.011
X3511_PT	0.135	0.859	0.004	0.002
X3511_MET_LUNG_1	0.263	0.708	0.027	0.003
X3522_PT	0.376	0.327	0.031	0.266
X3522_MET_LUNG_1	0.919	0.068	0.005	0.008
X3635_PT	0.060	0.934	0.005	0.001
X3635_MET_LUNG_1	0.628	0.367	0.004	0.001
X3700_PT	0.038	0.844	0.029	0.089
X3700_MET_LUNG_1	0.254	0.627	0.105	0.015
X3699_PT	0.005	0.061	0.814	0.120
X3699_MET_LUNG_1	0.617	0.093	0.041	0.249
X3644_PT	0.129	0.862	0.007	0.002
X3644_MET_LUNG_1	0.698	0.300	0.002	0.000
X3794_PT	0.153	0.831	0.013	0.003
X3794_MET_LUNG_1	0.954	0.045	0.001	0.000
X3631_PT	0.019	0.975	0.005	0.001
X3631_MET_LUNG_1	0.895	0.100	0.004	0.001
X3627_PT	0.931	0.067	0.002	0.001
X3627_MET_LUNG_1	0.905	0.093	0.001	0.001

Table B.7: Shown is the second part of the obtained cross-validation probabilities for each sample determined using multinomial zero-sum regression.

B.5 List of Used Hardware & Software

used workstations:

- **rhskl3:** 2× Intel(R) Xeon(R) CPU E5-2660 v4 @ 2.00GHz (in total 28 cores) debian 9
- **rhskl11:** 4× Intel(R) Xeon(R) CPU E5-4620 0 @ 2.20GHz (in total 32 cores), debian 8

used software:

- **gcc** 4.9.2, 6.3.0, 7.2.0
- **R** 3.4.2, 3.4.3, 3.4.4
- **R-packages:** doMC (1.3.4), foreach (1.4.3), glmnet (2.0-13), HandTill2001 (0.2-12), lattice (0.20-35), Matrix (1.2-11), pROC (1.10.0), rgl (0.99.16), stringr (1.2.0), tikzDevice (0.10-1), VennDiagram (1.6.18), xtable (1.82)

References

[1] B. Alberts, A. Johnson, J. Lewis, D. Morgan, M. Raff, K. Roberts, and P. Walter. *Molecular Biology of the Cell, Sixth Edition*. Garland Science, 2014. ISBN 9780815344322.

[2] A. A. Alizadeh, M. B. Eisen, R. E. Davis, C. Ma, I. S. Lossos, A. Rosenwald, J. C. Boldrick, H. Sabet, T. Tran, X. Yu, J. I. Powell, L. Yang, G. E. Marti, T. Moore, J. J. Hudson, L. Lu, D. B. Lewis, R. Tibshirani, G. Sherlock, W. C. Chan, T. C. Greiner, D. D. Weisenburger, J. O. Armitage, R. Warnke, R. Levy, W. Wilson, M. R. Grever, J. C. Byrd, D. Botstein, P. O. Brown, and L. M. Staudt. Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, 403(6769):503–511, 2000.

[3] M. Altenbuchinger, T . Rehberg, H. U. Zacharias, F. Stämmler, K. Dettmer, D. Weber, A. Hiergeist, A. Gessner, E. Holler, P. J. Oefner, and R. Spang. Reference point insensitive molecular data analysis. *Bioinformatics*, 33(2):219, 2017. doi: 10.1093/bioinformatics/btw598.

[4] M. Altenbuchinger, P. Schwarzfischer, T. Rehberg, J. Reinders, Ch. W. Kohler, W. Gronwald, J. Richter, M. Szczepanowski, N. Masqué-Soler, W. Klapper, P. J. Oefner, and R. Spang. Molecular signatures that can be transferred across different omics platforms. *Bioinformatics*, 33(14): i333–i340, 2017. doi: 10.1093/bioinformatics/btx241.

[5] P. E. Anderson, D. A. Mahle, T. E. Doom, N. V. Reo, N. J. DelRaso, and M. L. Raymer. Dynamic adaptive binning: an improved quantification technique for nmr spectroscopic data. *Metabolomics*, 7(2):179–190, Jun 2011. ISSN 1573-3890. doi: 10.1007/s11306-010-0242-7.

[6] P. C. Austin. Generating survival times to simulate cox proportional hazards models with time-varying covariates. *Statistics in Medicine*, 31(29):3946–3958, 2012. ISSN 1097-0258. doi: 10. 1002/sim.5452.

[7] S. Bates and R. Tibshirani. Log-ratio Lasso: Scalable, Sparse Estimation for Log-ratio Models. *ArXiv e-prints*, September 2017.

[8] J. K. Bradley, A. Kyrola, D. Bickson, and C. Guestrin. Parallel coordinate descent for l1-regularized loss minimization. *CoRR*, abs/1105.5379, 2011.

[9] N. E. Breslow. Contribution to the discussion of the paper by D.R. Cox. *Journal of the Royal Statistical Society. Series B (Methodological)*, 34(2):216–217, 1972. ISSN 00359246.

[10] L. Cascione, A. Rinaldi, A. Chiappella, I. Kwee, G. Ciccone, M. Altenbuchinger, C. Kohler, U. Vitolo, G. Inghirami, and F. Bertoni. Diffuse large b cell lymphoma cell of origin by digital expression profiling in the real07 phase 1–2 study. *British Journal of Haematology*. ISSN 1365-2141. doi: 10.1111/bjh.14817.

[11] A. Chawade, E. Alexandersson, and F. Levander. Normalyzer: A tool for rapid evaluation of normalization methods for omics data sets. *Journal of Proteome Research*, 13(6):3114–3120, 2014. doi: 10.1021/pr401264n.

[12] R. Chen and M. Snyder. Promise of personalized omics to precision medicine. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine*, 5(1):73–82, 2013. ISSN 1939-005X. doi: 10.1002/wsbm.1198.

[13] R. Chen, G. I. Mias, J. Li-Pook-Than, L. Jiang, H. Y.K. Lam, R. Chen, E. Miriami, K. J. Karczewski, M. Hariharan, F. E. Dewey, et al. Personal omics profiling reveals dynamic molecular and medical phenotypes. *Cell*, 148(6):1293–1307, 2012.

[14] F. S. Collins and H. Varmus. A new initiative on precision medicine. *New England Journal of Medicine*, 372(9):793–795, 2015. doi: 10.1056/NEJMp1500523. PMID: 25635347.

[15] D. R. Cox. Regression models and life-tables. *Journal of the Royal Statistical Society. Series B (Methodological)*, 34(2):187–220, 1972. ISSN 00359246.

[16] A. Craig, O. Cloarec, E. Holmes, J.K. Nicholson, and J.C. Lindon. Scaling and normalization effects in nmr spectroscopic metabonomic data sets. *Analytical Chemistry*, 78(7):2262–2267, 2006. doi: 10.1021/ac0519312.

[17] A. D. Cullmann. *HandTill2001: Multiple Class Area under ROC Curve*, 2016. URL `https://CRAN.R-project.org/package=HandTill2001`. R package version 0.2-12.

[18] G. Curhan. Cystatin c: A marker of renal function or something more? *Clinical Chemistry*, 51(2): 293–294, 2005. ISSN 0009-9147. doi: 10.1373/clinchem.2004.044388.

[19] S. Datta and B. JA. Mertens. *Statistical Analysis of Proteomics, Metabolomics, and Lipidomics Data Using Mass Spectrometry*. Springer, 2017.

[20] J. B. de Kok, R. W. Roelofs, B. A. Giesendorf, J. L. Pennings, E. T. Waas, T. Feuth, D. W. Swinkels, and P. N. Span. Normalization of gene expression measurements in tumor tissues: comparison of 13 endogenous control genes. *Laboratory investigation*, 85(1):154–159, 2005.

[21] T. De Meyer, D. Sinnaeve, B. Van Gasse, E. Tsiporkova, E. R. Rietzschel, M. L. De Buyzere, T. C. Gillebert, S. Bekaert, J. C. Martins, and W. Van Criekinge. Nmr-based characterization of metabolic alterations in hypertension using an adaptive, intelligent binning algorithm. *Analytical Chemistry*, 80(10):3783–3790, 2008. doi: 10.1021/ac7025964. PMID: 18419139.

[22] F. Dieterle, A. Ross, G. Schlotterbeck, and H. Senn. Probabilistic quotient normalization as robust method to account for dilution of complex biological mixtures. application in 1h nmr metabonomics. *Analytical Chemistry*, 78(13):4281–4290, 2006. doi: 10.1021/ac051632c.

[23] E. Eisenberg and E. Y. Levanon. Human housekeeping genes, revisited. *Trends in Genetics*, 29 (10):569–574, 2013.

[24] J. LM. Ferrara, J. E. Levine, P. Reddy, and E. Holler. Graft-versus-host disease. *The Lancet*, 373 (9674):1550–1561, 2009.

[25] J. Friedman, T. Hastie, H. Höfling, and R. Tibshirani. Pathwise coordinate optimization. *The Annals of Applied Statistics*, 1(2):302–332, 12 2007. doi: 10.1214/07-AOAS131.

[26] J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010. ISSN 1548-7660. doi: 10.18637/jss.v033.i01.

[27] R. L. Grossman, A. P. Heath, V. Ferretti, H. E. Varmus, D. R. Lowy, W. A. Kibbe, and L. M. Staudt. Toward a shared vision for cancer genomic data. *New England Journal of Medicine*, 375 (12):1109–1112, 2016. doi: 10.1056/NEJMp1607591.

[28] M. A. Hamburg and F. S. Collins. The path to personalized medicine. *New England Journal of Medicine*, 363(4):301–304, 2010. doi: 10.1056/NEJMp1006304. PMID: 20551152.

[29] D. J. Hand and R. J. Till. A simple generalisation of the area under the roc curve for multiple class classification problems. *Machine Learning*, 45(2):171–186, Nov 2001. ISSN 1573-0565. doi: 10.1023/A:1010920819831.

[30] T. Hastie and J. Qian. *Glmnet Vignette*, September 2016. URL `https://web.stanford.edu/~hastie/Papers/Glmnet_Vignette.pdf`.

[31] T. Hastie and R. Tibshirani. *Generalized Additive Models*. Chapman Hall & CRC/Monographs on Statistics & Applied Probability. Chapman and Hall/CRC, 1 edition, 1990. ISBN 9780412343902,0412343908.

[32] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. Springer Series in Statistics. Springer, 2009. ISBN 9780387848587.

[33] W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[34] A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970. doi: 10.1080/00401706.1970.10488634.

[35] E. Holler, P. Butzhammer, K. Schmid, C. Hundsrucker, J. Koestler, K. Peter, W. Zhu, D. Sporrer, T. Hehlgans, M. Kreutz, et al. Metagenomic analysis of the stool microbiome in patients receiving allogeneic stem cell transplantation: loss of diversity is associated with use of systemic antibiotics and more pronounced in gastrointestinal graft-versus-host disease. *Biology of Blood and Marrow Transplantation*, 20(5):640–645, 2014.

[36] R. Hornung, D. Causeur, C. Bernau, and A.L. Boulesteix. Improving cross-study prediction through addon batch effect adjustment or addon normalization. *Bioinformatics*, 33(3):397–404, 2017. doi: 10.1093/bioinformatics/btw650.

[37] H. Hosseini, M. M. S. Obradović, M. Hoffmann, K. L. Harper, M. S. Sosa, M. Werner-Klein, L. K. Nanduri, C. Werno, C. Ehrl, M. Maneck, N. Patwary, G. Haunschild, M. Gužvić, C. Reimelt, M. Grauvogl, N. Eichner, F. Weber, A. D. Hartkopf, F.-A. Taran, S. Y. Brucker, T. Fehm, B. Rack, S. Buchholz, R. Spang, G. Meister, J. A. Aguirre-Ghiso, and C. A. Klein. Early dissemination seeds metastasis in breast cancer. *Nature*, December 2016. ISSN 0028-0836. doi: 10.1038/nature20785.

[38] K. Hukushima and K. Nemoto. Exchange Monte Carlo Method and Application to Spin Glass Simulations. *Journal of the Physical Society of Japan*, 65:1604, June 1996. doi: 10.1143/JPSJ.65.1604.

[39] G. James, D. Witten, T. Hastie, and R. Tibshirani. *An Introduction to Statistical Learning: With Applications in R*. Springer Publishing Company, Incorporated, 2014. ISBN 1461471370, 9781461471370.

[40] J. L. Jameson and D. L. Longo. Precision medicine — personalized, problematic, and promising. *New England Journal of Medicine*, 372(23):2229–2234, 2015. doi: 10.1056/NEJMsb1503104. PMID: 26014593.

[41] E. Jones, T. Oliphant, P. Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. URL `http://www.scipy.org/`. [Online; accessed ¡today¿].

[42] A. R. Joyce and B. Ø. Palsson. The model organism as a system: integrating 'omics' data sets. *Nature Reviews Molecular Cell Biology*, 7(3):198–210, 2006.

[43] J. Jurczyk, T. Rehberg, A. Eckrot, and I. Morgenstern. Measuring critical transitions in financial markets. *Scientific Reports*, 7(1):11564, 2017.

[44] S. Kirkpatrick. Optimization by simulated annealing: Quantitative studies. *Journal of Statistical Physics*, 34(5):975–986, Mar 1984. ISSN 1572-9613. doi: 10.1007/BF01009452.

[45] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220 (4598):671–680, 1983. doi: 10.1126/science.220.4598.671.

[46] D. Kostka and R. Spang. Microarray based diagnosis profits from better documentation of gene expression signatures. *PLOS Computational Biology*, 4(2):1–6, 02 2008. doi: 10.1371/journal.pcbi.0040022.

[47] B. Krishnapuram, L. Carin, M. Figueiredo, and A.J. Hartemink. Sparse multinomial logistic regression: Fast algorithms and generalization bounds. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(6):957–968, 2005.

[48] D. Krstajic, L. J. Buturovic, D. E. Leahy, and S. Thomas. Cross-validation pitfalls when selecting and assessing regression and classification models. *Journal of Cheminformatics*, 6(1):10, Mar 2014. ISSN 1758-2946. doi: 10.1186/1758-2946-6-10.

[49] O. M. Kvalheim, F. Brakstad, and Y. Liang. Preprocessing of analytical profiles in the presence of homoscedastic or heteroscedastic noise. *Analytical Chemistry*, 66(1):43–51, 1994. doi: 10.1021/ac00073a010.

[50] S. Lemery, P. Keegan, and R. Pazdur. First fda approval agnostic of cancer site — when a biomarker defines the indication. *New England Journal of Medicine*, 377(15):1409–1412, 2017. doi: 10.1056/NEJMp1709968. PMID: 29020592.

[51] G. Lenz, G. Wright, S. S. Dave, W. Xiao, J. Powell, H. Zhao, W. Xu, B. Tan, N. Goldschmidt, J. Iqbal, J. Vose, M. Bast, K. Fu, D. D. Weisenburger, T. C. Greiner, J. O. Armitage, A. Kyle, L. May, R. D. Gascoyne, J. M. Connors, G. Troen, H. Holte, S. Kvaloy, D. Dierickx, G. Verhoef, J. Delabie, E. B. Smeland, P. Jares, A. Martinez, A. Lopez-Guillermo, E. Montserrat, E. Campo, R. M. Braziel, T. P. Miller, L. M. Rimsza, J. R. Cook, B. Pohlman, J. Sweetenham, R. R. Tubbs, R. I. Fisher, E. Hartmann, A. Rosenwald, G. Ott, H.-K. Muller-Hermelink, D. Wrench, T. A. Lister, E. S. Jaffe, W. H. Wilson, W. C. Chan, and L. M. Staudt. Stromal gene signatures in large-b-cell lymphomas. *New England Journal of Medicine*, 359(22):2313–2323, 2008. doi: 10.1056/NEJMoa0802885. PMID: 19038878.

[52] C. Y. Lin, J. Lovén, P. B. Rahl, R. M. Paranal, C. B. Burge, J. E. Bradner, T. I. Lee, and R. A. Young. Transcriptional amplification in tumor cells with elevated c-myc. *Cell*, 151(1):56–67, 2012.

[53] W. Lin, P. Shi, R. Feng, and H. Li. Variable selection in regression with compositional covariates. *Biometrika*, 101(4):785–797, 2014. doi: 10.1093/biomet/asu031.

[54] M. I. Love, W. Huber, and S. Anders. Moderated estimation of fold change and dispersion for rna-seq data with deseq2. *Genome Biology*, 15(12):550, Dec 2014. ISSN 1474-760X. doi: 10.1186/s13059-014-0550-8.

[55] L. Meier, S. Van De Geer, and P. Bühlmann. The group lasso for logistic regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(1):53–71, 2008.

[56] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

[57] D. Mitra, F. Romeo, and A. Sangiovanni-Vincentelli. Convergence and finite-time behavior of simulated annealing. In *Decision and Control, 1985 24th IEEE Conference on*, volume 24, pages 761–767, Dec 1985. doi: 10.1109/CDC.1985.268600.

[58] A. J. Nelder and R. W. M. Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society. Series A (General)*, 135(3):370–384, 1972. ISSN 00359238.

[59] M. E. J. Newman and G. T. Barkema. *Monte Carlo Methods in Statistical Physics*. Oxford University Press, 2008.

[60] A. Y. Ng. Feature selection, l1 vs. l2 regularization, and rotational invariance. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, pages 78–, New York, NY, USA, 2004. ACM. ISBN 1-58113-838-5. doi: 10.1145/1015330.1015435.

[61] Z. Nie, G. Hu, G. Wei, K. Cui, A. Yamane, W. Resch, R. Wang, D.R. Green, L. Tessarollo, R. Casellas, et al. c-myc is a universal amplifier of expressed genes in lymphocytes and embryonic stem cells. *Cell*, 151(1):68–79, 2012.

[62] I. Oschlies, C. W. Kohler, M. Szczepanowski, K. Koch, A. Gontarewicz, D. Metze, U. Hillen, J. Richter, R. Spang, and W. Klapper. Spindle-cell variants of primary cutaneous follicle center b-cell lymphomas are germinal center b-cell lymphomas by gene expression profiling using a formalin-fixed paraffin-embedded specimen. *The Journal of investigative dermatology*, 137(11): 2450, 2017.

[63] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[64] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 3 edition, 2007. ISBN 0521880688, 9780521880688.

[65] J. Quackenbush. Microarray data normalization and transformation. *Nature genetics*, 32:496–501, 2002.

[66] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2017. URL `https://www.R-project.org/`.

[67] A. Rosenwald, G. Wright, W. C. Chan, J. M. Connors, E. Campo, R. I. Fisher, R. D. Gascoyne, H. K. Muller-Hermelink, E. B. Smeland, J. M. Giltnane, E. M. Hurt, H. Zhao, L. Averett, L. Yang, W. H. Wilson, E. S. Jaffe, R. Simon, R. D. Klausner, J. Powell, P. L. Duffey, D. L. Longo, T. C. Greiner, D. D. Weisenburger, W. G. Sanger, B. J. Dave, J. C. Lynch, J. Vose, J. O. Armitage, E. Montserrat, A. López-Guillermo, T. M. Grogan, T. P. Miller, M. LeBlanc, G. Ott, S. Kvaloy, J. Delabie, H. Holte, P. Krajci, T. Stokke, and L. M. Staudt. The use of molecular profiling to predict survival after chemotherapy for diffuse large-b-cell lymphoma. *New England Journal of Medicine*, 346(25):1937–1947, 2002. doi: 10.1056/NEJMoa012914. PMID: 12075054.

[68] D. Ryan, K. Robards, P. D. Prenzler, and M.. Kendall. Recent and potential developments in the analysis of urine: A review. *Analytica Chimica Acta*, 684(1):17 – 29, 2011. ISSN 0003-2670. doi: http://dx.doi.org/10.1016/j.aca.2010.10.035.

[69] J. J. Schneider and S. Kirkpatrick. *Stochastic Optimization*. Springer-Verlag, Berlin Heidelberg, 2006.

[70] N. Simon, J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for cox's proportional hazards model via coordinate descent. *Journal of Statistical Software*, 39(1):1–13, 2011. ISSN 1548-7660. doi: 10.18637/jss.v039.i05.

[71] F. Stämmler, J. Gläsner, A. Hiergeist, E. Holler, D. Weber, P. J. Oefner, A. Gessner, and R. Spang. Adjusting microbiome profiles for differences in microbial load by spike-in bacteria. *Microbiome*, 4(1):28, Jun 2016. ISSN 2049-2618. doi: 10.1186/s40168-016-0175-0.

[72] L. A. Stevens and A. S. Levey. Measured gfr as a confirmatory test for estimated gfr. *Journal of the American Society of Nephrology*, 20(11):2305–2313, 2009. doi: 10.1681/ASN.2009020171.

[73] M. Szczepanowski, J. Lange, C. W. Kohler, N. Masque-Soler, M. Zimmermann, S. M. Aukema, M. Altenbuchinger, T. Rehberg, F. Mahn, R. Siebert, R. Spang, B. Burkhardt, and W. Klapper. Cell-of-origin classification by gene expression and myc-rearrangements in diffuse large b-cell lymphoma of children and adolescents. *British Journal of Haematology*, 179(1):116–119, 2017. ISSN 1365-2141. doi: 10.1111/bjh.14812.

[74] Franziska Taruttis, Maren Feist, Phillip Schwarzfischer, Wolfram Gronwald, Dieter Kube, Rainer Spang, and Julia C Engelmann. External calibration with drosophila whole-cell spike-ins delivers absolute mrna fold changes from human rna-seq and qpcr data. *BioTechniques*, 62(2):53–61, 2017.

[75] Y. Taur, J. B. Xavier, L. Lipuma, C. Ubeda, J. Goldberg, A. Gobourne, Y. J. Lee, K. A. Dubin, N. D. Socci, A. Viale, M. Perales, R. R. Jenq, M. R. M. van den Brink, and E. G. Pamer. Intestinal domination and the risk of bacteremia in patients undergoing allogeneic hematopoietic stem cell transplantation. *Clinical Infectious Diseases*, 55(7):905–914, 2012.

[76] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996. ISSN 00359246.

[77] R. Tibshirani. The lasso method for variable selection in the cox model. *Statistics in medicine*, 16 (4):385–395, 1997.

[78] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108, 2005. ISSN 1467-9868. doi: 10.1111/j.1467-9868.2005.00490.x.

[79] R. J. Tibshirani and J. Taylor. The solution path of the generalized lasso. *Ann. Statist.*, 39(3): 1335–1371, 06 2011. doi: 10.1214/11-AOS878.

[80] R. A. van den Berg, H. CJ. Hoefsloot, J. A. Westerhuis, A. K. Smilde, and M. J. van der Werf. Centering, scaling, and transformations: improving the biological information content of metabolomics data. *BMC Genomics*, 7(1):142, Jun 2006. ISSN 1471-2164. doi: 10.1186/1471-2164-7-142.

[81] S. S. Waikar, V. S. Sabbisetti, and J. V. Bonventre. Normalization of urinary biomarkers to creatinine during changes in glomerular filtration rate. *Kidney international*, 78(5):486–494, 2010.

[82] B. M. Warrack, S. Hnatyshyn, K.H. Ott, M. D. Reily, M. Sanders, Zhang. H., and D. M. Drexler. Normalization strategies for metabonomic analysis of urine samples. *Journal of Chromatography B*, 877(5):547 – 552, 2009. ISSN 1570-0232. doi: http://dx.doi.org/10.1016/j.jchromb.2009.01.007.

[83] D. Weber, P. J. Oefner, A. Hiergeist, J. Koestler, A. Gessner, M. Weber, J. Hahn, D. Wolff, F. Stämmler, R. Spang, W. Herr, K. Dettmer, and E. Holler. Low urinary indoxyl sulfate levels early after transplantation reflect a disrupted microbiome and are associated with poor outcome. *Blood*, 126(14):1723–1728, 2015.

[84] G. Wright, B. Tan, A. Rosenwald, E. H. Hurt, A. Wiestner, and L. M. Staudt. A gene expression-based method to diagnose clinically distinct subgroups of diffuse large b cell lymphoma. *Proceedings of the National Academy of Sciences*, 100(17):9991–9996, 2003. doi: 10.1073/pnas.1732008100.

[85] D. Yu, S. J. Lee, W. J. Lee, S. C. Kim, J. Lim, and S. W. Kwon. Classification of spectral data using fused lasso logistic regression. *Chemometrics and Intelligent Laboratory Systems*, 142(Supplement C):70 – 77, 2015. ISSN 0169-7439. doi: https://doi.org/10.1016/j.chemolab.2015.01.006.

[86] H. U. Zacharias, G. Schley, J. Hochrein, M. S. Klein, C. Köberle, K.-U. Eckardt, C. Willam, P. J. Oefner, and W. Gronwald. Analysis of human urine reveals metabolic changes related to the development of acute kidney injury following cardiac surgery. *Metabolomics*, 9(3):697–707, Jun 2013. ISSN 1573-3890. doi: 10.1007/s11306-012-0479-4.

[87] H. U. Zacharias, J. Hochrein, F. C. Vogl, G. Schley, F. Mayer, C. Jeleazcov, K.-U. Eckardt, C. Willam, P. J. Oefner, and W. Gronwald. Identification of plasma metabolites prognostic of acute kidney injury after cardiac surgery with cardiopulmonary bypass. *Journal of Proteome Research*, 14(7):2897–2905, 2015. doi: 10.1021/acs.jproteome.5b00219.

[88] H. U. Zacharias, T. Rehberg, S. Mehrl, D. Richtmann, T. Wettig, P. J. Oefner, R. Spang, W. Gronwald, and M. Altenbuchinger. Scale-invariant biomarker discovery in urine and plasma metabolite fingerprints. *Journal of Proteome Research*, 16(10):3596–3605, 2017. doi: 10.1021/acs.jproteome.7b00325.

[89] J. Zhu and T. Hastie. Classification of gene microarrays by penalized logistic regression. *Biostatistics*, 5(3):427–443, 2004. doi: 10.1093/biostatistics/kxg046.

[90] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of The Royal Statistical Society Series B-statistical Methodology*, 67(2):301–320, 2005.

List of Figures

List of Tables