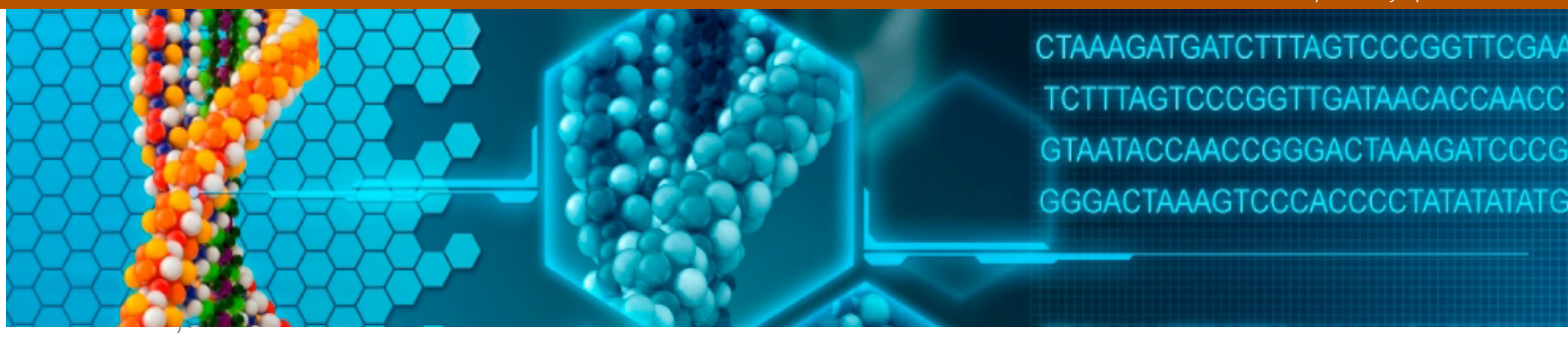


# Master's Thesis

## Clinical Outcome Prediction Based on Multi-Omics Data: Extension of IPF-LASSO

by

**Gerhard Schulze**

Ludwig-Maximilians-Universität München

Institut für Statistik


Supervisor: **Prof. Dr. Anne-Laure Boulesteix**

Institut für Medizinische Informationsverarbeitung

Biometrie und Epidemiologie (IBE)

München, November 9, 2017

Those who have knowledge, don't predict.

Those who predict, don't have knowledge.

*Lao Tzu ($6^{th}$ century B.C.)*

# Abstract

Predicting clinical outcomes via a set of omics data, consisting for example of genomics, proteomics or metabolomics data requires statistical methods which can deal with the situation of a huge number of covariates vs. a relatively small number of observations. One option is the application of penalized regression methods like the Lasso which also results in sparse models because variable selection is part of the model building process. In contrast to the standard Lasso model, the IPF-Lasso penalizes the various omics groups (modalities) of a multi-omics data set individually. However, this advantage has to be paid with exponentially increasing computation time, when there are more than two or three modalities involved. In this study, several two-step modifications of the IPF-Lasso model have been evaluated and compared to the original IPF-Lasso and some other competitors by analyzing simulated and real data sets. The results show that for a given data set, the new two-step IPF-Lasso model needs only a fraction of the computation time of the original IPF-Lasso and nevertheless can compete in terms of prediction performance.

# Contents

# List of Abbreviations

| | |
|---|---|
| AML | Acute Myeloid Leukemia |
| AUC | Area Under the Curve |
| BRCA | Breast Invasive Carcinoma |
| BS | Brier Score |
| CNV | Copy Number Variation |
| CPU | Central Processing Unit |
| CRAN | The Comprehensive R Archive Network |
| CV | Cross Validation |
| DNA | Deoxyribonucleic Acid |
| ESCA | Esophageal Carcinoma |
| IBS | Integrated Brier Score |
| IPF-Lasso | Integrative Lasso with Penalty Factors |
| KIRP | Kidney Renal Papillary Cell Carcinoma |
| LARS | Least Angle Regression Selection |
| LASSO | Least Absolute Shrinkage and Selection Operator |
| LRZ | Leibniz-Rechenzentrum |
| miRNA | micro Ribonucleic Acid |
| NIH | National Institute of Health |
| RAM | Random Access Memory |
| RNA | Ribonucleic Acid |
| ROC | Receiver Operating Characteristic |
| TCGA | The Cancer Genome Atlas |
| TSIPF | Two-step IPF-Lasso model |
| TSIPF1 | Ridge-Lasso/combined modalities in Step 1/arithmetic mean |
| TSIPF2 | Ridge-Lasso/combined modalities in Step 1/geometric mean |
| TSIPF3 | Lasso-Lasso/combined modalities in Step 1/arithmetic mean |
| TSIPF4 | Ridge-Lasso/separate modalities in Step 1/arithmetic mean |
| TSIPF5 | Ridge-Lasso/separate modalities in Step 1/geometric mean |
| TSIPF6 | Lasso-Lasso/separate modalities in Step 1/arithmetic mean |

# List of Figures

# List of Tables

# 1 Introduction

Between the years 2004 and 2010 the cost for DNA sequencing fell dramatically. The number of processed base pairs per US-Dollar doubled every 5 months and thus grew much faster than the amount of available computer disk space per US-Dollar [Stein, 2010]. This opened up new opportunities for research, but also created new challenges in terms of huge amounts of data which need to be analyzed by applying adequate methods. Data can be collected across the entire chain of biochemical processes, starting at the DNA and finally ending at a certain disease. Depending on the origin of the data, one can distinguish between many types of omics data like genomics, epigenomics, transcriptomics, proteomics, metabolomics and microbiomics data [Hasin et al., 2017].

Much effort has been put into developing methods for omics data integration in order to understand interactions and dependencies between the different components of the biochemical processes of a living organism and finally being able to understand the origins and pathways of a disease. Among these methods, there are also a few which specifically aim at prediction of cancer prognosis [Huang et al., 2017]. This study follows a naive approach to data integration by combining a certain number of omics data sets into one huge multi-omics data set which then serves as input for the prediction models to be investigated.

In omics data the number of covariates outnumbers by far the number of samples ($p \gg n$), especially when there are multiple data sets (modalities) involved. Therefore, predicting clinical outcome variables with traditional modeling techniques (e.g. linear or logistic regression) is not possible and there is the need to look for alternatives. One could either adapt the model input and reduce the data dimensions or apply alternative modeling techniques which can deal with this type of data. One class of methods among these alternatives is penalized regression and especially Ridge or Lasso regression [Hoerl and Kennard, 1970],[Tibshirani, 1996] with the latter becoming quite popular during the last years. Compared to Ridge regression, Lasso regression not only shrinks the parameter estimates towards zero, but actually can set parameter estimates equal to zero and thus also does variable selection during the estimation process. When a clinical outcome needs to be predicted based on several modalities, lasso regression equally penalizes each variable

across each modality by providing a single penalization parameter for the entire data set. However, it might make sense to switch from a one-size-fits-all approach to a method which distinguishes between the modalities and finally applies an individual penalty to each modality.

This idea has been realized through the IPF-Lasso model [Boulesteix et al., 2017a]. IPF-Lasso is estimated by first looking at alternative sets of penalty factors, each set containing one factor for each modality. The final model is then based on the set of penalty factors which gives the lowest prediction error as a result of repeated cross-validation. The method has been implemented in an R package called ipflasso which is publicly available on the R/CRAN website [Boulesteix and Fuchs, 2015]. The computation time of ipflasso essentially depends on the number of modalities (M) to be analyzed or how many sets of penalty factors need to be considered and grows exponentially with M [Boulesteix et al., 2017a]. Therefore, running ipflasso for multi-omics data with say more than two modalities on normal hardware like a standard 8GB RAM laptop, quickly becomes a challenge and it makes sense to look for opportunities which remove or at least widen the bottleneck of computation time.

A straightforward solution, but not widely applicable because of cost reasons would be a hardware upgrade. One could also work on modifying the ipflasso software in a way that limited RAM space can be used more efficiently. Another option would be related to the data: One could pre-select a subset of covariates and therefore reduce the dimensionality of the problem which consequently will reduce computation time. The focus of this study is on a fourth option which aims at changing the method itself.

The major change is letting the data decide about a single set of penalty factors in a first analysis step and then in a second step run IPF-Lasso with these penalty factors as input. A similar approach has been applied in the context of identifying biomarker-by-treatment interactions in high-dimensional data [Ternès et al., 2017] and the idea has been mentioned already earlier in a paper by Tai and Pan [2007]. There are three questions to be answered: Can such a two-step IPF-Lasso (TSIPF-Lasso) procedure handle more than two modalities in a reasonable time? Can the TSIPF-Lasso compete with the IPF-Lasso and other modeling approaches in terms

of prediction performance and model sparsity? Which of the six potential versions of the new TSIPF-Lasso is the best in terms of computation time and prediction performance? In this study, the computation time and prediction performance of the two-step IPF-Lasso has been compared to the original Lasso, separate Lasso models for each modality, sparse group Lasso (SGL) and the original IPF-Lasso by analyzing several simulated data scenarios. Additionally, four real data sets from The Cancer Genome Atlas (TCGA) have been analyzed and the results of the TSIPF-Lasso models have been compared to the original Lasso and the separate models approach. The IPF-Lasso and the SGL had to be excluded because of too long computation time and certain software issues. Overall, the comparison showed that the two-step approach needs only a fraction of the IPF-Lasso computation time and has a competitive prediction performance.

The new two-step IPF-Lasso model and its competitors are described in the following section. The analyzed data and how the analysis has been performed is explained in Sections 3 and 4. An overview and a discussion of the results is given in Section 5 and Section 6 contains conclusions from the study and some thoughts for future research.

# 2 Methods

Quite a number of concepts and methods have been developed to manage the p≫n issue of making predictions based on high-dimensional data. An important group of these methods belongs to the area of machine learning. Another group consists of regularized or penalized regression models with the Lasso family as a very popular member [Hastie et al., 2009, Bühlmann and van de Geer, 2011]. This section describes the new two-step IPF-Lasso model, the original IPF-Lasso, the standard Lasso, sparse group Lasso, separate Lasso models for each modality and Ridge regression as optional part of the new method.

## 2.1 Ridge and Lasso regression

The parameter estimate $\hat{\beta}$ of a classical linear regression model as described for example in Fahrmeir et al. [2009] is given by

$$\hat{\beta} = (X'X)^{-1}X'y \tag{1}$$

where X is the design matrix and y a continuous response variable.
As long as X has full rank, $\hat{\beta}$ is the best unbiased estimator for the parameter vector $\beta$. However, with high-dimensional data (p≫n) the matrix X'X is not invertible and $\hat{\beta}$ is no longer uniquely defined. This problem can be avoided by introducing a penalization term which shrinks the parameter estimates towards zero and although the result is no longer unbiased, it might produce an acceptable mean square error.
Ridge regression [Hoerl and Kennard, 1970] uses the $L_2$ norm in a penalization term and estimates the $\beta$ parameter vector by

$$\|y - X\beta\|_2^2 \; + \; \lambda\|\beta\|_2^2 \quad \overset{\beta}{\rightarrow} \quad min \tag{2}$$

with $\|.\|_2$ as $L_2$ norm and $\lambda > 0$ a scalar. The parameter estimate $\hat{\beta}$ for $\beta$ is then given by

$$\hat{\beta} = (X'X + \lambda I)^{-1}X'y \tag{3}$$

where I is the identity matrix.
A high-dimensional Ridge regression model can still contain hundreds of thousands

of covariates as it only shrinks the parameter estimates towards zero, but never sets them actually equal to zero. Setting parameter estimates equal to zero is a way of variable selection in order to get a sparse model. This can be achieved by the Lasso model which uses another penalty term.

Compared to Ridge regression, the penalty term of the Lasso model is built on the $L_1$ norm. The parameter vector $\beta$ is now estimated by

$$\|y - X\beta\|_2^2 \; + \; \lambda\|\beta\|_1 \quad \stackrel{\beta}{\rightarrow} \quad min \tag{4}$$

with $\|.\|_1$ as $L_1$ norm and $\lambda > 0$ a scalar. This optimization problem has no explicit solution like the Ridge model and needs to be solved numerically. The parameter estimates have been originally calculated by a quadratic program solver, some time later by the very efficient LARS algorithm and nowadays also with coordinate descent algorithms [Tibshirani, 2011].

These models can also be applied to binary response variables and to survival data. For the logistic Lasso the term $\|y - X\beta\|_2^2$ has to be replaced by $-l(\beta, \beta_0)$, the negative log-likelihood function of $\beta$ and the intercept $\beta_0$ and for the Cox Lasso through $-pl(\beta)$, the negative partial log-likelihood.

## 2.2 Sparse group Lasso

The scope of the original Lasso model has been expanded to a number of specific modeling situations and different Lasso variants have been developed by modifying the original penalty term [Hastie et al., 2015]. In a scenario where the covariates belong to different groups (like for example in omics-data), it might make sense to look for a solution which only contains a few of these groups. Such a situation can be modeled by the group Lasso approach [Yuan and Lin, 2006]. However, with omics-data one usually expects some of the covariates of a given modality to be important for predicting the outcome and it wouldn't make sense to exclude the entire modality. Thus, a better solution is the sparse group Lasso model which combines the group Lasso and the standard Lasso, making sure that the final result will be a model being sparse within and between groups [Simon et al., 2013b]. Let M be the number of modalities in a multi-omics data set, $X^{(m)}$ the matrix of covariates in modality m (m=1,...,M) and $\beta^{(m)}$ the corresponding model parameter

vector.

The parameter estimate $\hat{\beta}$ of a sparse group Lasso model is calculated by minimizing

$$\frac{1}{2n}\|y - \sum_{m=1}^{M} X^{(m)}\beta^{(m)}\|_2^2 \; + \; (1-\alpha)\lambda \sum_{m=1}^{M} \sqrt{p_m}\|\beta^{(m)}\|_2 \; + \; \alpha\lambda\|\beta\|_1 \quad \stackrel{\beta}{\rightarrow} \quad min \;\; (5)$$

where $\|.\|_1$ and $\|.\|_2$ are the $L_1$ and $L_2$ norms, $p_m$ is the length of modality m, $\lambda > 0$ is a scalar and $\alpha \in [0, 1]$. The factor $\alpha$ controls the emphasis of model sparsity on the group versus the covariate level. With $\alpha = 1$, the model turns into the standard Lasso model and with $\alpha = 0$ it becomes the group Lasso model.

## 2.3 Separate models for each modality

There are various further options to integrate the heterogeneous group structure of multi-omics data into the model building process. One could simply build separate models for each modality and then combine these into an overall model [Zhao et al., 2014]. Another approach includes some pre-knowledge about the prediction capabilities of certain modalities into the estimation process. Especially clinical data is often already known to impact an outcome like cancer prognosis. This part of the data is usually very small compared to molecular measurements like gene expression data and therefore, important clinical covariates might get lost in a model which treats all modalities the same way. In a penalized model, the so called "favoring" approach would only penalize the molecular part of the covariates and leave the clinical covariates un-penalized. The favoring approach and some alternatives are discussed in a paper by Boulesteix and Sauerbrei [2011].

This study follows the naive method by running separate Lasso models for each modality and combining the coefficient estimates into a final model. The model serves as a direct competitor to the new two-step IPF-Lasso and is also used as optional first step model in the new method.

## 2.4 IPF-Lasso

Like the separate Lasso models approach, the "integrative Lasso with penalty factors" (IPF-Lasso) penalizes each modality individually. The parameter estimation is done by

$$\|y - \sum_{m=1}^{M} X^{(m)}\beta^{(m)}\|_2^2 \; + \; \sum_{m=1}^{M} \lambda_m \|\beta^{(m)}\|_1 \quad \xrightarrow{\beta} \quad min \tag{6}$$

where $X^{(m)}$ is the matrix of covariates in modality m (m=1,...,M) and $\beta^{(m)}$ the corresponding model parameter vector. The scalar $\lambda_m > 0$ is the individual penalty for modality m. IPF-Lasso can use the existing Lasso algorithms and estimates $\hat{\beta}_j^{(m)*}$ are obtained by using the same penalty parameter $\lambda_1$ for all covariates on the transformed data ${x_{ij}}^{(m)*} = {x_{ij}}^{(m)}/(\lambda_m/\lambda_1)$ (i=1,...,n and j=1,...,p). The final IPF-Lasso estimates $\hat{\beta}_j^{(m)}$ are then calculated by re-scaling $\hat{\beta}_j^{(m)*}$ as $\hat{\beta}_j^{(m)} = \hat{\beta}_j^{(m)*}/(\lambda_m/\lambda_1)$ [Boulesteix et al., 2017a]. In their paper, Boulesteix et al. [2017a] compared IPF-Lasso to the original Lasso, the separate Lasso models approach and to spares group Lasso. They demonstrated that IPF-Lasso has a better prediction performance than its competitors when the modalities are different in terms of their relevance for prediction. The performance turned out to be slightly worse for similar modalities.

However, from a practical point of view there is an issue related to computation time: In order to get the M penalties $\lambda_m$, a certain number C of candidate vectors of penalty factors $s^{(c)} = (1, \lambda_2/\lambda_1, ..., \lambda_M/\lambda_1)$ with c=1,...,C has to be defined. Candidate vector $s^{(1)}$ is usually defined as $s^{(1)} = (1, ..., 1)$ and stands for the original Lasso. Therefore, IPF-Lasso includes the original Lasso and returns original Lasso results whenever an equal penalization for all modalities makes sense.

Cross validation (CV) is applied for each candidate vector to determine the $\lambda_1$ which gives the best prediction performance. Finally the vector $s^{(c_{opt})}$ is selected which gives the best fit according to a selected performance metric. With standard computer hardware, these calculation steps limit the practical use of IPF-Lasso in terms of computation time. Eliminating this bottle-neck by a modification of the method could help to extend its practical applicability to more than two or three modalities.

## 2.5 Two-step IPF-Lasso

The idea of the two-step IPF-Lasso is to determine first a single candidate vector of penalty factors based on the data and thus avoid to manually defining candidate vectors and most of the above mentioned calculations. This vector is then used in a second step to calculate the penalty factor $\lambda_1$ which gives the optimal prediction performance of the model. A well-known two-step Lasso model is the adaptive Lasso of Zou [2006]. Coefficient estimates are determined by ordinary least squares in a first step. These pilot estimates are then used as weights in the penalty term of the second-step Lasso. However, the approach only works for $p < n$ and thus is not applicable for multi-omics data. Huang et al. [2008] extended the model to $p \gg n$ data by using univariate regression in the first step.

In contrast to the adaptive Lasso, the two-step IPF-Lasso model uses the means of the absolute values of the first-step coefficient estimates of each modality as weights in the penalty term of the second step. A similar procedure has been applied by Ternès et al. [2017] in the context of estimating interactions in randomized clinical trials and the idea has been mentioned by Tai and Pan [2007] in an earlier paper. Several versions for the first step of the two-step IPF-Lasso can be set and have been evaluated in this study (see Table 1). The parameter estimates might be calculated by a Ridge or a Lasso model either on the whole data set or separately for each modality. The logic behind trying separate models in Step 1 is due to the fact that variables from separate modalities can be correlated. A combined Lasso model usually picks one of these correlated variables and ignores the others. Running separate models in Step 1 could therefore save some relevant variables for Step 2. The output of Step 1 is a set of M means of the respective parameter estimates $\hat{\beta}_j^{(m)}$ of each modality calculated as arithmetic means by

$$\bar{\hat{\beta}}^{(m)}_{arith} = \frac{1}{p_m} \sum_{j=1}^{p_m} |\hat{\beta}_j^{(m)}| \tag{7}$$

and as geometric means by

$$\bar{\hat{\beta}}^{(m)}_{geom} = \sqrt[p_m]{\prod_{j=1}^{p_m} |\hat{\beta}_j^{(m)}|} \tag{8}$$

with m=1,...,M and $p_m$ the length of modality m.

| Version | Modalities | Model | Mean |
|---|---|---|---|
| TSIPF1 | combined | Ridge | arithmetic |
| TSIPF2 | combined | Ridge | geometric |
| TSIPF3 | combined | Lasso | arithmetic |
| TSIPF4 | separate | Ridge | arithmetic |
| TSIPF5 | separate | Ridge | geometric |
| TSIPF6 | separate | Lasso | arithmetic |

Table 1: Six versions of the two-step IPF-Lasso, which are defined by the model type of Step 1, applied to combined or separate modalities and the arithmetic or geometric mean of the resulting coefficient estimates for each modality.

Some or even all of the coefficient means can become zero if a Lasso model is used in Step 1. In these cases, the respective modalities are either excluded from the calculations of Step 2 or the modeling process stops after Step 1. The reciprocal values of the non-zero coefficient means of Step 1 are combined into a single candidate vector of penalty factors $pf = (1/\bar{\hat{\beta}}^{(1)}, 1/\bar{\hat{\beta}}^{(2)}, ..., 1/\bar{\hat{\beta}}^{(M)})$. Alternatively, one could use $pf^* = (1, \bar{\hat{\beta}}^{(1)}/\bar{\hat{\beta}}^{(2)}, ..., \bar{\hat{\beta}}^{(1)}/\bar{\hat{\beta}}^{(M)})$, inspired by the original IPF-Lasso paper. But this is an equivalent option, because multiplying the penalty vector by a scalar has no influence on the final modeling result [Boulesteix et al., 2017a]. Taking the reciprocal means of the coefficient estimates as candidate penalty factors assigns the lowest penalty factor to the modality with the highest mean of coefficient estimates. Likewise, the modality with the lowest coefficient estimate mean gets the highest penalty factor assigned. Thus, the results of Step 1 will be supported in the second step in a sense that covariates which survived the first step and seem to have an impact on the response are less penalized in the second step than covariates which almost failed. In contrast, one could also enter the coefficient means directly into the candidate vector and achieve a relatively high penalty for the modality with the highest mean of coefficient estimates and a relatively low penalty for the modality with the lowest mean of coefficient estimates from Step 1. However, these statements become less valid, the more the means of the coefficient estimates are similar so that the penalty factors in Step 2 are also similar and the modalities are penalized equally. It is also important to realize that the size of the Step 1 means of coefficient estimates depends not only on the

estimates, but also on the size of the modalities. Larger modalities tend to achieve smaller means compared to small modalities if the sums of the absolute coefficient estimates per modality are of similar size.

This study follows the reciprocal approach and thus modalities with high absolute coefficient estimates or low modality size get a "penalization bonus" in Step 2. The actual parameter estimates of the two-step IPF-Lasso model are then determined in the second step by

$$\|y - \sum_{m=1}^{M} X^{(m)}\beta^{(m)}\|_2^2 \; + \; \lambda \sum_{m=1}^{M} \frac{1}{\bar{\hat{\beta}}^{(m)}} \|\beta^{(m)}\|_1 \quad \stackrel{\beta}{\rightarrow} \quad min \tag{9}$$

where $\bar{\hat{\beta}}^{(m)}$ is the arithmetic or geometric mean of the Step 1 coefficient estimates of modality m.

# 3 Data

The new two-step IPF-Lasso model and its competitors have been evaluated with various simulated multi-omics data scenarios and with four real data sets from TCGA.

## 3.1 Simulated data

The approach for creating simulated multi-omics data has been adopted from Boulesteix et al. [2017a]. The data consists of a binary response variable (probability for success $\pi = 0.5$) and two modalities of varying sizes $p_1$ and $p_2$. The modalities contain a certain number of truly relevant variables $p_1^r < p_1$ and $p_2^r < p_2$. The entire set of covariates is drawn from a multivariate normal distribution with mean $\mu = (\beta_1^{(1)}, ..., \beta_1^{(p_1^r)}, 0, ..., 0, \beta_2^{(p_1+1)}, ..., \beta_2^{(p_1+p_2^r)}, 0, ..., 0)$ where $\beta_1$ and $\beta_2$ are the effects of the $p_1^r$ and $p_2^r$ relevant covariates of the first and second modality. The covariance matrix $\Sigma = I_{(p_1+p_2)\times(p_1+p_2)}$ is the identity matrix which means the data is uncorrelated with unit variance.

Six scenarios with uncorrelated data are shown in Table 2. The modality sizes vary from being equal (scenario A) to extremely different (scenarios E and F). Scenarios E and F simulate a case where low-dimensional clinical data is combined with a molecular data set, like gene expression data.

| Scenario | $p_1$ | $p_2$ | $p_1^r$ | $p_2^r$ | $\beta_1$ | $\beta_2$ |
|---|---|---|---|---|---|---|
| A | 1000 | 1000 | 10 | 10 | 0.5 | 0.5 |
| B | 100 | 1000 | 3 | 30 | 0.5 | 0.5 |
| C | 100 | 1000 | 10 | 10 | 0.5 | 0.5 |
| D | 100 | 1000 | 20 | 0 | 0.3 | - |
| E | 20 | 1000 | 3 | 10 | 1 | 0.3 |
| F | 20 | 1000 | 15 | 3 | 0.5 | 0.5 |

Table 2: Simulation scenarios A to F for uncorrelated data, $p_1$ and $p_2$ are the modality sizes, $p_1^r$ and $p_2^r$ the numbers of relevant covariates in each modality and $\beta_1$ and $\beta_2$ their impact.

Scenario A is less realistic, but could be thought of as a combination of two molecular data sets. The number of relevant variables in the two modalities and their impacts are varied in a similar way from equal to very different.
In reality, multi-omics data sets are correlated within and between modalities. Therefore, an additional set of six scenarios A' to F' has been defined by using the same parameters of scenario A to F and only changing the covariance matrix accordingly. The assumption is that b=10 groups of correlated variables exist in each modality and group j of the first modality is correlated with group j of the second modality where the correlation is set to $\rho = 0.4$. The resulting covariance matrix 10 consists of matrices $A_{p_1/b}(\rho)$ and $A_{p_2/b}(\rho)$ which have ones on the diagonal and $\rho$ off-diagonal. The matrices $B_{p_1/b,p_2/b}(\rho)$ and $B_{p_2/b,p_1/b}(\rho)$ represent the correlation between the modalities and all their elements are equal to $\rho$. All other elements of the covariance matrix $\Sigma$ are zero.

$$\Sigma = \left(\begin{array}{ccc|ccc} A_{p_1/b}(\rho) & & & B_{p_1/b,p_2/b}(\rho) & & \\ & A_{p_1/b}(\rho) & & & & \\ & & \ddots & & \ddots & \\ & & A_{p_1/b}(\rho) & & & B_{p_1/b,p_2/b}(\rho) \\ \hline B_{p_2/b,p_1/b}(\rho) & & & A_{p_2/b}(\rho) & & \\ & \ddots & & & \ddots & \\ & & B_{p_2/b,p_1/b}(\rho) & & & A_{p_2/b}(\rho) \end{array}\right) \quad (10)$$

For each scenario 100 data sets of size n=100 have been created. On top of the two times six scenarios there are 33 more scenarios with further modifications of the simulation parameters (Appendix B, Table 10).

## 3.2 Real data

The Cancer Genome Atlas (TCGA) was founded in 2005 by the US-American National Institute of Health (NIH) and aims at improving our ability to diagnose, treat and prevent cancer. Currently, genome profiles of 33 cancer types are online available. Among these are gene expression, miRNA expression, copy number variation (CNV) or DNA methylation data [Tomczak et al., 2015].

TCGA data sets on acute myeloid leukemia (AML), breast invasive carcinoma (BRCA), esophageal carcinoma (ESCA) and kidney renal papillary cell carcinoma (KIRP) are used to evaluate the prediction performance of the new two-step IPF-Lasso. Each data set contains gene expression, CNV and clinical data. The AML data set has methylation and miRNA data as additional modalities with the latter also being part of the ESCA and KIRP data sets.

The original data contains a reasonable number of duplicate covariates positioned next or close to each other i.e. different covariates have completely identical observations (Table 3). This seems to be caused by the sequencing procedures and as a consequence of the identical columns in the data matrix, these covariates are 100% correlated. Penalized models can deal with such data by selecting one of the correlated covariates and ignoring the other. Nevertheless, in each case only one of the duplicates has been left in the final data sets in order to save computation time.

| Data | Unique | Duplicate | %Duplicate |
|---|---|---|---|
| AML | 414162 | 17222 | 4 |
| BRCA | 42639 | 4326 | 9 |
| ESCA | 70692 | 45266 | 39 |
| KIRP | 61820 | 18360 | 23 |

Table 3: Absolute number of unique and duplicate covariates and percentage of duplicates in TCGA datasets.

Each molecular modality has been checked for missing values and combined according to patient labels. Only clinical covariates with no or at least a minimum number of missing values have been selected to avoid a further reduction of observations and merged with the molecular data. The covariates of the clinical

modalities are summarized in Table 11 of Appendix B. Some of them are factors and need to be converted into numerical dummy variables as the used software

| Data | mRNA | miRNA | CNV | Methylation | Clinical | n | p | Response |
|---|---|---|---|---|---|---|---|---|
| AML | 19204 | 550 | 1140 | 393264 | 4 | 173 | 414162 | surv |
| BRCA | 23287 | - | 19339 | - | 13 | 1006 (501) | 42639 | surv |
| ESCA | 55410 | 1441 | 13834 | - | 7 | 129 | 70692 | surv |
| KIRP | 53441 | 1380 | 6993 | - | 6 | 121 | 61820 | bin |

Table 4: Structure of TCGA datasets: Number of covariates per modality, overall number of covariates p and sample size n for each data set and type of response variable (surv=survival, bin=binary).

packages only accept a numerical data matrix as input. The response variables are right-censored survival times for the AML, BRCA and ESCA data and tumor type, a binomial variable for the KIRP data. Table 4 shows the structure of the final data sets in terms of modalities, their respective number of covariates, sample sizes, overall number of covariates and the response types. Because of software issues (see Section4.4), only 501 observations of the BRCA data could be analyzed.

# 4 Analysis

This section gives an overview about the performed data analysis, how the results have been evaluated and reports some issues related to software.

## 4.1 Simulation

The simulation data has been analyzed with the adapted R-scripts of Boulesteix et al. [2017a] using version 3.4.1 of the R-software [R Core Team, 2017]. The scripts work with R-packages ipflasso [Boulesteix and Fuchs, 2015], glmnet [Friedman et al., 2010, Simon et al., 2011], SGL [Simon et al., 2013a], psych [Revelle, 2017] and mvtnorm [Genz et al., 2017]. The six optional versions of the two-step IPF-Lasso model have been included into the original script so that in summary ten different models have been run for each of the $niter = 100$ data sets in each scenario. Apart from SGL, each model uses a 5-fold cross validation (CV) with $ncv = 10$ repeats based on misclassification rates for tuning the $\lambda$ penalty. The SGL package doesn't allow repeated CV and therefore the SGL models have been estimated with a single CV.
The prediction performance of all models has been evaluated by calculating the misclassification rates and the areas under the curve (AUC) based on an additional test data set of size $n = 5000$. The complete R-scripts for the simulation analysis can be found in Appendix C.

## 4.2 Real data analysis

Version 3.4.1 of the R-software [R Core Team, 2017] has been also used for analyzing the real data sets. Applying the R-script of the two-step IPF-Lasso model to the TCGA data requires the R-packages caret [from Jed Wing et al., 2017], survival [Therneau, 2015] and pec [Mogensen et al., 2012] on top of the already mentioned R-packages ipflasso, glmnet, SGL and psych. Similar to the ipflasso package, it allows right-censored survival times (Cox-PH model), binomial (logistic model) and continuous variables as responses. The AML, the BRCA and the ESCA data have right-censored survival times as responses and the KIRP data has a binomial response. The data has to be formatted as single numerical matrix combining all

modalities with the first column containing a binomial or continuous response or survival times and the event indicator in the first two columns. The parameters to be entered before running the script are the length of each modality $(p_1, ..., p_M)$, the model type (in this case "cox" or "binomial"), the type of loss measure for CV ("deviance" for the Cox models and "class" for the logistic models), the size of the CV folds (nfolds) and the number of CV repetitions (ncv). The CV default values have been set to $nfolds = 5$ and $ncv = 10$ for all models (apart from SGL, which only allows single CV), whereas the modality lengths had to be changed according to the respective data sets.

Again, the purpose of the 5-fold cross validation is the tuning of the $\lambda$ penalty. However, in contrast to the simulation data analysis, there was the need to follow the approach of randomly splitting the given data sets into a training and a test set in order to properly estimate the prediction error [Boulesteix et al., 2008]. These splits have been repeated 100 times per data set and average performance measures reported as final results.

Right after each data split, the training and test sets are checked for covariates with near zero variance and these are removed from both data sets. Training and test data are then standardized and fed into nine models (standard Lasso, separate models, SGL and the six two-step IPF-Lasso versions). The original IPF-Lasso model was not included into the analysis because of the computation time issue. During the analysis, it also turned out that the SGL package doesn't deliver stable results and therefore the SGL model had to be excluded from studying the real data applications. The complete R-script for the real data analysis can be found in Appendix C.

## 4.3 Evaluation of results

The model performance has been evaluated into several directions.

The computation times for analyzing the simulated and the real data sets have been determined and compared to each other. However, this comparison could not be done for all models. For example the IPF model could only be tested with simulated data because it cannot handle large data sets in a reasonable time.

Another aspect is the capability of a model to discover relevant covariates and

enter them into the final model. Of course, the proportion of discovered relevant covariates can only be determined in situations where these are known in advance. This could be the case in some real data applications, but is definitely possible for simulated data and therefore the proportions of discovered relevant covariates has been determined for the simulation scenarios.

The chance to capture relevant covariates increases with the size of a model, but from a practical point of view one would like to have sparse models. Model sparsity has been visualized by the distribution of the number of covariates with non-zero coefficients for each model.

Finally a prediction error for each model has been estimated. There are many options to measure the prediction error in which the type of regression model plays a certain role. In this study the misclassification rate and the AUC (area under the curve) are used for the logistic regression models and prediction error curves based on the Brier score are determined for the Cox regression models.

The misclassification rate is estimated for the simulation scenarios based on test data which has been generated independently of the training data. It is defined by dividing the sum of misclassifications by the number of all model predictions [Boulesteix et al., 2008], ranges from 0 to 1 and smaller values are better.

The AUC as described for example in Gerds et al. [2008] is derived from the receiver operating characteristic (ROC). The ROC curve is a plot of the true positive classification rates of a binary model against the false positive rates at different classification thresholds and the AUC is the area under the ROC curve. The AUC ranges from 0.5 to 1 with 0.5 as useless and 1 as perfect prediction capability. Like for the misclassification rate, the AUC for the simulation scenarios is estimated on randomly generated test data and for the real data models, it is based on the repeated random splits into training and test data.

The Brier score [Mogensen et al., 2012] can be used to measure the prediction error of Cox survival models. If one interprets the estimated probabilities for an event at a certain time $t^*$ as predictions of the event status, the Brier score is the mean square error of the predictions [Graf et al., 1999]. Brier scores can be calculated for time points over the entire time axis and visualized as prediction error curves. For comparing the prediction performance of several models via prediction error

curves, usually the prediction errors of a null model are added to the graph as a reference. The null model corresponds to the Kaplan-Meier survival estimates. The prediction error curves can be further summarized by the integrated Brier score (IBS), which is defined as

$$IBS(BS, \tau) = \frac{1}{\tau} \int_0^\tau BS(u, \hat{S}) du \tag{11}$$

where BS stands for Brier score, $\hat{S}$ is the predicted survival probability calculated on test data and $\tau$ is any point on the time axis smaller than or equal to the minimum of the lengths of all the available time axis in the 100 repetitions of the random data splits. In order to get a single graph representing the overall prediction performance of the 100 model estimations, the time axis of each split has been divided into equally spaced time points, the IBS has been calculated for each of these time points in each split and averaged over all splits. The resulting IBS averages for all models have been plotted over time in order to compare the prediction performance of the models.

However, it is important to realize that statements about one method being better than the other because one has a lower prediction error than the other are based on the limited number of data sets at hand and thus can hardly be generalized. Comparing figures in graphs and tables without formal statistical test procedure also has a subjective element in it and therefore any conclusions should be drawn with caution.

## 4.4 Software issues

During the analysis process several issues related to software occurred. It turned out that the SGL package cannot handle very large data sets in a stable manner. The AML data matrix has 173 rows and 414162 columns. In many attempts, the SGL package managed only two times to actually return model estimates for this data set. The computation time for getting these results was extremely long with 17 hours at the computer of the Leibniz-Rechenzentrum (LRZ) and 12 hours on the laptop. The measured time for the LRZ computer might be biased, because it can vary depending on the number of users and the size of their jobs running at the time of the evaluation. However, these two instances have

been anyhow rare exceptions. Usually, R-Studio crashed issuing the message *"R session aborted/R encountered a fatal error/The session was terminated/Start new session"*. Changing the computer, the operating system, the R-version or running SGL without R-Studio via the R graphical user interface still returned the same message after a couple of minutes run time. A similar behavior was observed with the other three real data sets, although these are smaller than the AML data set. There have been no problems running the SGL package with the simulation data. Therefore it was decided to analyze the real data without the SGL model and only include it into the simulation analysis.

Another issue was related to convergence in the glmnet function. The maxit parameter defines the maximum number of passes over the data for all lambda values and per default is set to $10^5$. For some data sets (simulated and real) the default value was too low and glmnet issued a message like *"... Convergence for* $1^{st}$ *lambda value not reached after maxit=100000 iterations; solutions for larger lambda returned"*. The problem could be solved for most of the cases by setting the maxit parameter to $10^6$, without a noticeable increase of computation time. However, the maxit parameter is not accepted by the cvr.ipflasso function which is used in the second step of the TSIPF-Lasso models. Therefore, depending on the data, the IPF-Lasso and the TSIPF-Lasso models sometimes deliver too large lambda values.

The glmnet function also issued an error message during the analysis of the BRCA data: *"In getcoef(fit,nvars,nx,vnames): an empty model has been returned; probably a convergence issue"*. These problems did not completely disappear after randomly reducing the sample size from n=1006 to n=501 as 13 of the 100 runs still returned a NULL result. In the original data set, the binary response variable for the events was quite imbalanced (904 times "0" and 102 times "1") and therefore it was decided to modify the random sampling by drawing only from the "0" rows. Thus the response of the resulting data set contains 102 times the event "1" and 399 times the event "0". Running the analysis with this reduced data set hasn't generated any more error messages.

Finally there was a problem in the second step of the two-step IPF-Lasso script. The program uses the cvr.ipflasso function from the ipflasso package to estimate

the final model coefficients. The cvr.ipflasso function calls the cvr.glmnet function which uses the cv.glmnet function of the glmnet package to generate a first set of lambda values. These lambda values are then fed into a for-loop to go through the ncv cross validation steps. Sometimes, instead of generating a set of first lambda values, there is only a single lambda value returned. However, as the cv.glmnet function doesn't accept single lambda values as input, the program generates an error message and stops. This problem can be avoided by changing the default value of the minimum number of lambdas generated by the cv.glmnet function. The default value for this parameter is mnlam = 5 and setting it to mnlam = 10 in the glmnet.control function eliminates the problem.

# 5 Results

This section gives an overview about the main analysis results summarized by the various measures described in Section 4. Further results can be found in the Appendix.

## 5.1 Computation times

A major reason to look for alternatives of the IPF-Lasso model is the computation time which increases exponentially with the number of modalities. In general, computation time for a statistical model depends on various factors like the size of the data set to be analyzed, the model type, the software algorithm, the available hardware and the option of parallel vs. linear computing for certain tasks. Computation times have been measured on a standard laptop with a 2.7 GHz CPU and 8 GB RAM and a workstation with a CPU running at 4.2 GHz and 128 GB RAM.

| Model | Computation time (average in seconds) | Computation time (% of IPF-Lasso) |
|---|---|---|
| Standard Lasso | 2.6 | 12.0 |
| Separate models | 3.7 | 17.0 |
| SGL | 30.2 | 138.8 |
| TSIPF1 | 3.2 | 14.7 |
| TSIPF2 | 3.2 | 14.7 |
| TSIPF3 | 2.5 | 11.5 |
| TSIPF4 | 3.1 | 14.3 |
| TSIPF5 | 3.1 | 14.3 |
| TSIPF6 | 2.6 | 11.8 |
| IPF-Lasso | 21.8 | 100.0 |

Table 5: Average computation times in seconds and percent computation times of IPF-Lasso over 100 runs for all studied models under simulation scenarios A-F (uncorrelated data), measured at an 8 GB RAM laptop with a 2.7 GHz CPU.

Table 5 shows the average computation times of the models for simulation scenarios A-F (uncorrelated data), calculated on the laptop. For each investigated model and each scenario, the absolute computation time has been measured and the average calculated over the 100 repetitions of the analysis. Finally the means of all scenarios A-F have been averaged to get the values in the table. Relative times in % of the IPF-Lasso computation time are shown as well.
All the two-step IPF-Lasso models need less than 15% of the IPF-lasso run time. The Lasso-Lasso versions run faster than the Ridge-Lasso versions and have practically the same computation times as the standard Lasso. The SGL model has the longest computation time and on average runs almost 40% longer than the IPF-Lasso. The average computation times for the scenarios A'-F' (correlated data) can be found in Table12 of Appendix B. There is practically no difference between the results of the correlated and uncorrelated scenarios.

| Model | Computation time (hours) |
|---|---|
| Standard Lasso | 1.27 |
| Separate models | 1.25 |
| SGL | 12.03 |
| TSIPF1 | 1.41 |
| TSIPF2 | 1.41 |
| TSIPF3 | 1.21 |
| TSIPF4 | 1.29 |
| TSIPF5 | 1.29 |
| TSIPF6 | 1.27 |
| IPF-Lasso | - |

Table 6: Computation times for AML data: Single runs on a 4.2 GHz CPU/128 GB RAM workstation with 5-fold CV and 10 repetitions for each model.

Comparisons of computation times for real data could only be made for the six versions of the two-step IPF-Lasso model, the standard Lasso and the separate models approach. The result for the SGL model could not be verified by a second run and therefore needs to be interpreted with care. The IPF-Lasso model had to

be completely excluded because of the exponentially increasing run time as soon as there are more than two modalities to be analyzed. Table 6 shows the actual computation times for a single run of the AML data. The run includes 5-fold CV with 10 repetitions for each model and has been done on the workstation.
The relative computation times for the two-step IPF-models are similar to the simulation scenarios, with the Ridge-Lasso models having a slightly higher result than the Lasso-Lasso models. Again, the times are not far away from the standard Lasso model and the separate models approach and the SGL model has by far the longest time. Table 13 in Appendix B shows computation times for a single run of the ESCA data on the workstation. The ESCA data set is much smaller compared to the AML data and therefore the run times are relatively short. The times vary a bit more than the AML results, but the Ridge-Lasso models again need more time than the Lasso-Lasso models and overall, the timing is comparable to the standard Lasso and the separate models approach.

## 5.2 Prediction performance on simulated data

For each simulation scenario A-F, A'-F' and each model, the distributions of the misclassification rate, the AUC and the number of non-zero coefficients (model sparsity) are displayed by box plots (Figures 1 and 2). The model performance varies between and within the different scenarios.
Scenario A is a case which is rather unrealistic in practice as the two modalities have the same size and the same number of relevant covariates with identical impact. Here the performance of models TSIPF1 and TSIPF2 is better than the performance of the IPF-Lasso model in terms of misclassification rate and AUC and similar to the result of the standard Lasso. The TSIPF3 model which is the Lasso-Lasso type with a combined first step analysis and arithmetic mean performs similar to the separated models approach. The performance of the models TSIPF3 to TSIPF6 is slightly below the IPF-Lasso model with the TSIPF6 model having the worst result.
In terms of model sparsity, all two-step IPF-Lasso models contain approximately a similar number of non-zero coefficients compared to the IPF-Lasso and also show some increased variation.

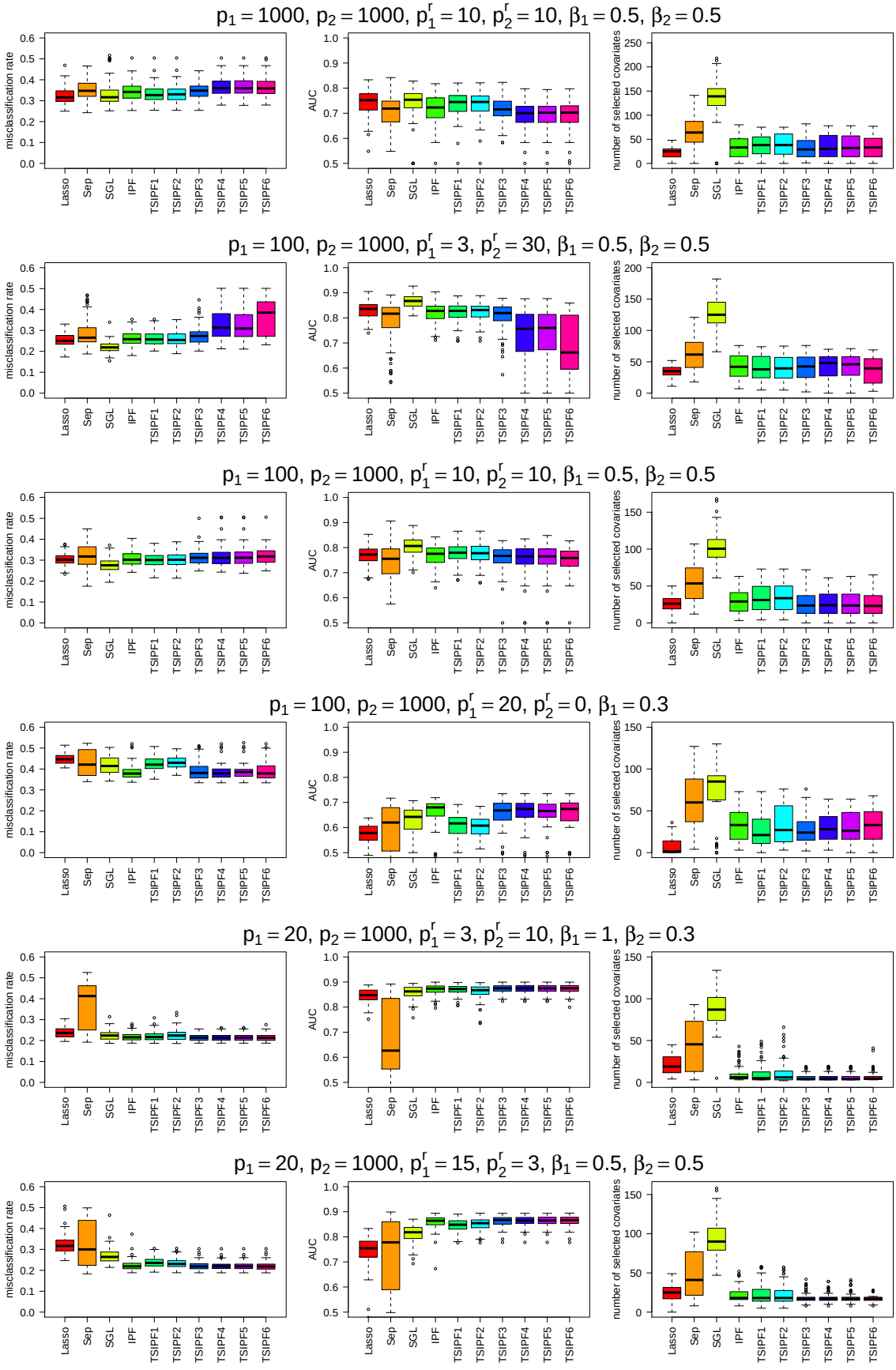


Figure 1: Misclassification rate (left), AUC (center) and number of selected covariates (right) for simulation scenarios A-F (uncorrelated data).

The standard Lasso model contains the lowest number and the SGL model by far the highest number of non-zero coefficients. The high number of non-zero coefficients in the SGL model might contribute to its better prediction performance relative to the other models. Also for correlated data the models TSIPF1 and TSIPF2 have both a better prediction performance than the IPF-Lasso model. The misclassification rate and AUC for the three model types with a separated Step 1 analysis (TSIPF3 to TSIPF6) are practically the same and not as good as the results of the other competitors. The standard Lasso and the SGL model show the best misclassification rate and AUC, but in terms of model sparsity they have the lowest and highest values.

Scenario C has the same number and same impact of relevant covariates like scenario A, but the two modality sizes are different. The results of the IPF-Lasso model and the six two-step IPF-Lasso versions are approximately similar and also in line with the standard Lasso model. For correlated data, the TSIPF1 and TSIPF2 models perform slightly better than the IPF-Lasso model, which might be due to the higher number of non-zero coefficients in the two models.

Scenario B further generalizes the data structure by introducing different numbers of relevant covariates into the two modalities. In this scenario the models TSIPF1 and TSIPF2 have the same results as the IPF-Lasso model, whereas the models TSIPF3 to TSIPF6 perform worse with model TSIPF6 having the highest misclassification rate and the lowest AUC and model sparsity. These results are quite similar for the uncorrelated and the correlated data.

Scenarios E and F have extremely different modality sizes which simulates the case of combining clinical data with genetic covariates. The numbers of relevant covariates are different as well with the same impact in each modality in scenario F and a different impact in scenario E. The performance of the IPF-Lasso model and the six two-step IPF-Lasso versions are the same for scenario E and TSIPF1 and TSIPF2 two are slightly below in scenario F for both the uncorrelated and the correlated data. In scenario F the results of the SGL model are even below the IPF-Lasso model and models TSIPF1 to TSIPF6. However, such a comparison is not really fair, because the SGL model has been run with the tuning parameter $\alpha$ at its default value.

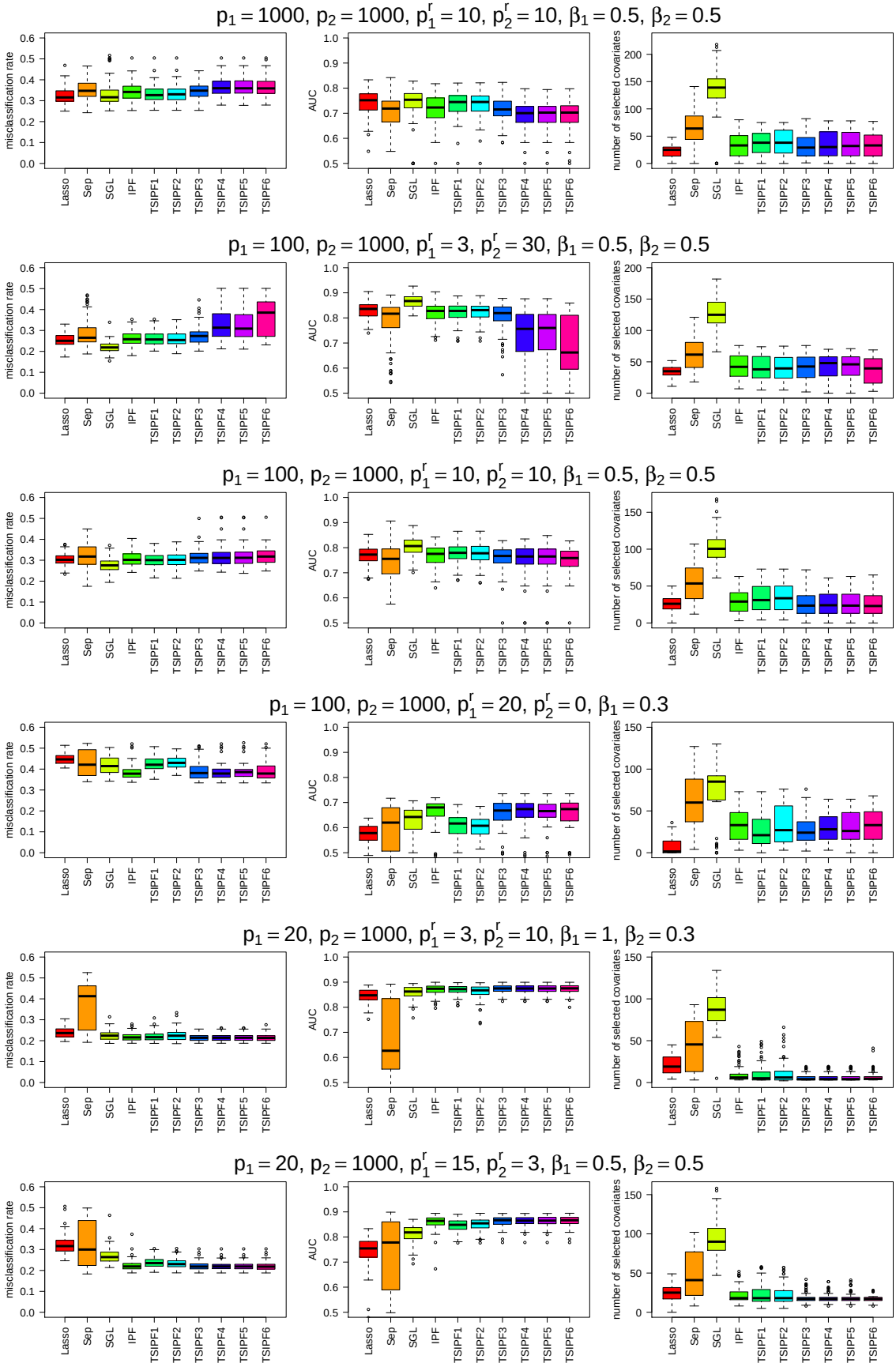


Figure 2: Misclassification rate (left), AUC (center) and number of selected covariates (right) for simulation scenarios A'-F' (correlated data).

Modifying this parameter might have improved the results for the SGL model under scenario F and perhaps also in general. In both scenarios, the results of the separate models approach show the highest variation and in scenario E the worst result.

Finally, scenario D has different modality sizes and no relevant covariates in the bigger modality. This scenario simulates a real data scenario where a number of clinical covariates and almost no omics-covariates are found to be relevant. In the correlated and uncorrelated version of this scenario, the models TSIPF3 to TSIPF6 show the same AUC and misclassification rate as the IPF-Lasso model and these five models perform better than the other competitors.

On top of the basic simulation scenarios A-F and A'-F', 33 additional scenarios have been run (Table 10) each for correlated and uncorrelated data. For each scenario the median of the misclassification rate, the AUC and the number of non-zero model coefficients of the 100 repetitions has been determined. The boxplots in Figures 9 and 10 display these medians for uncorrelated and correlated data, including the respective medians of the basic scenarios.

The IPF-Lasso model and the TSIPF3 model have similar median misclassification rates and AUCs in the uncorrelated data case. The misclassification rates for TSIPF1, TSIPF2 and TSIPF4 to TSIPF6 are slightly above the IPF-Lasso model with the TSIPF6 model at the highest position. These results are reflected on the AUC boxplots as well. In terms of the number of non-zero coefficients, the TSIPF3 model has the lowest position, TSIPF1 and TSIPF2 are similar to the IPF-Lasso model and the TSIPF4 to TSIPF6 models are less sparse than the other two-step versions of the IPF-Lasso model. The results for the correlated data are similar. Only the model sparsities for the IPF-Lasso model and the TSIPF-Lasso models have a wider range and the SGL model contains less non-zero coefficients compared to the uncorrelated data scenarios.

With simulated data the number of relevant covariates is known and therefore one can determine how many of these variables end up in the final model equation. The proportions of relevant covariates discovered by the models for scenarios A to F (uncorrelated data) are shown in Table 7 and for scenarios A' to F' (correlated data) in Table 14 of Appendix B.

| | Scenario | IPF | TSIPF1 | TSIPF2 | TSIPF3 | TSIPF4 | TSIPF5 | TSIPF6 |
|---|---|---|---|---|---|---|---|---|
| Modality 1 | A | 0.34 | 0.42 | 0.41 | 0.33 | 0.33 | 0.33 | 0.30 |
| | B | 0.26 | 0.43 | 0.44 | 0.37 | 0.52 | 0.53 | 0.60 |
| | C | 0.68 | 0.57 | 0.55 | 0.72 | 0.72 | 0.72 | 0.74 |
| | D | 0.48 | 0.27 | 0.26 | 0.42 | 0.46 | 0.46 | 0.44 |
| | E | 0.99 | 1.00 | 0.98 | 1.00 | 1.00 | 1.00 | 1.00 |
| | F | 0.88 | 0.80 | 0.83 | 0.88 | 0.89 | 0.89 | 0.88 |
| Modality 2 | A | 0.25 | 0.35 | 0.35 | 0.29 | 0.26 | 0.26 | 0.24 |
| | B | 0.44 | 0.44 | 0.44 | 0.39 | 0.31 | 0.31 | 0.16 |
| | C | 0.21 | 0.38 | 0.40 | 0.09 | 0.12 | 0.13 | 0.03 |
| | D | - | - | - | - | - | - | - |
| | E | 0.03 | 0.04 | 0.06 | 0.0 | 0.0 | 0.0 | 0.01 |
| | F | 0.10 | 0.23 | 0.21 | 0.01 | 0.02 | 0.01 | 0.0 |

Table 7: Proportions of discovered relevant covariates for simulation scenarios A-F (uncorrelated data).

Scenario A simulates equal numbers of relevant covariates for the two modalities. In this case all models apart from the TSIPF1 and TSIPF2 model contain about the same proportion of relevant covariates within the two modalities with lower proportions for Modality 2. In both modalities the TSIPF1 and TSIPF 2 models have higher proportions compared to the other models. Scenario C has the same number of relevant covariates in the two modalities, but Modality 1 has a smaller size than Modality 2. For Modality 1 the models TSIPF3 to TSIPF6 have higher proportions and the models TSIPF1 and TSIPF2 have lower proportions compared to the IPF-Lasso. Modality 2 has here the bigger size and the proportions for each model are just the opposites of Modality 1: The models TSIPF1 and TSIPF2 have higher proportions and the models TSIPF3 to TSIPF6 have lower proportions compared to the IPF-Lasso. Scenario B has 3 relevant covariates in Modality 1 and 30 in Modality 2. All TSIPF-Lasso models have higher proportions than the IPF-Lasso model in Modality 1 and in Modality 2 the IPF-Lasso has the same result than the TSIPF1 and TSIPF2 models. The models TSIPF3 to TSIPF6 have lower proportions. Scenario E has 3 relevant covariates in the small

modality and 10 in the big modality. Almost all models achieve a proportion of 1 in Modality 1 and close to 0 in Modality 2. Scenario F has a high number of relevant covariates in the small Modality 1 and a low number in the big Modality 2. Again all the models achieve high proportions in Modality 1 and low ones in Modality 2 although the models TSIPF1 and TSIPF2 perform better than the others. Scenario D has no relevant covariates in Modality 2 and 20 in the smaller Modality 1. In this scenario the IPF-Lasso model achieves the same results as the models TSIPF3 to TSIPF6 whereas the models TSIPF1 and TSIPF2 have lower proportions. The outcome for the correlated simulation data is similar to the outcome of the uncorrelated data.

## 5.3 Prediction performance on real data

The prediction results of modeling the AML, the ESCA and the KIRP data are presented in this section and the results for the BRCA data can be found in the Appendix (Figure 11 and Table 15). The responses for the AML, the BRCA and the ESCA data are censored survival times and therefore Cox PH models have been applied. The KIRP data has a binary response and thus the models have been estimated via logistic regression.

As already explained in Section 4.3, the integrated Brier scores (IBS) have been calculated for each survival model at equally spaced time points and averaged over the 100 runs. Figure 3 shows the results plotted against time for the AML data. The Kaplan-Meier (K-M) curve represents the IBS averages based on the estimation of the Kaplan-Meier survival curve, which is calculated from the survival times only, without considering any covariates. Therefore it can be used as a reference against which the other models are compared.

For times above approximately month 18, the curves for all models are below the Kaplan-Meier reference, indicating a better prediction performance and thus the usefulness of the models. In this range the separate models approach achieves the best results and the standard Lasso model the worst, but still better than the reference. The curves of the six two-step IPF-Lasso model versions lie between the curves of the standard Lasso and the separate models approach.

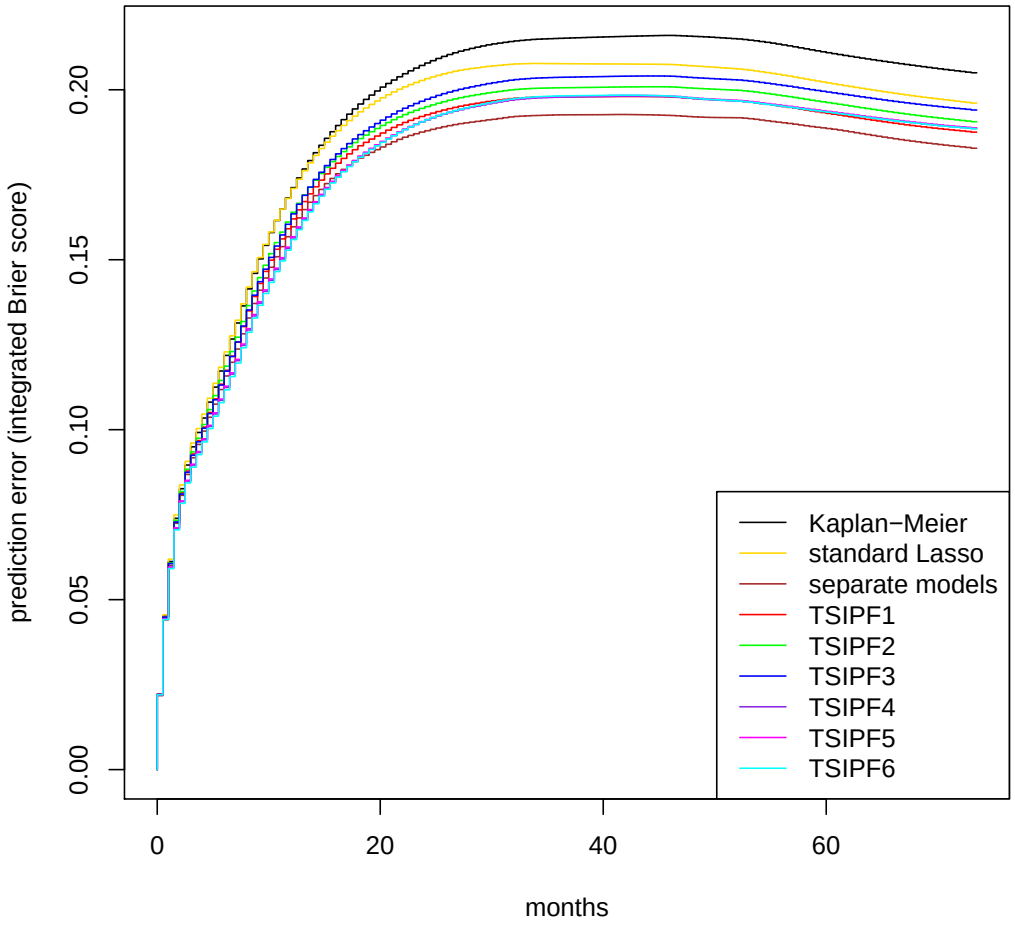


Figure 3: AML data: Integrated Brier score (averages over 100 runs at equally spaced time points) for standard Lasso, separate models and the six versions of the TSIPF-Lasso model. The Kaplan-Meier curve serves as reference.

| Model | K-M | Lasso | Sep | TSIPF1 | TSIPF2 | TSIPF3 | TSIPF4 | TSIPF5 | TSIPF6 |
|---|---|---|---|---|---|---|---|---|---|
| IBS20 | 0.198 | 0.196 | 0.182 | 0.185 | 0.188 | 0.189 | 0.183 | 0.183 | 0.182 |
| IBS40 | 0.215 | 0.208 | 0.193 | 0.198 | 0.201 | 0.204 | 0.198 | 0.198 | 0.198 |
| IBS60 | 0.211 | 0.203 | 0.189 | 0.194 | 0.197 | 0.200 | 0.194 | 0.194 | 0.194 |

Table 8: AML data: Average integrated Brier score at 20, 40 and 60 months

The TSIPF4, TSIPF5 and TSIPF6 models have identical curves indicating that for separated first step models it seems to make no difference whether this step

is done with Ridge or Lasso regression and whether the weights for the second step are estimated by the arithmetic or the geometric means. Before month 30, the TSIPF1 model curve is above the curves for the models TSIPF4, TSIPF5 and TSIPF6, becoming equal after month 30. Using the geometric mean (model TSIPF2) instead of the arithmetic mean (model TSIPF1) results in a higher average IBS across the entire time axis. The model with the Lasso Step 1 (TSIPF3) achieves the highest IBS averages of all two-step IPF-Lasso models, which means it has the lowest prediction performance. Table 8 shows the cross-section of the IBS curves at 20, 40 and 60 months.

The relatively good performance of the separate models approach might be related to the high number of covariates in the models (see Figure 4).

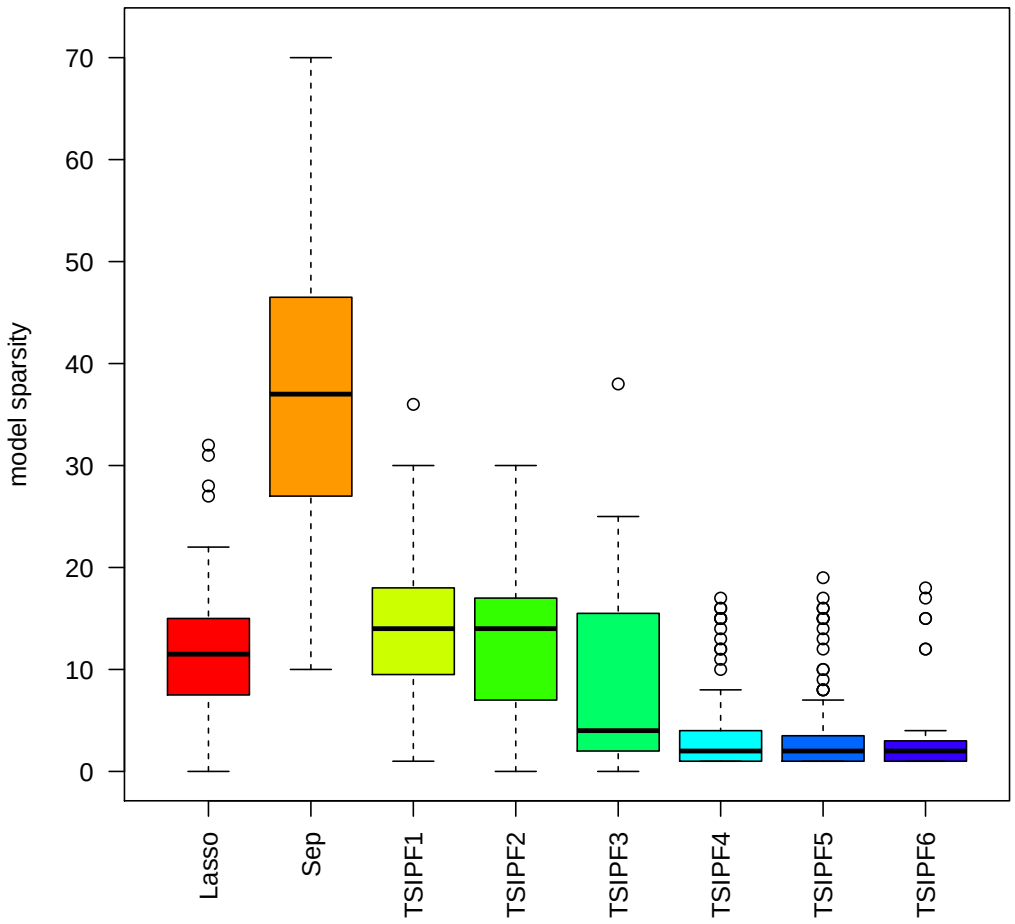


Figure 4: AML data: Number of selected covariates (model sparsity).

The two-step IPF-Lasso models with the separated Step 1 have very low numbers of covariates compared to the TSIPF1 and TSIPF2 versions, but have achieved a similar prediction performance represented by their respective average IBS curves in Figure 3. The TSIPF3 model sparsity is close to the results of TSIPF4 to TSIPF6, but with much higher variation and the highest average IBS curve of all TSIPF-Lasso versions.

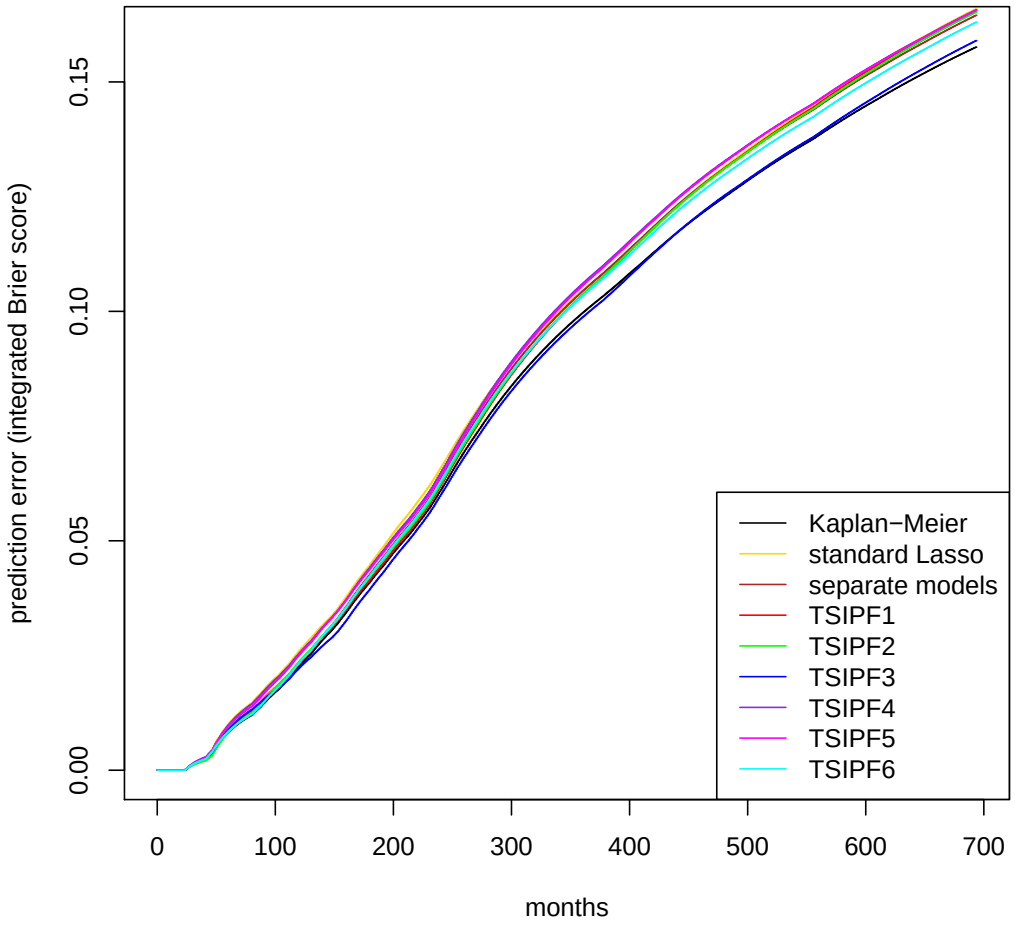


Figure 5: ESCA data: Integrated Brier score (averages over 100 runs at equally spaced time points) for standard Lasso, separate models and the six versions of the TSIPF-Lasso model. The Kaplan-Meier curve serves as reference.

Similar to the AML data, the ESCA data has been analyzed with the same set of models. Again, random splits into a training and test set have been repeated 100 times and the IBS averages over these 100 results for each model have been calculated and plotted against time (Figure 5).

| Model | K-M | Lasso | Sep | TSIPF1 | TSIPF2 | TSIPF3 | TSIPF4 | TSIPF5 | TSIPF6 |
|---|---|---|---|---|---|---|---|---|---|
| IBS200 | 0.047 | 0.051 | 0.049 | 0.047 | 0.048 | 0.046 | 0.050 | 0.050 | 0.049 |
| IBS400 | 0.108 | 0.113 | 0.113 | 0.113 | 0.113 | 0.107 | 0.115 | 0.115 | 0.112 |
| IBS600 | 0.145 | 0.152 | 0.151 | 0.152 | 0.151 | 0.145 | 0.152 | 0.152 | 0.150 |

Table 9: Average integrated Brier score for ESCA data at 200, 400 and 600 days

The curve for the Kaplan-Maier reference is almost completely matching with the TSIPF3 model curve which means that in terms of prediction performance, these two approaches have similar results. However, all other models have IBS curves which lie above the reference curve. Table 9 shows a cross-section of IBS values at 200, 400 and 600 days.

Therefore one needs to conclude that choosing any of the studied models doesn't improve the prediction performance compared to a simple Kaplan-Meier estimation. A possible reasons for this result could be related to the proportional hazards (PH) assumption of the Cox model which is a pre-requisite of all the studied survival models. Tests done for some of the random data splits indicate, that the PH assumption is hurt for some of the covariates and also overall for the respective models. Therefore the Cox PH model might not be the best option for modeling the ESCA data. Another reason could be the low number of covariates in the various models (Figure 6). The median of the model sparsity of the original Lasso model, of the TSIPF3 and of the TSIPF4 models is zero, meaning that 50 out of the 100 models have no covariates. The other 50 models have a very low number of non-zero coefficients, like the standard Lasso model which contains only 1 or 2 covariates. This extreme model sparsity could be a consequence of model misspecification, of failing convergence because of a still too low maxit parameter and therefore higher lambda penalties or of the data itself.

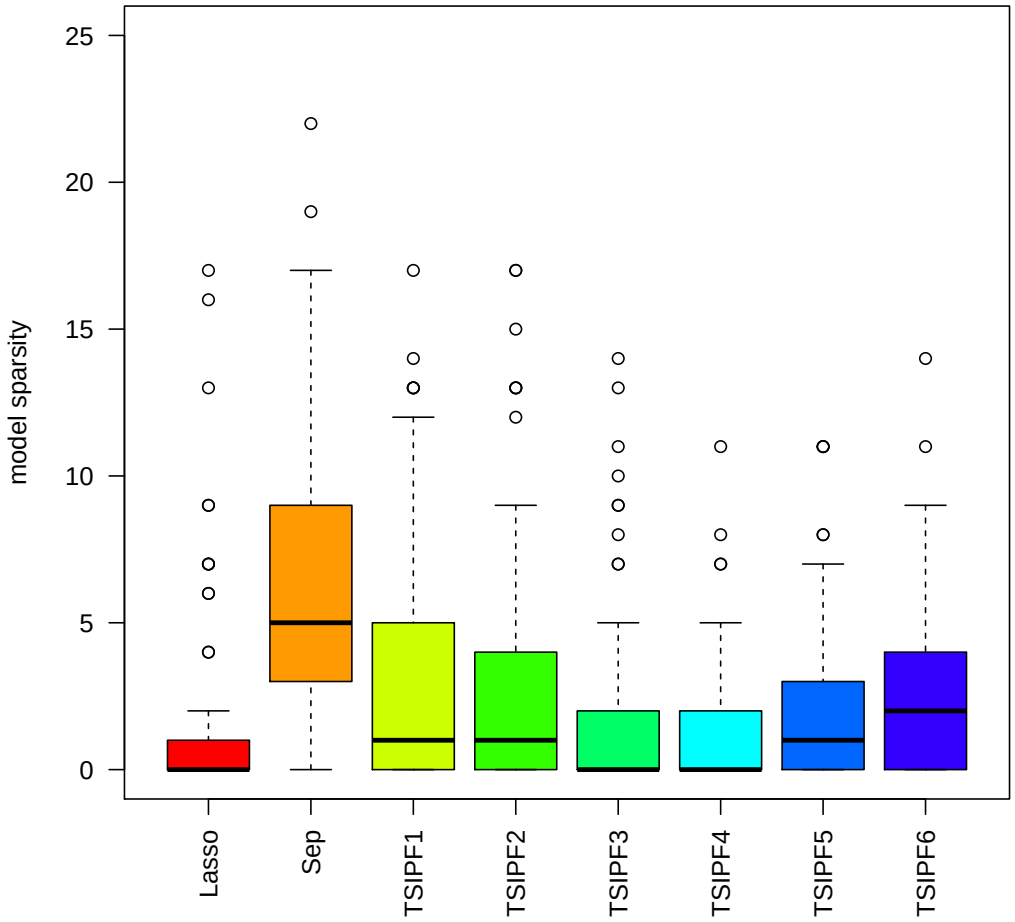


Figure 6: ESCA data: Number of selected covariates (model sparsity).

Similar results have been observed after analyzing the BRCA data. The average IBS curves, a cross-section of specific IBS averages and the boxplots of the model sparsities can be found in Figure 11, Table 15 and Figure 12 in the Appendix.
The response of the KIRP data set is the tumor type which is a binary variable and therefore logistic regression has been applied for all models. Although this data set is the smallest of the four real data sets, neither the SGL nor the IPF-Lasso model could be included into the study because of the earlier mentioned reasons. The prediction performance has been measured by the AUC and the model sparsity. In terms of AUC the standard Lasso, the separated models approach, the TSIPF1 and the TSIPF2 model achieve similar results (Figure 7).

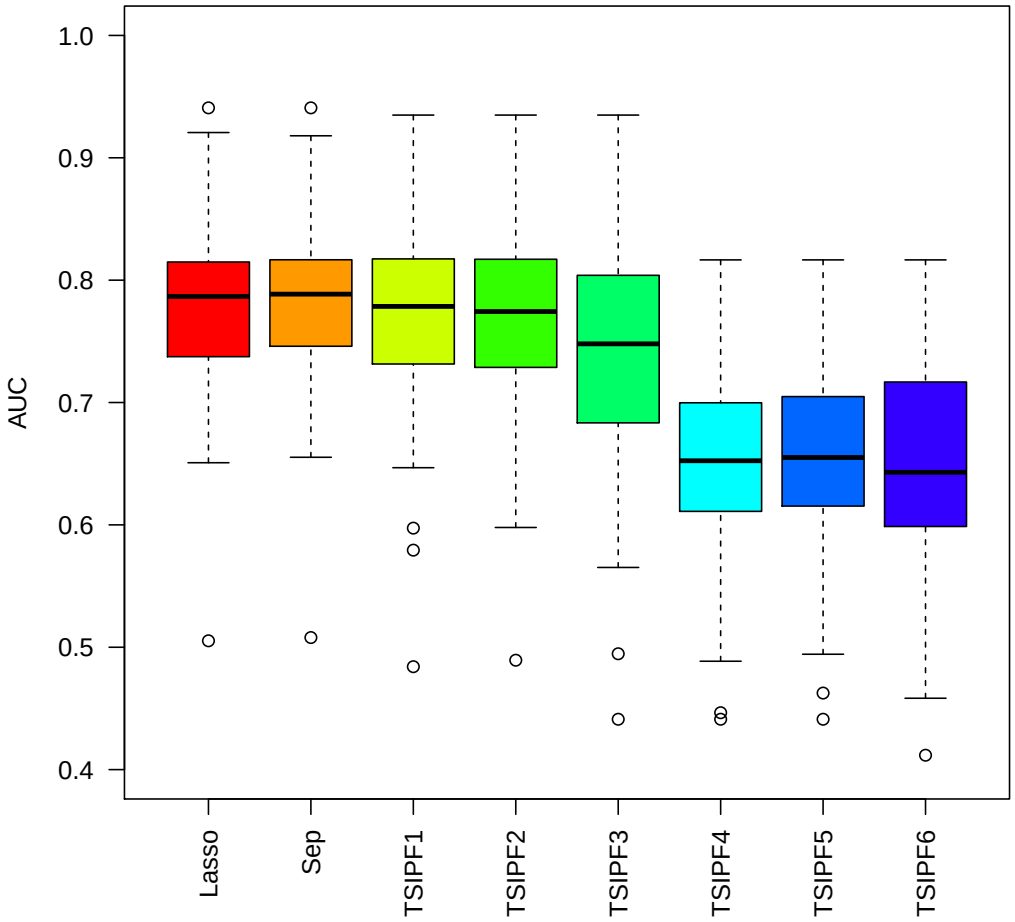


Figure 7: KIRP data: AUC based on 100 runs with random splits into training and test data.

The group of TSIPF-Lasso models with separated Step 1 shows the lowest AUCs and the TSIPF3 model lies between the two groups. Figure 8 displays the model sparsity. The separated models approach generates less sparse results than the rest of the competitors. Among the TSIPF-Lasso models, the two groups with a combined and a separated Step 1 achieve different results compared to each other with the second group having less covariates in the models. The results within each group are the same for the Ridge-Lasso versions and lower for the Lasso-Lasso models. The sparsities of the standard lasso models lie in between the two TSIPF-Lasso groups.

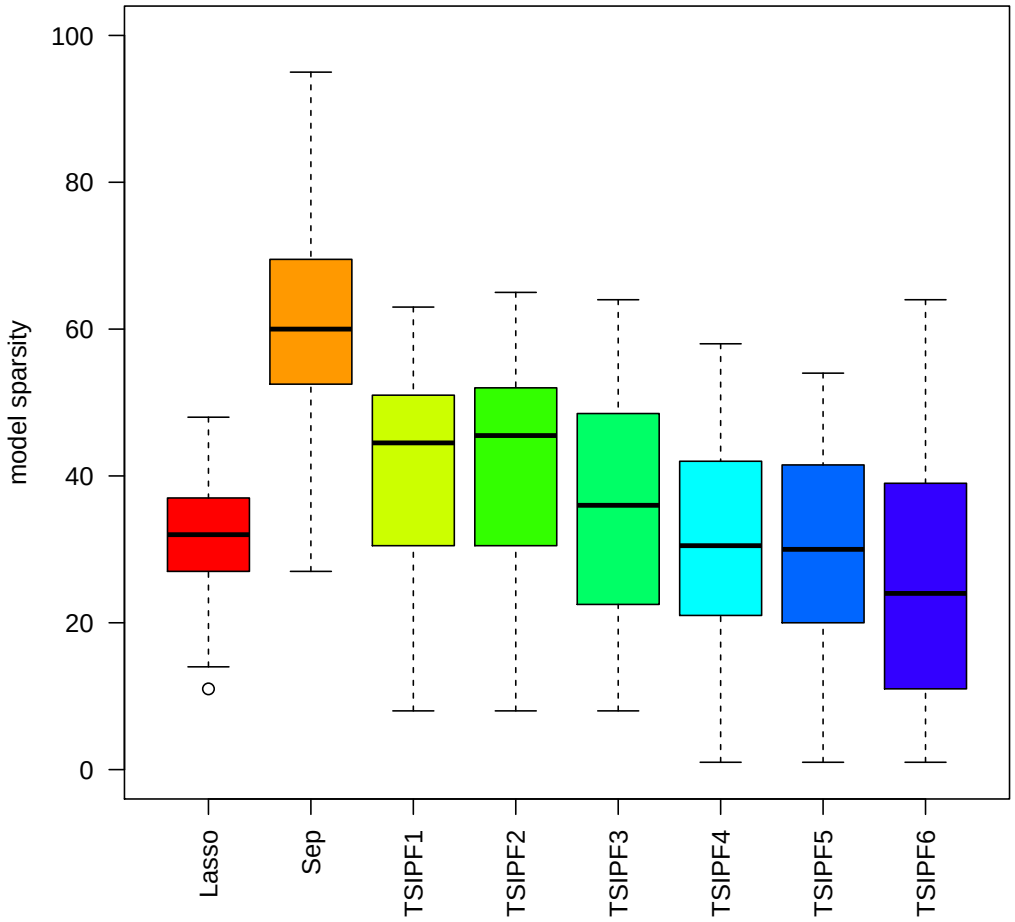


Figure 8: KIRP data: Number of selected covariates (model sparsity).

# 6 Conclusions

This study aimed at investigating two-step extensions of the IPF-Lasso model, in particular related to computation time, prediction error and model sparsity. Correlated and uncorrelated simulation data sets with different scenarios and four real data sets have been analyzed. Six versions of the new two-step IPF-Lasso model have been compared to the IPF-Lasso, the standard Lasso, separate models for each modality and the SGL model. Comparisons have been done in terms of computation time, prediction error, model sparsity and for the simulated data, the proportion of identified relevant covariates has been calculated.

The six TSIPF-Lasso models need 12-15% of the computation time of the IPF-Lasso model for analyzing the simulation data. A direct comparison with the IPF-Lasso model based on real data couldn't be done, but the TSIPF-Lasso models need approximately 20% more time than the standard Lasso model. Running an IPF-Lasso model on four or five modalities would need much more time which increases exponentially with each added modality.

A comparison of the prediction errors shows less obvious differences than computation time. Looking into the results for simulated data, one can see better or worse performances of the TSIPF-Lasso models vs. the IPF-Lasso, depending on the different scenarios. In scenario A, for example, the TSIPF1 and TSIPF2 models have a higher AUC than the IPF-Lasso and in scenario D it is the exact opposite. That is also reflected in the proportions of selected relevant covariates. In scenario A, the TSIPF-Lasso models contain more relevant covariates than the IPF-Lasso and in scenario D the IPF-Lasso model performs better. Considering all results based on simulated data, one can conclude that at least the TSIPF1-, TSIPF2- and TSIPF3-Lasso models show similar AUCs than the IPF-Lasso whereas the other TSIPF-Lasso models have slightly lower results.

In the real data analysis, all TSIPF-Lasso models perform better than the standard Lasso and worse than the separated models approach (for AML data) or slightly below both competitors (for KIRP data). The ESCA and BRCA data analysis show a bad performance for all models compared to the Kaplan-Meier reference, probably due to a model misspecification.

In terms of model sparsity, the medians of the numbers of non-zero covariates

based on the analysis of simulated data are very close to each other for all models apart from the results of the separated models approach and the SGL model which are much higher.

The sparsities of the TSIPF1- and TSIPF2-lasso models are both the same, slightly above the standard Lasso and below the separate models approach for the AML and the KIRP data. The TSIPF3- to TSIPF6-Lasso models have lower sparsities for the KIRP data and very low ones for the AML data.

It is difficult to decide which of the six TSIPF-Lasso models is the best. It seems to make no difference whether the means for Step 2 are estimated by an arithmetic mean or a geometric mean. The results also point rather to the models with the combined Step 1 than to the separated Step 1 models and there especially to the Ridge-Lasso combinations, although the group of separated Step 1 models has a similar prediction performance as the TSIPF1-Lasso model in the AML data analysis.

All these statements about the prediction performance can hardly be generalized and would require more research. The number of real data sets needs to be higher in order to be able to apply powerful statistical tests for comparing the prediction errors of the various candidate models and get statements which can be generalized. This approach would follow an idea of Boulesteix et al. [2017b] where research in methodological computational statistics is compared with clinical trials and the strict requirements in this area. Accordingly, data sets would play the role of patients and the investigated statistical methods correspond to different treatments. Such a full-blown benchmark study would require a high number of omics data sets which might not be easily available. Nevertheless one could calculate at least the power of the applied statistical tests for a given number of data sets and get an idea about the validity of such studies and the drawn conclusions [Boulesteix et al., 2015].

For analyzing high numbers of multi-omics data sets one would need access to appropriate hardware in order to reduce the overall computation time. Another option for reducing the computation time would be the application of pre-selection rules to the covariates. The so-called strong rules for discarding covariates are already implemented in the glmnet package [Tibshirani et al., 2010] and the new

R-package biglasso contains a selection of several alternative rules [Zeng and Breheny, 2017]. The biglasso package also enables the analysis of big data sets in a situation where there is not sufficient RAM volume available. Using biglasso for studying the performance of the two-step IPF-Lasso model vs. IPF-Lasso and its competitors would require replacing the glmnet-functions in the original IPF-Lasso script by the corresponding biglasso functions.

It would be also interesting to extend the study to alternatives of the Cox PH model. Parametric survival models which don't assume proportional hazard rates might better fit to the data to be analyzed. One of the outcomes of this study were the stability problems of the SGL package with big data sets. It is not clear whether these stability problems could be reduced or avoided by changing the second tuning parameter $\alpha$, which was fixed at the default value during this study. Nevertheless, the SGL package seems to work not as efficient as the glmnet package. In their paper Wang and Ye [2014] mention that "...in large-scale applications, the complexity of the regularizers entails great computational challenges."(p.1) and present a modification of the SGL model which "...improves the efficiency of SGL by orders of magnitude."(p.1). This approach could finally enable a comparison of the two-step IPF-Lasso models to the SGL on big real data sets.

One could also modify the penalties in the second step of the two-step IPF-Lasso model similar to the elastic net or fundamentally change the approach of assigning individual penalties to multi-omics data like Velten [2017] did with their hierarchical Bayes model. Bayes models introduce subjective elements into the data analysis through prior distributions. The IPF-Lasso model also contains a subjective element, as a practitioner has to choose appropriate penalty factors. A characteristic that has been lost by the completely data driven approach of the two-step IPF-Lasso model.

# References


Anne-Laure Boulesteix and Mathias Fuchs. *ipflasso: Integrative Lasso with Penalty Factors*, 2015. URL `https://CRAN.R-project.org/package=ipflasso`. R package version 0.1.

Anne-Laure Boulesteix and Willi Sauerbrei. Added predictive value of high-throughput molecular data to clinical data and its validation. *Briefings in bioinformatics*, 12(3):215–229, 2011.

Anne-Laure Boulesteix, Carolin Strobl, Thomas Augustin, and Martin Daumer. Evaluating microarray-based classifiers: An overview. *Cancer Informatics*, 6: 77–97, 2008.

Anne-Laure Boulesteix, Robert Hable, Sabine Lauer, and Manuel J. A. Eugster. A statistical framework for hypothesis testing in real data comparison studies. *The American Statistician*, 69(3):201–212, 2015.

Anne-Laure Boulesteix, Riccardo De Bin, Xiaoyu Jiang, and Mathias Fuchs. Ipflasso: Integrative-penalized regression with penalty factors for prediction based on multi-omics data. *Computational and mathematical methods in medicine*, 2017, 2017a.

Anne-Laure Boulesteix, Rory Wilson, and Alexander Hapfelmeier. Towards evidence-based computational statistics: lessons from clinical research on the role and design of real-data benchmark studies. *BMC Medical Research Methodology*, 17(138), 2017b.

Peter Bühlmann and Sarah van de Geer. *Statistics for high-dimensional data*. Springer, Berlin Heidelberg, 2011.

Ludwig Fahrmeir, Thomas Kneib, and Stefan Lang. *Regression*. Springer, Heidelberg Dordrecht London New York, 2nd edition, 2009.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010. URL `http://www.jstatsoft.org/v33/i01/`.

Max Kuhn. Contributions from Jed Wing, Steve Weston, Andre Williams, Chris Keefer, Allan Engelhardt, Tony Cooper, Zachary Mayer, Brenton Kenkel, the R Core Team, Michael Benesty, Reynald Lescarbeau, Andrew Ziem, Luca Scrucca, Yuan Tang, Can Candan, and Tyler Hunt. *caret: Classification and Regression Training*, 2017. URL `https://CRAN.R-project.org/package=caret`. R package version 6.0-76.

Alan Genz, Frank Bretz, Tetsuhisa Miwa, Xuefei Mi, Friedrich Leisch, Fabian Scheipl, and Torsten Hothorn. *mvtnorm: Multivariate Normal and t Distributions*, 2017. URL `https://CRAN.R-project.org/package=mvtnorm`. R package version 1.0-6.

Thomas A. Gerds, Tianxi Cai, and Martin Schumacher. The performance of risk prediction models. *Biometrical Journal*, 50:457–479, 2008.

Erika Graf, Claudia Schmoor, Willi Sauerbrei, and Martin Schumacher. Assessment and comparison of prognostic classification schemes for survival data. *Statistics in Medicine*, 18:2529–2545, 1999.

Yehudit Hasin, Marcus Seldin, and Aldons Lusis. Multi-omics approaches to disease. *Genome biology*, 18(1):83, 2017.

Trevor Hastie, Robert Tibschirani, and Jerome Friedman. *The elements of statistical learning*. Springer, New York, 2nd edition edition, 2009.

Trevor Hastie, Robert Tibshirani, and Martin Wainwright. *Statistical learning with sparsity: the lasso and generalizations*. CRC press, 2015.

Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

Jian Huang, Shuangge Ma, and Cun-Hui Zhang. Adaptive lasso for sparse high-dimensional regression models. *Statistica Sinica*, pages 1603–1618, 2008.

Sijia Huang, Kumardeep Chaudhary, and Lana X Garmire. More is better: Recent progress in multi-omics data integration methods. *Frontiers in Genetics*, 8:84, 2017.

Ulla B. Mogensen, Hemant Ishwaran, and Thomas A. Gerds. Evaluating random forests for survival analysis using prediction error curves. *Journal of Statistical Software*, 50(11):1–23, 2012. URL `http://www.jstatsoft.org/v50/i11/`.

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/`.

William Revelle. *psych: Procedures for Psychological, Psychometric, and Personality Research*. Northwestern University, Evanston, Illinois, 2017. URL `https://CRAN.R-project.org/package=psych`. R package version 1.7.5.

Noah Simon, Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for cox's proportional hazards model via coordinate descent. *Journal of Statistical Software*, 39(5):1–13, 2011. URL `http://www.jstatsoft.org/v39/i05/`.

Noah Simon, Jerome Friedman, Trevor Hastie, and Rob Tibshirani. *SGL: Fit a GLM (or cox model) with a combination of lasso and group lasso regularization*, 2013a. URL `https://CRAN.R-project.org/package=SGL`. R package version 1.1.

Noah Simon, Jerome Friedman, Trevor Hastie, and Robert Tibshirani. A sparse-group lasso. *Journal of Computational and Graphical Statistics*, 22(2):231–245, 2013b.

Lincoln D Stein. The case for cloud computing in genome informatics. *Genome biology*, 11(5):207, 2010.

Feng Tai and Wei Pan. Incorporating prior knowledge of predictors into penalized classifiers with multiple penalty terms. *Bioinformatics*, 23(14):1775–1782, 2007.

Nils Ternès, Federico Rotolo, Georg Heinze, and Stefan Michiels. Identification of biomarker-by-treatment interactions in randomized clinical trials with survival outcomes and high-dimensional spaces. *Biometrical Journal*, 59(4):685–701, 2017.

Terry M. Therneau. *A Package for Survival Analysis in S*, 2015. URL `https://CRAN.R-project.org/package=survival`. R package version 2.38.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

Robert Tibshirani. Regression shrinkage and selection via the lasso: a retrospective. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(3):273–282, 2011.

Robert Tibshirani, Jacob Bien, Jerome Friedman, Trevor Hastie, Noah Simon, Jonathan Taylor, and Ryan J. Tibshirani. Strong rules for discarding predictors in lasso-type problems. *Journal of the Royal Statistical Society. Series B (Methodological)*, 74(2):245–266, 2010.

Katarzyna Tomczak, Patrycja Czerwińska, and Maciej Wiznerowicz. The cancer genome atlas (tcga): an immeasurable source of knowledge. *Contemporary oncology*, 19(1A):A68, 2015.

Britta Velten. Adaptive penalization for high-dimensional regression and classification with non-exchangeable features. Presentation at the 49th workshop of the working groups Statistical Computing (GMDS/IBS-DR) and Klassifikation und Datenanalyse in den Biowissenschaften (GfKl), Schloss Reisensburg (Günzburg, Germany), 2017.

Jie Wang and Jieping Ye. Two-layer feature reduction for sparse-group lasso via decomposition of convex sets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2132–2140. Curran Associates, Inc., 2014.

Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

Yaohui Zeng and Patrick Breheny. The biglasso package: A memory- and computation-efficient solver for lasso model fitting with big data in r. arXiv:1701.05936, 2017.

Qing Zhao, Xingjie Shi, Yang Xie, Jian Huang, BenChang Shia, and Shuangge Ma. Combining multidimensional genomic measurements for predicting cancer prognosis: observations from tcga. *Briefings in bioinformatics*, 16(2):291–303, 2014.

Hui Zou. The adaptive lasso and its oracle properties. *Journal of the American statistical association*, 101(476):1418–1429, 2006.

# A Figures

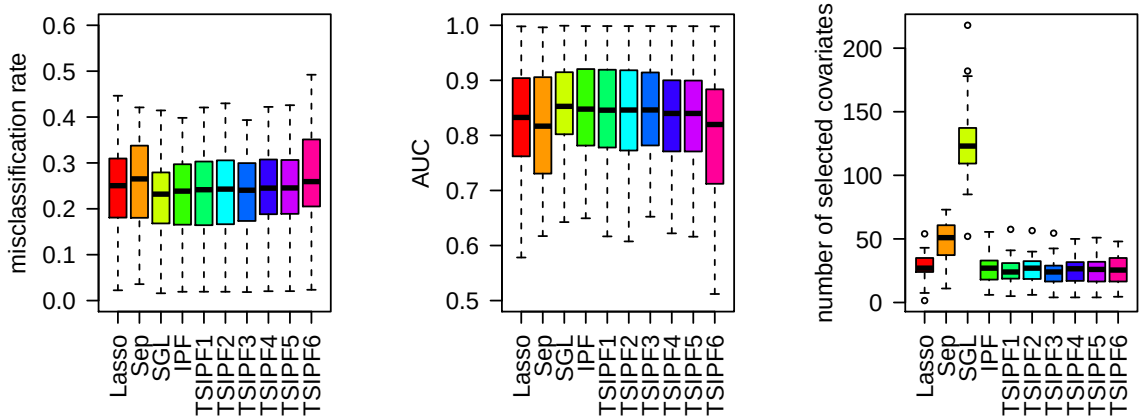


Figure 9: Median misclassification rate (left), median AUC (center) and median number of selected covariates (right) of 39 simulation scenarios (uncorrelated data).

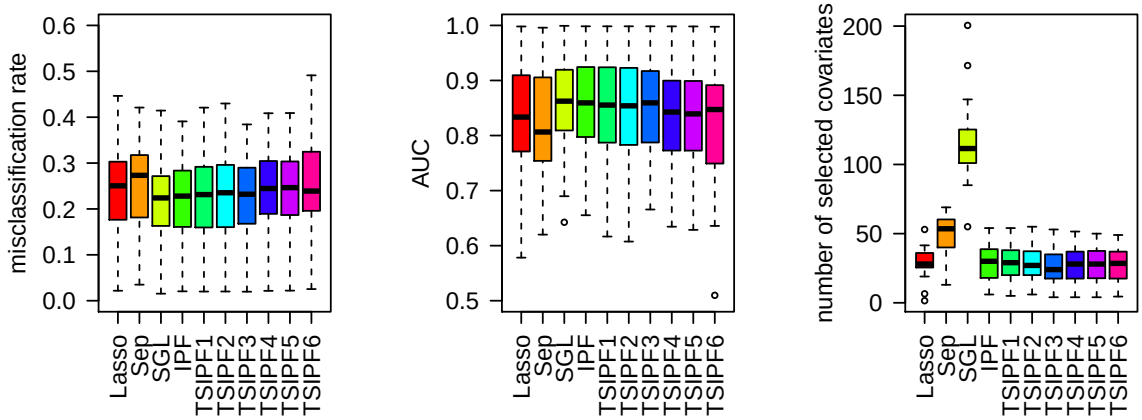


Figure 10: Median misclassification rate (left), median AUC (center) and median number of selected covariates (right) of 39 simulation scenarios (correlated data).

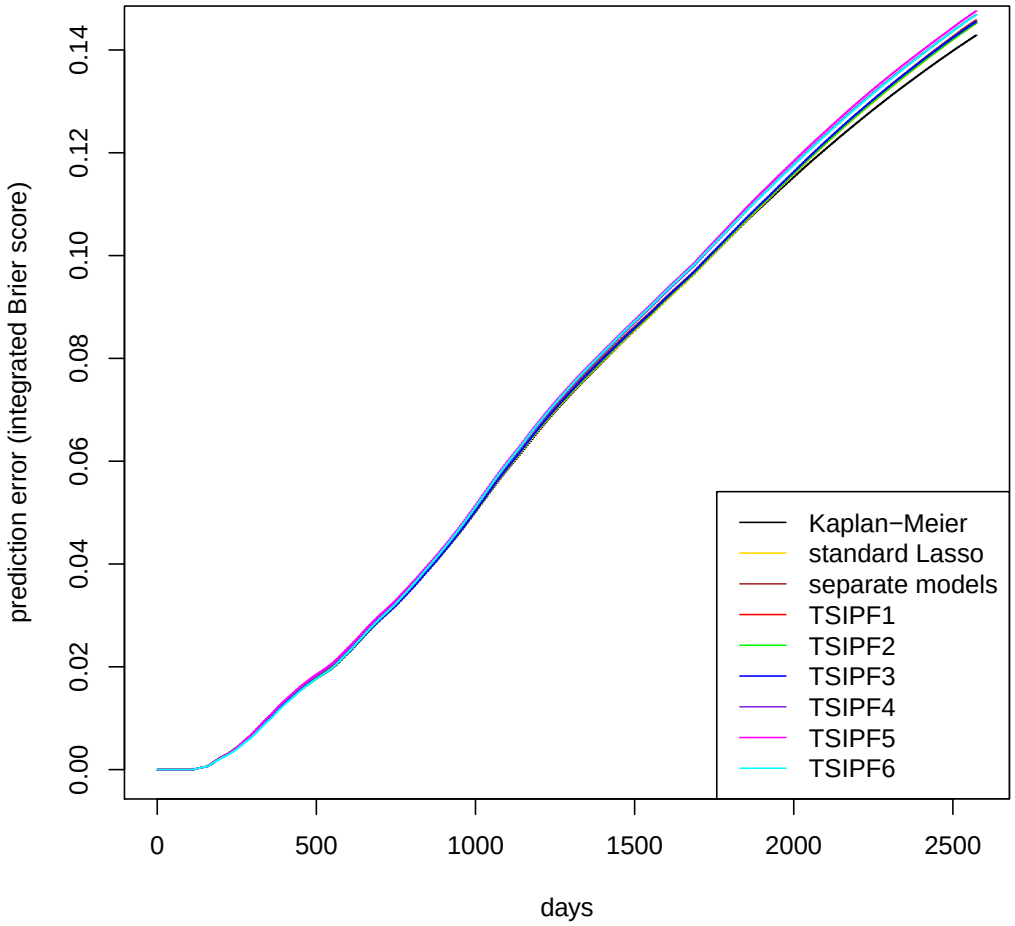


Figure 11: BRCA data: Integrated Brier score (averages over 100 runs at equally spaced time points) for standard Lasso, separate models and the six versions of the TSIPF-Lasso model. The Kaplan-Meier curve serves as reference.

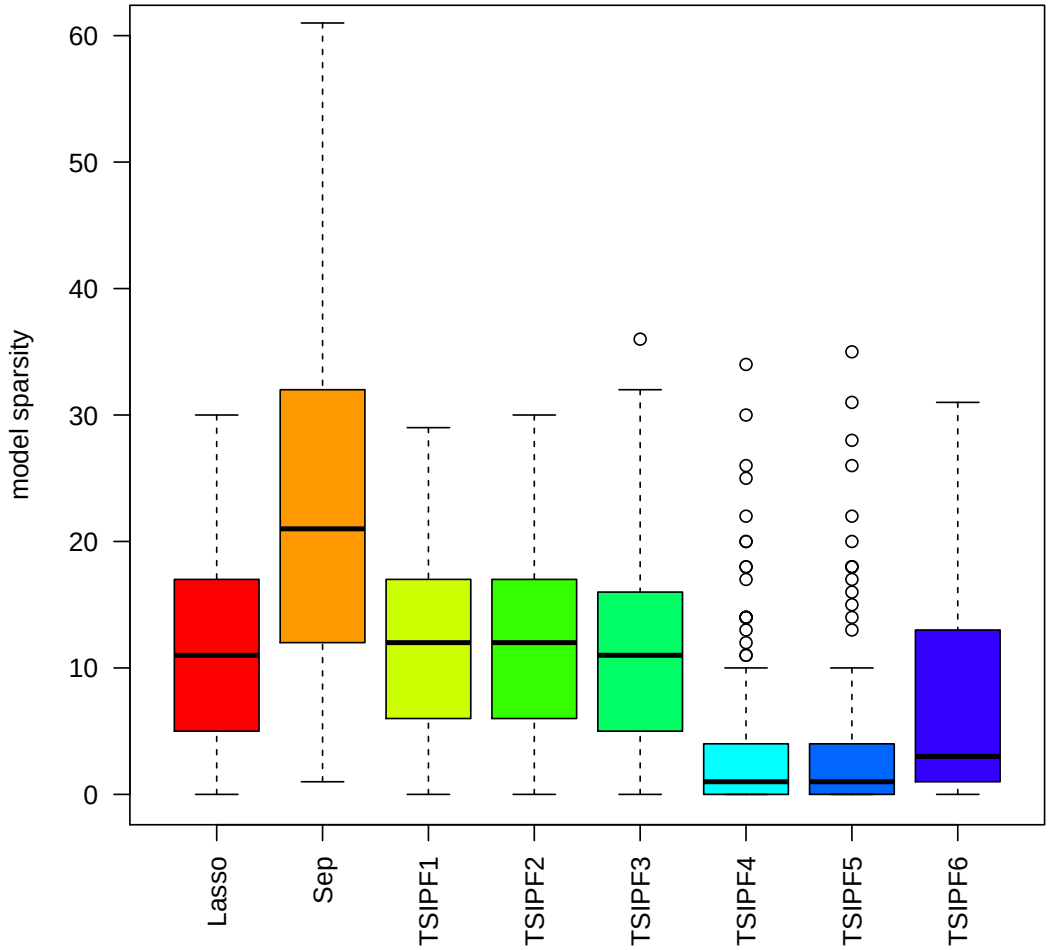


Figure 12: BRCA data: Number of selected covariates (model sparsity) for standard Lasso, separate models (Sep) and the six versions of the TSIPF-Lasso model.

# B Tables

| Setting | $p_1$ | $p_2$ | $p_1^r$ | $p_2^r$ | $\beta_1$ | $\beta_2$ |
|---|---|---|---|---|---|---|
| 1 | 500 | 500 | 10 | 10 | 0.5 | 0.5 |
| 2 | 500 | 500 | 20 | 0 | 0.5 | - |
| 3 | 500 | 500 | 10 | 10 | 0.3 | 0.8 |
| 4 | 100 | 1000 | 20 | 0 | 0.5 | - |
| 5 | 100 | 1000 | 0 | 20 | - | 0.5 |
| 6 | 100 | 100 | 10 | 10 | 0.5 | 0.5 |
| 7 | 500 | 500 | 3 | 3 | 1 | 1.5 |
| 8 | 500 | 500 | 20 | 20 | 1 | 0.3 |
| 9 | 500 | 500 | 20 | 0 | 1 | - |
| 10 | 500 | 500 | 40 | 0 | 0.3 | - |
| 11 | 500 | 500 | 40 | 0 | 0.5 | - |
| 12 | 500 | 500 | 20 | 10 | 0.5 | 0.5 |
| 13 | 500 | 500 | 10 | 10 | 0.3 | 1.5 |
| 14 | 500 | 500 | 10 | 10 | 0.4 | 0.7 |
| 15 | 20 | 2000 | 10 | 10 | 0.5 | 0.5 |
| 16 | 300 | 800 | 10 | 10 | 0.5 | 0.5 |
| 17 | 20 | 2000 | 10 | 10 | 1 | 0.3 |

| Setting | $p_1$ | $p_2$ | $p_1^r$ | $p_2^r$ | $\beta_1$ | $\beta_2$ |
|---|---|---|---|---|---|---|
| 18 | 20 | 2000 | 1 | 100 | 0.5 | 0.5 |
| 19 | 300 | 800 | 3 | 8 | 0.8 | 0.8 |
| 20 | 100 | 1000 | 3 | 30 | 1 | 0.3 |
| 21 | 20 | 2000 | 20 | 0 | 0.5 | - |
| 22 | 300 | 800 | 20 | 0 | 0.5 | - |
| 23 | 100 | 1000 | 20 | 10 | 0.5 | 0.5 |
| 24 | 100 | 1000 | 40 | 0 | 0.5 | - |
| 25 | 100 | 1000 | 20 | 0 | 1 | - |
| 26 | 20 | 2000 | 0 | 20 | - | 0.5 |
| 27 | 300 | 800 | 0 | 20 | - | 0.5 |
| 28 | 100 | 1000 | 0 | 20 | - | 1 |
| 29 | 100 | 1000 | 0 | 40 | - | 0.5 |
| 30 | 100 | 1000 | 10 | 20 | 0.5 | 0.5 |
| 31 | 20 | 1000 | 5 | 0 | 1 | - |
| 32 | 20 | 1000 | 10 | 10 | 0.5 | 0.5 |
| 33 | 20 | 1000 | 3 | 3 | 1 | 1 |

Table 10: Additional simulation scenarios for correlated and uncorrelated data

| Data | Covariates | Type |
|---|---|---|
| AML | gender | factor/2 levels |
| | age | integer |
| | % bone marrow blast | integer |
| | white blood-cell count | numerical |
| BRCA | age | integer |
| | surgical procedure | factor/5 levels |
| | histological type | factor/9 levels |
| ESCA | gender | factor/2 levels |
| | age | integer |
| | primary pathology histological type | factor/2 levels |
| | primary pathology neoplasm histologic grade | factor/4 levels |
| KIRP | gender | factor/2 levels |
| | age | integer |
| | height | numerical |
| | weight | numerical |
| | laterality | factor/3 levels |

Table 11: Clinical covariates in TCGA datasets

| Model | Computation time (average in seconds) | Computation time (% of IPF-Lasso) |
|---|---|---|
| Standard Lasso | 2.6 | 11.9 |
| Separate models | 3.8 | 17.0 |
| SGL | 29.1 | 131.3 |
| TSIPF1 | 3.2 | 14.6 |
| TSIPF2 | 3.3 | 14.7 |
| TSIPF3 | 2.5 | 11.2 |
| TSIPF4 | 3.2 | 14.3 |
| TSIPF5 | 3.2 | 14.3 |
| TSIPF6 | 2.6 | 11.8 |
| IPF-Lasso | 22.2 | 100.0 |

Table 12: Average computation times in seconds and percent computation times of IPF-Lasso over 100 runs for all studied models under simulation scenarios A'-F' (correlated data), measured at an 8 GB RAM laptop with a 2.7 GHz CPU.

| Model | Computation time (minutes) |
|---|---|
| Standard Lasso | 8.48 |
| Separate models | 6.39 |
| SGL | - |
| TSIPF1 | 9.26 |
| TSIPF2 | 7.27 |
| TSIPF3 | 5.16 |
| TSIPF4 | 6.27 |
| TSIPF5 | 6.06 |
| TSIPF6 | 4.42 |
| IPF-Lasso | - |

Table 13: Computation times for ESCA data: Single runs on a 4.2 GHz CPU/128 GB RAM workstation with 5-fold CV and 10 repetitions for each model.

| | Scenario | IPF | TSIPF1 | TSIPF2 | TSIPF3 | TSIPF4 | TSIPF5 | TSIPF6 |
|---|---|---|---|---|---|---|---|---|
| Modality 1 | A' | 0.40 | 0.45 | 0.46 | 0.39 | 0.30 | 0.30 | 0.32 |
| | B' | 0.33 | 0.45 | 0.45 | 0.36 | 0.60 | 0.60 | 0.62 |
| | C' | 0.69 | 0.60 | 0.58 | 0.74 | 0.74 | 0.73 | 0.73 |
| | D' | 0.52 | 0.29 | 0.30 | 0.46 | 0.48 | 0.48 | 0.50 |
| | E' | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| | F' | 0.89 | 0.80 | 0.82 | 0.90 | 0.90 | 0.90 | 0.90 |
| Modality 2 | A' | 0.36 | 0.43 | 0.44 | 0.34 | 0.32 | 0.32 | 0.30 |
| | B' | 0.48 | 0.47 | 0.48 | 0.44 | 0.32 | 0.32 | 0.20 |
| | C' | 0.21 | 0.41 | 0.43 | 0.08 | 0.08 | 0.08 | 0.02 |
| | D' | - | - | - | - | - | - | - |
| | E' | 0.02 | 0.04 | 0.06 | 0.00 | 0.00 | 0.00 | 0.00 |
| | F' | 0.10 | 0.23 | 0.20 | 0.02 | 0.02 | 0.02 | 0.00 |

Table 14: Proportions of discovered relevant covariates for simulation scenarios A'-F' (correlated data).

| Model | K-M | Lasso | Sep | TSIPF1 | TSIPF2 | TSIPF3 | TSIPF4 | TSIPF5 | TSIPF6 |
|---|---|---|---|---|---|---|---|---|---|
| IBS500 | 0.017 | 0.017 | 0.018 | 0.017 | 0.017 | 0.017 | 0.018 | 0.018 | 0.017 |
| IBS1500 | 0.085 | 0.085 | 0.086 | 0.085 | 0.085 | 0.085 | 0.087 | 0.087 | 0.087 |
| IBS2500 | 0.140 | 0.142 | 0.142 | 0.142 | 0.142 | 0.142 | 0.143 | 0.144 | 0.143 |

Table 15: Average integrated Brier score for BRCA data at 500, 1500 and 2500 days

# C Electronic appendix

The electronic appendix contains the major R-scripts and the real data analyzed in this study. It consists of a Readme-file (this text), a folder *Simulation* and a folder *Real Data Analysis* with the sub-folders *AML data*, *BRCA data*, *ESCA data* and *KIRP data*. Table 16 shows the contents of each folder and subfolder.

| Folders | Simulation | Real Data Analysis | | | |
|---|---|---|---|---|---|
| Subfolders | | AML data | BRCA data | ESCA data | KIRP data |
| R-scripts | simu.function<br>simu.script<br>fig.function<br>fig.script<br>tab.function<br>tab.script | AML.parallel<br>AML.linear<br>AML.plotfunctions<br>AML.make.plots | BRCA.parallel<br>BRCA.linear<br>BRCA.plotfunctions<br>BRCA.make.plots | ESCA.parallel<br>ESCA.linear<br>ESCA.plotfunctions<br>ESCA.make.plots | KIRP.analysis<br>KIRP.plotfunctions<br>KIRP.make.plots |
| Data sets | | xyfinal | xyfinal | xyfinal | xyfinal |

Table 16: Folder structure (Electronic appendix)

Simulation:

The R-script *simu.function* simulates data and runs a standard Lasso, separate models, SGL, six versions of the two-step IPF-Lasso and the IPF-Lasso. It contains the functions *simulclassif*, *covMatrix* and *simulation*.

The R-script *simu.script* defines the data scenarios A to F plus 33 additional scenarios for uncorrelated data and A' to F' plus 33 additional scenarios for correlated data. It calls the function *covMatrix* to generate covariance matrices and the function *simulation* which returns the file *result* and saves it in a folder *results_uncorrelated* or *results_correlated*.

The R-script *fig.function* defines the functions *plotfunction*, *plotmedfunction* and *perfmed*. *Plotfunction* creates boxplots of misclassification rate, AUC and number of non-zero coefficients for all tested models, using the *result* file as input. *Perfmed* calculates medians of misclassification rate, AUC and number of non-zero coefficients of all tested models. *Plotmedfunction* generates boxplots of the 39 median misclassification rates, median AUCs and median numbers of non-zero coefficients from the analysis of each data type and for each tested model.

The R-script *fig.script* generates the input to Figures 1 and 2 by using *plotfunction*, and the overall median results (Figures 9 and 10) for the 39 scenarios of uncorrelated and correlated data by using *plotmedfunction*.
The function *relcov* (R-script *tab.function*) calculates the proportions of discovered relevant covariates for each tested model, using the *result* files as input. *Tab.script* generates the input for Table 10 (additional scenarios) and for Tables 7 and 14 (proportions of discovered relevant covariates).

Real Data Analysis/AML data:
The R-script *AML.parallel* generates a list of prediction errors and coefficients for the standard Lasso, separate models, six versions of the two-step IPF-Lasso and the IPF-Lasso. All models are Cox PH models and the calculations are done in parallel by a pre-specified number of cores. The script contains also code for the SGL model, but this hasn't been included into the analysis. It uses the data file *xyfinal* as input and returns a file called *res*.
The R-script *AML.linear* is the linear version of *AML.parallel*.
The R-script *AML.plotfunction* defines the functions *ibsmeans*, *timeminmax* and *spars*. *Timeminmax* calculates the minimum of the maximum values of the time axis of all model results based on the 100 random data splits. *Ibsmeans* calculates the IBS averages over the 100 repetitions and returns a matrix with these averages for each model. The function *spars* determines the number of non-zero coefficients for each model over all runs. Inputs for all functions are the results of the *AML.parallel* script.
The R-script *AML.make.plots* generates Figure 3, Figure 4 and Table 8 by using the functions of *AML.plotfunction*.

Real Data Analysis/BRCA data:
Similar to AML data for Figure 11, Figure 12 and Table 15.

Real Data Analysis/ESCA data:
Similar to AML data for Figure 5, Figure 6 and Table 9.

Real Data Analysis/KIRP data:

The R-script *KIRP.analysis* generates lists of the AUC, the number of non-zero coefficients and the actual coefficients for each random split of the data and each tested model.

The R-script *plot.functions* contains the functions *aucmat* and *spars*. *Aucmat* returns a matrix of AUC values for each model over all runs and the function *spars* determines the number of non-zero coefficients for each model over all runs. Inputs to these functions are the lists, created by *KIRP.analysis*.

The R-script *KIRP.make.plots* generates boxplots of the AUC and the number of non-zero coefficients generated by the functions *aucmat* and *spars*.

# Picture credits

Source of image on title page: "Master in Omics Data Analysis", Universitat de VIC/ Universitat Central de Catalunya, http://mon.uvic.cat/master-omics (last time accessed on September 4, 2017).

# Declaration of Authorship

I hereby confirm that I have authored this Master's thesis independently and without use of others than the indicated sources. All passages which are literally or in general matter taken out of publications or other sources are marked as such.

München, November 9, 2017

Gerhard Schulze