Master's Thesis

Large-scale benchmark study of prediction methods using multi-omics data

Moritz Herrmann

Ludwig-Maximilians-Universität München

Institut für Statistik

Supervisor: Prof. Dr. Anne-Laure Boulesteix

Institut für Medizinische Informationsverarbeitung, Biometrie und Epidemiologie (IBE)

Augsburg, December 24, 2018

Abstract

This study provides a large-scale benchmark experiment for survival time prediction based on multi-omics data for 18 cancer types from the Cancer Genome Atlas (TCGA). Several complex prediction methods from the fields of statistics and machine learning, comprising two boosting methods, three Lasso-based methods and two random forest variants, are compared. At that, the methods vary in their use of the multi-omics data by including the group structure in different ways. As reference a simple Cox model only using clinical variables and the Kaplan-Meier estimate are used, which are standard methods in the context of survival prediction.

The findings show that none of the complex methods using the whole multi-omics data clearly outperforms the standard Cox model only using the clinical variables on average over all data sets. Only likelihood-based boosting favoring clinical variables performs comparable. This indicates the importance of clinical variables. Nevertheless, for several data sets there is at least one complex method outperforming the Cox model. Thus, the findings show that using multi-omics data may lead to better prediction performance. At that, it becomes evident that learners using the group structure outperform in general the naive methods treating all features equally. Moreover, the findings indicate that the clinical variables should be favored, whether or not the molecular variables are distinguished.

Among the naive methods, random forest shows a tendency to outperform the other methods. Furthermore, likelihood-based boosting clearly outperforms priority-Lasso among the methods favoring clinical variables. The Lasso variants using the multi-omics structure outperform the standard Lasso.

Contents

List of Figures

List of Tables

1 Introduction

In the last two decades high-throughput technologies made data stemming from molecular processes available on a large scale and for many patients. Starting from the analysis of whole genomes, other molecular subject matters such as RNA levels or peptide characteristics came into focus with the advancing technologies (Hasin et al, 2017). Hasin et al (2017, p. 83) point out that adding "'omics' [to such] a molecular term implies comprehensive, or global, assessment of a set of molecules". Thus, several omics objects are under investigation in several disciplines today, comprising genomics, epigenomics, transcriptomics, proteomics, metabolomics, or microbiomics.

From a statistical and practical perspective it is of interest to include such data in prediction models to predict outcomes like survival times or the occurrence of specific diseases. At the beginning, only data from a single omics type was used to build such prediction models, together or without standard clinical data (Boulesteix and Sauerbrei, 2011). Since more and more omics types are easily available, in recent years the integration and combined use of several omics groups for the outcome prediction came into focus. This led to the term of multi-omics data, where data of different omics types is present in one data set. Several important questions arise in this context: Whether and how to include such multi-omics data? Does treating all multi-omics types equally suffice or does the inclusion of the group structure information into the prediction model lead to better prediction performances? Which methods are best suited to fulfil the several requirements from practical and statistical perspective?

An important aspect that comes along with multi-omics is the high-dimension of the resulting data sets, which not infrequently have more than 100,000 variables. This makes special demands on the methods used to build the prediction models. Above all, they must be able to handle data where the

number of covariates exceeds the number of observations by far. Moreover, often practitioners prefer sparse and interpretable models including only few variables (Klau et al, 2018).
Several methods have been specifically proposed to handle multi-omics data. Other already established methods from the fields of statistics and machine learning seem reasonable to be used in such a context. Although studies have shown that they lead to promising results, these findings have been obtained based only on a small amount of data sets, leading to illustrative method comparisons (Boulesteix et al, 2013). To our knowledge there is still the lack of studies that neutrally compare the performance of prediction methods on the basis of several multi-omics data sets.
The study at hand aims at providing such a large-scale benchmark study for prediction methods using multi-omics data. It is based on 18 cancer data sets from the Cancer Genome Atlas (TCGA) and focuses on survival time prediction. We use several methods that are based on three widely used modelling approaches from the fields of statistics and machine learning: the Lasso, statistical boosting, and random forest. The aim is to investigate how different forms of multi-omics data inclusion influence the prediction performance and to compare the performance of the methods in that setting, especially with respect to prediction accuracy. In addition, the added predictive value of multi-omics data is assessed.

The study is structured as follows: In the *Background* section we outline the theory behind the benchmark study, comprising several aspects: The *Survival prediction and multi-omics data* subsection provides the basis of survival or time-to-event modelling and prediction and describes multi-omics data in detail. In the *Assessing prediction methods* subsection the characteristics of a sound comparison study, the concept of the added predictive value of molecular data, and performance measures for survival prediction are described. In the subsequent *Overview of methods* subsection the methods used in the study are described in general, in the following subsections *Statistical boosting*, *Lasso methods*, and *Random forest* the methods of each modelling approach are outlined in detail. It follows the description of the benchmark experiment in the *Benchmark experiment* section, which includes the characteristics of the data sets and the actual implementations/configurations of the methods used. In the *Results* section the findings of the study are presented. Finally, the findings are discussed and a conclusion is drawn in the *Discussion and Conclusion* section.

2 Background

In this section we discuss the underlying theory of the conducted benchmark experiment. We describe multi-omics as special kind of high-dimensional data and the basis of sound comparison studies within this area and in general. Furthermore, the three major prediction approaches used within the benchmark study are outlined.

2.1 Survival prediction and multi-omics data

The focus of this study is to investigate and compare the performance of several prediction methods in time-to-event/survival contexts, taking into account a special kind of data denoted as *multi-omics*. In this section we will give a short recap of the underlying formalisation of survival analysis and describe multi-omics data and their potential benefit when predicting survival.

In survival analysis one observes data of the form $D = (\mathbf{t}, \mathbf{d}, \mathbf{X})$, with $\mathbf{X}$ the $N \times p$ matrix of independent variables/prognostic factors, $\mathbf{t} = (t_1, ..., t_N)^T$ the vector of event times and $\mathbf{d} = (\delta_1, ..., \delta_N)^T$ the vector of censoring indicators, where δ_i indicates whether the event of interest has occurred at time t_i. Moreover, $i = 1, ..., N$ and N is the number of observations. Furthermore, with $x_i = (x_{i1}, ..., x_{ip})^T$ we denote the ith row and with $x_{\cdot j} = (x_{1j}, ..., x_{Nj})^T$ the jth column of $\mathbf{X}$. In the following the term *features* is used for the independent variables $x_{\cdot 1}, ..., x_{\cdot p}$.

An observation is said to be censored, if the occurrence of the event is not observed. Although there are different types of censoring, we focus on *right-censoring*, meaning, for example, a patient is observed until a certain time at which either the event occurs or the subject leaves the study without having had the event. Thus, in the later case, it is only known to the researcher that the event occurred at some time after time point t_i (Cox, 1972).

One is then interested in the probability $P(t > t^*)$ that an event (depending on the context: death, relapse, failure etc.) has not occurred until time t^*. For the survival function $S(t^*) = P(t > t^*)$ it holds

$$S(t^*) = exp(-\Lambda(t^*)) = exp(-\int_0^{t^*} \lambda(u)du), \tag{1}$$

where $\lambda(u)$ is the hazard rate and $\Lambda(t^*)$ the cumulative hazard rate. A very common approach to model survival is the Cox proportional hazards model (Cox, 1972), where

$$\lambda(t^*, x_i) = \lambda_0(t^*)exp(x_i^T\beta), \tag{2}$$

with baseline hazard $\lambda_0(t^*)$. Plugging this in (1) one obtains

$$S(t^*) = exp(-\int_0^{t^*} \lambda_0(u)exp(x_i^T\beta)du) = exp(-\Lambda_0(t^*)exp(x_i^T\beta)), \tag{3}$$

where $\Lambda_0(t^*)$ is the cumulative baseline hazard. Respectively, this can be expressed as

$$S(t^*) = S_0(t^*)^{exp(x_i^T\beta)}, \tag{4}$$

as model for survival until time t^* with baseline survival $S_0(t^*) = exp(-\Lambda_0(t^*))$. The model for the hazard given in (2) consists of a factor that only depends on time (baseline hazard) and a factor only depending on the (individual) prognostic features. When using this for prediction, the baseline hazard respectively the baseline survival must be estimated. That is usually achieved via the *Breslow estimate*

$$\hat{\Lambda}_0(t^*) = \sum_{t_i \leq t^*} \frac{1}{\sum_{l \in R_i} exp(x_l^T\beta)}, \tag{5}$$

with R_i the risk set at time t_i.

In a medical context, the features used in the model usually comprise clinical data. But in the last two decades, data stemming from molecular procedures, such as micro-array gene expression data, has been gaining great attention and has been investigated extensively. Such data is now often routinely included as prognostic factors when survival should be analysed and predicted. Yet, it has been shown that their predictive benefit is limited compared to the optimistic initial findings, especially when molecular data is used solely. Instead, combining clinical and molecular data is promising (De Bin et al, 2014b). With the advancing technologies several different molecular data types are often available within one study. These types might include for example gene expression, copy number variation, proteomic, metabolomic, or methylation data (Boulesteix et al, 2017a), often denoted as omics data. For stringent notation, in this study the term *molecular* data is used for this kind of data. *Clinical* data refers to features easily accessible within a clinical context such as sex, age, performance scores, or features resulting from medical investigations such as blood levels. Finally, the term *multi-omics data* covers joined molecular and clinical data, regarding clinical data as one multi-omics group.

While the use of a single form of molecular data to produce prediction models with and without being combined with clinical data has been widely examined, the incorporation of multi-omics data (i.e. several molecular data types and clinical data) has yet not gained as much attention. With these data at hand it is the question whether and how to include the different types in a prediction model. We will first concentrate on the "whether" and take a closer look on the "how" later.

Although the naive approach to not distinguish the different data types, i.e. not giving emphasis on the group structure, is easily achievable (taking into account that the number of features most likely is much larger than the number of observations), several aspects speak for the use of the information lying in the group structure.

First of all, physicians and researchers with domain knowledge often have some kind of prior knowledge of which data type might be especially useful in the given context. If so, it is desirable to include such information through the incorporation of the group structure. This is strongly related to the fact that there are often established prognostic features which are known to be beneficial for building prediction models in a specific context. Most often this holds true for clinical features and it is of great interest to include these kinds of features by all means. But clinical data is usually low-dimensional, with often not more than 4 to 20 features. As molecular data is, in contrast, high-dimensional, usually with thousands or hundreds of thousands of features, the clinical features might get lost within the huge amount of molecular data when the group structure is not considered (De Bin, 2016). The same might be true for different kinds of molecular data. If, for example, in some context the copy number variation is more important than gene expression, it might be useful to incorporate that into the prediction model or to use methods which automatically include those data types that are of special interest.

These points indicate that taking the group structure of multi-omics data into account in some way or the other is potentially beneficial to the development of prediction models. Other important aspects are sparsity and transportability. As Klau et al (2018) point out, clinicians often prefer easily interpretable and applicable models including only few features that are of data groups they favor. Methods using the group structure and resulting in models which are sparse regarding the number of features as well as the number of used multi-omics types might be preferable from a practical perspective.

All this indicates that the incorporation of different data types might be beneficial when building survival prediction models. The scope of the study at hand is to compare different methods of building survival prediction models using this kind of data, i.e. multi-omics data. Special attention is given to the difference between naive methods, treating all features equally, and methods taking the group structure into account. While several methods have been proposed to build models by combining clinical data and one molecular data type, far less methods have been proposed to include several data types.

When comparing the methods, there a several ways to assess them. Some have been broached before, such as sparsity and transportability. Also, prediction accuracy is, of course, very important. Since it is not a trivial task to conduct a well-founded comparison of prediction methods, the next section describes how this can be achieved.

2.2 Assessing prediction methods

When assessing prediction methods, there are several aspects to be considered. First of all, it should be clear which properties of the methods should actually be assessed. In the context of prediction this might cover prediction performance and generalisability, to name but a few. Furthermore, if this has been clarified, the question arises of how to measure the considered properties and how to draw conclusions on whether the method is useful or not.

The later raises questions on a meta-level. How may a prediction method be judged useful? Usually this means a new method performs better than an already existing one, in some way or the other (Smith et al, 2013). At a lower level, the question is how the performance of predictive methods can be measured.

In this section, we will first describe the need of appropriate comparative studies in general. As multi-omics data is used, the following section describes how to capture the additional predictive value of molecular data. Finally, the theory of metrics for measuring predictive performance in survival time contexts is presented.

2.2.1 Design of benchmark experiments

As Smith et al (2013) point out, proposing a new prediction method should always be accompanied by the comparison of the new method with other, already established methods on several different data sets, which they register as a less well-established process as it should be. Boulesteix (2013, p. 2666) emphasises that the application of a new algorithm to at least two distinct data sets should be a “minimum nonnegotiable requirement for publication”. In addition, an issue is raised that points to the need for neutral comparison studies.

It is pointed out that, when carrying out a study on the proposal for a new method, there are several reasons speaking against the possibility that the researchers additionally provide a profound comparison with other methods themselves. Conducting a sound and ample comparison study is a difficult and time consuming process which researchers developing a new method will most likely not be able to additionally carry out. Besides that, it is not unlikely that the new method would be privileged if the researchers challenged their own method. Might be just for the sheer reason that they are experts on the new method, but do not have as much expertise on the competing methods.

Therefore, Boulesteix (2013) concludes that there is a need of neutral comparison studies. Such studies conduct a *representative* comparison opposed to studies conducting an *illustrative* comparison, both of which are legit, but within different scopes. The later are reported together with a new method, to highlight the possibilities of the new method and give a notion of its performance. To do so, they are carried out on few data sets and with few competing methods. Such studies should not draw conclusions about the superiority of one method over the other. Nevertheless, Boulesteix et al (2013) note a bias in favor of newly proposed classification methods. They claim that this also applies to subjects other than classification.

To answer the question of the superiority of one method over the other, neutral comparison studies should be conducted by researchers not having proposed one of the methods, using an adequate number of data sets and state-of-the-art competitor methods. This is necessary, because the performance is too variable across data sets to be adequately captured based on only a few data sets (Boulesteix et al, 2017b).

The study at hand is meant to carry out such a neutral comparison study or, to name another usual term, *benchmark experiment*, on several well established prediction methods for survival, based on high-dimensional multi-omics data.

As mentioned before, the number of data sets on which the comparison is based plays a crucial role in making a comparison study an ample experiment to draw conclusions about the superiority of one method over the other. Boulesteix et al (2017b) compare such studies to clinical trials, where the assessed new method equals, for example, a new treatment, and data sets play the role of patients. As with such clinical trials a reasonable number of observations (number of patients versus number of data sets) is required to draw conclusions that can be generalised.

While there have been a lot of studies proposing new methods (including illustrative comparisons), by far fewer neutral comparison studies based on a suitable amount of data sets have been published. As Boulesteix et al (2013) highlight, in the context of high-dimensional molecular data over a hundred articles have been published proposing new classification methods, most of them yielding comparisons based on only a few data sets. Neutral comparison studies, in contrast, are scarce. The study of Bøvelstad et al (2009) on combined clinical and molecular data may be treated as such, but again only few data sets are used. The same holds true for the study by Zhao et al (2014).

A neutral comparison that fulfils the above mentioned requirements is presented by Couronné et al (2018), but focuses on low-dimensional data. They conduct a large scale benchmark study to compare the very popular methods *logistic regression* and *random forest* for classification, based on 243 data sets. It is concluded that *random forest* yields significantly better performance. Likewise, Probst et al (2018) conduct a benchmark experiment on 39 data sets, but focus on tuning.

Finally, Lang et al (2015) present a study of automatic model selection in high-dimensional survival settings, using similar prediction methods as the study at hand. But again, only four data sets are used. To our knowledge there is no comparable benchmark experiment for survival prediction in the context of high-dimensional multi-omics data that uses an adequat amount of data sets.

The study at hand is meant to fill this gap. The goal is to provide a neutral comparison for several well-established prediction methods, based on 18 data sets. Compared to the aforementioned studies, the study at hand uses fewer data sets. This is due to fact that there are not as many easily available data sets yielding a reasonable multi-omics structure. Nevertheless, the number is high enough to draw proper conclusions (Boulesteix et al, 2015). The data sets will be further described in the *Data sets* section.

Based on that data, eleven prediction methods are compared in several dimensions. One of the most important dimensions is prediction accuracy or prediction performance. It has been emphasised by many, and is now considered as crucial, that an evaluation of the prediction performance should be based on an independent validation or test data set, which has not been used in any way to derive the prediction model (Boulesteix and Sauerbrei, 2011; Bøvelstad et al, 2009; De Bin et al, 2014a). Otherwise, the estimated prediction error, calculated based on the data used for model fitting (training data), will be over-optimistically biased.

Boulesteix and Sauerbrei (2011) point out that an external or temporal validation set should be used to make the conclusion generalizable to other than the population present for model fitting. While external means that the validation set stems from a different population, temporal means another sample of the same population gathered at a different point in time. Since such different data sets are often not available, splitting the data into a test and a training set (before a model is fit) is also appropriate.
Yet, splitting the data implies that there is a reasonable amount of observations. Also, in general, splitting the data in one training set and one test set is not fully sufficient to draw conclusions about the usefulness of a method. The performance might depend on the specific split, thus on a completely random aspect. Therefore, resampling strategies such as cross-validation (CV) should be used, to assess the performance on average over several test and training splits (Bischl et al, 2012).

In this section, it was outlined how to conduct a suitable comparison study in general. This is not restricted to specific data types or methods. An important aspect of building prediction models based on molecular and clinical data is discussed in the next section.

2.2.2 Added predictive value

Since we face multi-omics data, not only the probable implementation of a benchmark experiment plays an important role. In this setting, it is also important to assess whether the molecular data contribute any value to the predictive performance. As Boulesteix and Sauerbrei (2011) outline, there is a need to investigate this added predictive value of molecular data. Alongside to their and other findings, many of the proposed molecular features claimed to be of value for predicting disease outcomes, could eventually not be validated to outperform models using clinical data only (Boulesteix and

Sauerbrei, 2011; Bøvelstad et al, 2009; De Bin et al, 2014a). This indicates that the solitary use of molecular data does often not improve prediction performance compared to models using clinical data only. This leads to the question whether molecular data could be of any substantial use in the presence of clinical data, which are often standardly available. As has been mentioned before, several reasons speak for the inclusion of molecular data and a couple of findings show that the integrative use of clinical data and molecular data outperforms clinical models, revealing the potential of molecular data to add predictive value (Binder and Schumacher, 2008; De Bin et al, 2014b; Bøvelstad et al, 2009). But, as for example Bøvelstad et al (2009) point out, this strongly depends on whether the molecular data provides additional information or holds the same information as the clinical data. Similarly, De Bin et al (2014b) show, that, in the case of a breast cancer example, the inclusion of molecular data does not increase the prediction performance. This ambiguous findings make an evaluation of the added predicted value in comparison studies based on both clinical and molecular data necessary.
In the context of the study at hand this plays an important role, since the combined usage of clinical and molecular data is of major interest. Moreover, the molecular data may be further subdivided into different groups. Therefore, it is also of interest whether the consideration of the multi-omics group structure contributes to the predictive performance.

Boulesteix and Sauerbrei (2011) outline a framework to assess the predictive value, which was applied in several illustrative comparison studies (De Bin et al, 2014a; De Bin et al, 2014b). This framework comprises several strategies to receive combined prediction methods and approaches to validate the added predictive value. This includes a *naive*, a *residual* or *clinical offset* (De Bin et al, 2014b), a *favoring*, a *dimension reduction* and a *replace-*

ment strategy for combining models. We will only further discuss the first three strategies, since they are relevant for the methods used in this study. For the details of the other aspects, we refer the reader to the original study.
Within the naive strategy, all methods which do not distinguish between the different groups, thus do not take any group structure into account, can be subsumed. The major drawback is that the few clinical features might be lost within the huge amount of molecular features, leading to models not fully capable to use the information of the clinical features (an aspect, which might also apply for different sized molecular groups in a multi-omics setting).

On the contrary, the clinical offset strategy uses a fixed predefined clinical score, for example derived via Cox regression, as offset in a second-stage model including the molecular data. Therefore, the clinical features will not be penalised if any feature selection method is used.

Finally, the favoring strategy is an intermediate form of the first two strategies. Models are derived by favoring the clinical features in one way or the other, for example by posing different penalties on the two groups.

So far, this only considers the case where two kinds of data (clinical and molecular) can be distinguished. For the present study the molecular data is additionally grouped, leading to multi-omics data. Nevertheless, the rational behind the cited strategies also applies for multi-omics data. For example, several different penalties might be included to distinguish between the different multi-omics groups.

The study at hand discusses several methods and variants of them. This comprises methods of the *naive* strategy, methods taking the multi-omics groups into account individually, and methods using the clinical features as an offset. To differentiate the later two approaches, we use a slightly different terminology. If the clinical features are included as an offset and are not penalised, we speak of a method *favoring clinical features* (over molecular features), in contrast to the second strategy mentioned above. Methods incorporating all multi-omics groups individually are denoted as methods *using the (multi-omics) group structure.* As De Bin et al (2014a) point out, incorporating this additional group information is a relatively new field and only a few methods have been proposed.

Finally, to assess the added predictive value of molecular data, models using only clinical data and combined models using multi-omics data should be compared. When doing so, it is important that the combined model is not derived via the naive strategy (Boulesteix and Sauerbrei, 2011).

Summarising, in this section we first discussed the concept of added predictive value of molecular data. We then presented the framework to assess the added predictive value of molecular data based on Boulesteix and Sauerbrei (2011) and transferred it to the context of the study at hand.

As both, the concept of neutral benchmark experiments as well as the concept of the added predictive value (within a neutral as well as within an illustrative comparison), strongly rely on the comparison of the prediction performance, we discuss measures to evaluate the prediction performance in survival analysis in the following.

2.2.3 Performance measures

In general, there are two major prediction performance properties that may be assessed: calibration and discrimination. Calibration measures the accordance of the predicted and the true outcome value, whereas discrimination measures the ability to tell observations apart according to the outcome (Steyerberg et al, 2010). For survival time prediction, discrimination means how well a method predicts the right order of survival times (De Bin et al, 2014b).

Since the outcome of interest in the present study is survival time, we discuss measures fit to assess the performance of survival time prediction methods. Of major interest is the probability $P(t_i > t^*|x_i)$ that an individual survives until a certain time t^*, given the respective prognostic features. The aim of using a specific prediction method $\mathcal{M}$ in the time-to-event context is to estimate these probabilities (Graf et al, 1999). Let $\mathcal{D}$, $|\mathcal{D}| = n_T$, be the set of observations used for testing. Let further $\hat{\pi}^{\mathcal{M}}(t^*|x_i)$ denote the estimate for $P(t_i > t^*|x_i)$ obtained by using method $\mathcal{M}$ on the training data $D \setminus \mathcal{D}$, $|D \setminus \mathcal{D}| =$ n. To assess whether the predicted probability $\hat{\pi}^{\mathcal{M}}(t^*|x_i)$ is a good estimate, computing the squared error $(I(t_i > t^*) - \hat{\pi}^{\mathcal{M}}(t^*|x_i))^2$, where I refers to the indicator function yielding 1 if $t_i > t^*$ and 0 otherwise, is a suitable approach. In general, to assess the method $\mathcal{M}$, the empirical mean squared error

$$\hat{BS}(t^*) = n_T^{-1} \sum_{i=1}^{n_T} (I(t_i > t^*) - \hat{\pi}^{\mathcal{M}}(t^*|x_i))^2 \tag{6}$$

as a surrogate for the expected mean squared error is used and known as *Brier-score.* Here n_T refers to the observations used for testing. The Brier-score is an overall assessment measure for one specific time point, taking into account discrimination and calibration alike. In the case of censoring, the Brier-score must be adjusted to account for the information loss by inserting individual weights. The lower the Brier-score, the better the prediction performance. A value of 0.25 corresponds to a prediction without taking any information into account (Graf et al, 1999).

In comparison, the *concordance statistic* or *c-index* is a measure to assess discrimination. Gerds et al (2013) define the simple c-index as

$$\hat{C}(t) = \frac{\frac{1}{n_T^2} \sum_{i=1}^{n_T} \sum_{k=1}^{n_T} I[\hat{\pi}^{\mathcal{M}}(t^*|x_i) > \hat{\pi}^{\mathcal{M}}(t^*|x_k)] I[t_i < t_k] \mathcal{N}_i(t^*)}{\frac{1}{n_T^2} \sum_{i=1}^{n_T} \sum_{k=1}^{n_T} I[t_i < t_k] \mathcal{N}_i(t^*)}. \tag{7}$$

Again I is the indicator function and $\mathcal{N}_i(t^*) = I[t_i \leq t^*, \delta_i = 1]$. It describes the ratio of concordant pairs among all concordant or discordant pairs.

Through $\mathcal{N}_i(t^*)$ only those pairs are regarded, for which the observation with the shorter survival time is not censored. Thus, only those pairs are considered for which the concordant/discordant status is definitely known. This censoring bias can be avoided using *inverse probability-of-censoring weighting* (IPCW) for the estimation of the c-index, although this comes at the price of further modelling the conditional survival function (Gerds et al, 2013). Another version of the c-index is based on Uno et al (2011) and is a special case of IPCW-based concordance statistics. It does not depend on a correctly specified survival model and is thus usually favored (Gerds et al, 2013). We use this measure, assuming that the censoring does not depend on the features.

One can think of several other measures to assess the prediction performance of survival models (Steyerberg et al, 2010). Still the most common ones are the *Brier-score* and the *c-index* (Gerds et al, 2013; Steyerberg et al, 2010).

Taking the Brier-score as a function of time, one can compute prediction error curves. Plotting these curves makes it possible to visually inspect and compare the prediction performance of different models. Usually, the performance is followed until a specific point in time (Gerds et al, 2008). Also, the *integrated Brier-score*

$$I\hat{B}S(t_0) = \int_0^{t_0} [n_T^{-1} \sum_{i=1}^{n_T} (I(t_i > t^*) - \hat{\pi}_i^{\mathcal{M}}(t^*|x_i))^2] dW(t^*), \tag{8}$$

where $W(t)$ is a weighting function, is a measure not based on a single time point (Graf et al, 1999). Instead, all time points until t_0 will be taken into account. Hence, the IBS yields a single overall performance value for every model under investigation, similar to the c-index.

To assess the prediction performance in the benchmark study at hand we use the IBS and the c-index based on Uno et al (2011). We describe the methods used in this benchmark study in the next section.

2.3 Overview of methods

Now, appropriate prediction methods for multi-omics data are described from a theoretical point of view. Several approaches from the fields of machine learning and statistics seem reasonable. Nevertheless, the focus lies on two main classes of prediction methods from the field of statistics and one approach from the field of machine learning. As the methods used in this study can be grouped into these three general approaches, this section gives a brief, high-level overview. In the following sections the single methods will be described in detail.

Since all of the methods to be considered must at least be able to perform feature selection in a high-dimensional setting where the number of features p exceeds the number of (training) observations n by huge extend, there is the need of more sophisticated (statistical) methods. Techniques like generalised multivariate or Cox regression fail to work in that setting.

For this problem two approaches have been emerging in the statistical community over the last two decades. One of these are regularisation methods based on the *Lasso* (Tibshirani, 1996), the other is *statistical boosting* (Mayr et al, 2014). Furthermore, *random forest*, a method from the field of machine learning introduced by Breiman (2001), is a promising approach.

Of course other methods and approaches are conceivable. See for example Zhao et al (2014), Sutton et al (2018), Wiel et al (2015) and Hong et al (2018), the later two with a slightly different scope.

The *Lasso*, first described by Tibshirani (1996), is one of the most widely used methods to conduct regression in a high-dimensional setting. The method penalises large coefficient values and leads to sparse final models by setting a substantial amount of coefficients to zero, if the penalty parameter is large enough. Several specifications and variants have been proposed to meet specific problems, such as time-to-event regression (Tibshirani, 1997), some of them with focus on multi-omics data (e.g. Klau et al, 2018; Boulesteix et al, 2017a).

Instead, *boosting* is a technique introduced in the context of classification in the machine learning community, which has then been transferred to statistical contexts and gained a lot of attention there (Mayr et al, 2014). As Friedman et al (2000) showed, boosting fits additive models in a stage-wise manner, yet yielding sparse models by early stopping. Not being prone to overfitting, is one of several strengths (Hastie et al, 2009).

In contrast, *random forest* is a method from the field of machine learning introduced by Breiman (2001). It is an ensemble method based on classification and regression trees, using bootstrap aggregation (bagging) to obtain a result based on the tree committee. It yields great prediction performance on the one hand, but can be regarded as a black box method, which does not yield easily interpretable models, on the other hand (Hastie et al, 2009).

We focus on the Lasso methods, statistical boosting methods, and random forest since they are widely used and studied, have shown to offer good prediction performance in many settings, and can more and more be regarded as standard methods (Hastie et al, 2009). Also, based on their theoretical properties, they are promising candidates to deal with multi-omics data. Last but not least, solid implementations exist for the statistical programming language R.

2.4 Statistical boosting

Boosting was originally developed in the machine learning community for the sake of the theoretical question, whether weak classification methods can be revised to better ones. The main idea of boosting is to iteratively fit a series of weak models, thereby forcing the algorithm to concentrate on observations that are hard to predict and update the estimates accordingly in every step. Friedman et al (2000) transferred this to terms of a statistical framework. They showed that the most powerful boosting algorithm at that time, AdaBoost, fits an additive logistic regression model in a stage-wise manner by means of the exponential loss function. Generalising the problem as a gradient descent in function space, Friedman (2001) paved the way for statistical boosting, a very powerful and widely used tool for statistical modelling.

A comprehensive overview of the evolution of boosting is given by Mayr et al (2014). For more recent developments in the field of statistical boosting see Mayr et al (2017).

2.4.1 Introduction to statistical boosting

Statistical boosting can be seen as a form of iterative function estimation. In its initial form it was derived as a gradient descent in function space (Friedman, 2001). On this basis two different frameworks were developed. The approach copiously described by Bühlmann and Hothorn (2007) is strongly based on the basic notion and therefore usually referred to as *gradient boosting.* Tutz and Binder (2006) introduced a different approach, known as *likelihood-based boosting.* Still, they may both be regarded in the notion of a gradient descent in function space (De Bin, 2016). We first describe the basic conception of statistical boosting as gradient descent and then highlight the two specific manifestations used in this study.

In general, one is interested in a function f that minimises the expected loss when used to model the data. Given a target variable Y and a matrix of independent variables $\mathcal{X}$, this can be expressed as

$$f^*(.) = \underset{f(.)}{\arg\min}\, \mathbf{E}[\rho(Y, f(\mathcal{X}))], \tag{9}$$

with ρ a loss function (Bühlmann and Hothorn, 2007). Having realisations y of Y and $\mathbf{X}$ of $\mathcal{X}$, estimating $f^*(.)$ based on boosting relies on three core concepts: *(1)* gradient descend with respect to the loss function ρ; *(2)* using a base learner $\mathcal{G}$ to model the iterative updates; *(3)* steering the learning by a learning rate ν. The following algorithm is based on De Bin (2016, p. 516) and Bühlmann and Hothorn (2007, p. 480) and describes boosting in its general form. After initiating $\hat{f}^{[0]}(\mathbf{X})$ by a constant

1. the negative gradient of the loss ρ is computed with respect to f

$$u^{[m]} = -\frac{\delta}{\delta f}\rho(y, f)|_{f=\hat{f}^{[m-1]}(\mathbf{X})} \tag{10}$$

2. fitting the base learner $\mathcal{G}$ to $u^{[m]}$ leads to the update $\hat{\mathcal{G}}^{[m]}(u^{[m]}, \mathbf{X})$

3. penalising $\hat{\mathcal{G}}^{[m]}$ leads to the estimate $\hat{f}^{[m]}(\mathbf{X}) = \hat{f}^{[m-1]}(\mathbf{X}) + \nu\hat{\mathcal{G}}^{[m]}(u^{[m]}, \mathbf{X})$

As the gradient is pointing to the direction of the steepest ascent, in every iteration computing the negative gradient in (10) leads to an approach of the minimum of the loss function in the direction of the steepest descent. To perform the algorithm it is necessary to predefine the number of boosting steps m_{stop}. Repeating m_{stop} times steps 1. to 3. leads to the final estimate $\hat{f}^{*}(\mathbf{X})$. Therefore, as Hastie et al (2009, p. 341) emphasise, "boosting fits an additive model"

$$\hat{f}^{*}(\mathbf{X}) = \sum_{m=0}^{m_{stop}} \hat{f}^{[m]}(\mathbf{X}) = \hat{f}^{[0]}(\mathbf{X}) + \nu \sum_{m=1}^{m_{stop}} \hat{\mathcal{G}}^{[m]}(u^{[m]}, \mathbf{X}). \tag{11}$$

The number of boosting steps m_{stop} and also the learning rate ν, $0 < \nu < 1$, are hyper-parameters, which have to be set in advance. The learning rate ν steers how much every update contributes to the final boosting estimate. It turned out that slow learning, when ν is set to a small value, leads to a better prediction performance. Bühlmann and Hothorn (2007, p. 480) state that, in contrast to the number of boosting iterations, it is of "minor importance" as long as it is set to a small value. They suggest $\nu = 0.1$. We take a closer look on m_{stop} in the *Early stopping* section.

The described procedure is a general approach. Considering specific loss functions and base learners leads to different boosting algorithms. The base learners may be chosen as desired. For example, Bühlmann and Hothorn (2007) mention smoothing splines. Often *gradient boosting* is associated with tree stumps as base learners (Hastie et al, 2009). We focus on a different version: First of all, only one feature is updated per iteration, known as *component-wise* boosting (Bühlmann and Hothorn, 2007; Hofner et al, 2014). This is specifically useful in the context of multi-omics prediction. Furthermore, using component-wise boosting along with univariate linear models as base learners, leads to *model-based boosting* (Hofner et al, 2014).

2.4.2 Model-based boosting

This approach strongly adapts the general boosting algorithm (Hofner et al, 2014). As stated above, it uses univariate linear models as base learners. For the resulting component-wise boosting algorithm, in every iteration all of the features are individually regarded, but only that one gets updated, that reduces the loss the most. Thus, in step 2 of the general algorithm, p possible updates are computed and an additional step is introduced where the loss minimising feature is chosen. It then gets its coefficient updated.

This satisfies two very important requirements: First of all, it leads to a integrated feature selection property that allows to use data where the number of features exceeds the number of observations. Secondly, as it is a sum of linear models, the resulting model is interpretable.

At that, the strength of the feature selection depends on the number of boosting iterations. Also the learning rate ν acts like a shrinkage parameter as the coefficients get updated only slightly and, concerning the feature selection property, interacts with the number of boosting iterations.

Having censored survival times as target variable, the loss is set to the negative partial log-likelihood. So in step 2 the gradient is computed as

$$\hat{u}_i^{[m]} = \delta_i - \sum_{l \in R_i} \delta_l \frac{exp(x_l^T \hat{\beta}^{[m-1]})}{\sum_{k \in R_l} exp(x_k^T \hat{\beta}^{[m-1]})}, \tag{12}$$

with R_i the risk set at time t_i. Let $x_{\cdot j}$ be the jth column of the feature matrix $\mathbf{X}$ (in contrast to x_i, the ith row). Setting the base learners as univariate linear models leads to the possible updates $\hat{\beta}_j^{[m]} = (x_{\cdot j}^T x_{\cdot j})^{-1} x_{\cdot j}^T \hat{u}^{[m]}$, and the feature minimising the squared error loss is chosen to be updated (De Bin, 2016).

2.4.3 Likelihood-based boosting

In contrast, likelihood-based boosting uses a penalised version of the partial log-likelihood as a loss function,

$$pl_{pen}(\beta) = pl(\beta) - 0.5\lambda\beta^T \mathbf{P}\beta, \tag{13}$$

where $\mathbf{P}$ is a $p \times p$ matrix, typically the identity matrix (De Bin, 2016). In comparison to model-based boosting, there is no fixed learning rate ν. Instead, the shrinkage is applied via the penalty parameter λ, offering more flexibility. Also, the penalty λ is directly applied in the coefficient estimation step, whereas ν is applied on the selected updates.

As it is still an iterative procedure, the updates of previous iterations have to be included in expression (13), to make use of the information gained. This is achieved by including an offset term. The term $\hat{\eta}_i^{[m-1]} = x_i^T \hat{\beta}^{[m-1]}$ holds the information gained in the previous iterations and is added as an offset, leading to the expression

$$pl_{pen}(\beta | \hat{\beta}^{[m-1]}) = \sum_{i=1}^{n} \delta_i [\hat{\eta}_i^{[m-1]} + x_i^T \beta - log(\sum_{l \in R_i} exp(\hat{\eta}_l^{[m-1]} + x_l^T \beta))] - 0.5\lambda\beta^T \mathbf{P}\beta. \tag{14}$$

In likelihood-based boosting this function is maximised in every boosting iteration to obtain the iterative updates (De Bin, 2016; Binder and Schumacher, 2008).
Since the information of previous iterations is included as an offset and a penalty is applied on the coefficients, it is easily possible to define features that must be included in the model mandatorily. This can be achieved by setting the corresponding diagonal elements of the matrix $\mathbf{P}$ to zero, an approach suited to include group structure information and to assess the additional predictive value provided by the penalised optional features. Often clinical features are set to be mandatory and the molecular data used as potentially, but not necessarily, additive information (Binder and Schumacher, 2008).

2.4.4 Early stopping

As has been pointed out before, the number of boosting iterations is the main tuning parameter for boosting. Although boosting is in general not very prone to overfit, respectively its overfitting behaviour is slow, it still eventually overfits if it is allowed to iterate until the convergence of the loss function (Bühlmann and Hothorn, 2007). So early stopping, i.e. stopping before the convergence of the loss function, is necessary. This not only prevents overfitting, but also serves as a feature selection mechanism. The choice of the number of boosting iterations strongly effects the strength of the feature selection and the whole procedure. As Seibold et al (2018) emphasise, choosing it too small may lead to models not including relevant features, choosing it too large may lead to models with irrelevant features. So, usually this parameter is chosen by means of cross-validation. An important aspect in doing so is the fact that simple cross-validation does not lead to optimal solutions. Hence, using repeated CV is recommended, which leads to better results, i.e. more stable values, of m_{stop} (Seibold et al, 2018).

Boosting gains its feature selection property through early stopping and in combination with a component-wise approach. As has been described earlier, component-wise boosting fits several univariate linear models, each for every feature in the data, and updates the coefficient of that feature, which minimises the loss. If $p > m_{stop}$, this automatically means that not all features can be included in the model. Only those leading to the best prediction results will be used. If $p > n$ non-component-wise boosting is not applicable at all (De Bin, 2016). Also, as usually slow learning is adapted, only a penalised value of the estimated coefficient is added as an update. If a feature leads to the best prediction in several steps, it gets its coefficient updated several times. Thus ν interacts with m_{stop}, and p does not necessarily need to be greater than m_{stop} to gain the feature selection property (De Bin, 2016).

2.4.5 Differences of the two approaches

In general, the two boosting approaches are very different. Not only do they use two different ways of computing the iterative updates, but also the penalisation is applied differently. In model-based boosting the estimation is based on a regression to the negative gradient, where the base learners are fit to the negative gradient, resulting in a regression on the (pseudo-)residuals of the previous step. In that way the algorithm concentrates on observations that are hard to predict (Mayr et al, 2017; Hastie et al, 2009).

By contrast, in likelihood-based boosting likelihood maximisation is used. The iterative updates are obtained based on the maximisation of the base learners' likelihood, thus by directly estimating the base learners. At that, the information gained in previous steps is incorporated by an offset term. Moreover, in likelihood-based boosting the penalty term is applied to the likelihood directly and the parameter λ (in (13) and (14)) steers the penalisation. Instead, in model-based boosting the updates get penalised via the learning rate ν (Mayr et al, 2017).

It should be mentioned that for component-wise boosting the differences are less striking and even coincide in the standard continuous linear regression context, if the inner loss is set to be the L_2-loss function for gradient boosting and the normal likelihood is chosen for likelihood-based boosting (De Bin, 2016).
One advantage of likelihood-based boosting over model-based/gradient boosting - important in the context of multi-omics prediction - is the possibility to naturally include mandatory features by applying the penalty term only to the coefficients of the features that should be penalised (Binder and Schumacher, 2008). Thus, it is possible to easily include some kind of group structure. Normally the clinical features are set as mandatory and molecular features get penalised (even if that means that different molecular features are not differentiated). Indeed, the inclusion of mandatory features is also possible in model-based boosting, but not as inherently as in likelihood-based boosting (De Bin, 2016).

Comprising, statistical boosting can be seen as one major approach of fitting prediction models with the ability to be used in high-dimensional settings, to include some kind of group structure information, and yielding sparse and interpretable models. Based on a general framework, two specific boosting strategies have been introduced, namely model-based boosting and likelihood-based boosting. In the next section, the Lasso is presented as a second general approach and specific Lasso-based methods are described.

2.5 Lasso methods

The *Lasso*, first introduced by Tibshirani (1996), is a regression technique combining the strengths of *subset selection* and *Ridge regression*, leading to sparse and stable models. Due to the sparse nature of the resulting models, the Lasso has two required properties especially relevant in high-dimensional multi-omics contexts. First of all, since many of the estimated regression coefficients will be zero, the models are interpretable. As outlined, a property often required in clinical contexts. Besides that, its inherent shrinkage and selection properties make the Lasso a predestined technique to tackle high-dimensional problems, where the number of features exceeds the number of observations.

2.5.1 Standard Lasso

The standard Lasso, in the following also simply called *Lasso*, gains its shrinkage properties through penalising the regression coefficients. In general, the estimate in the Lagrangian form is given by

$$\hat{\beta}^{lasso} = \arg\min_{\beta}\{\frac{1}{2}\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda\sum_{j=1}^{p}|\beta_j|\} \tag{15}$$

In contrast to Ridge regression, which uses a quadratic penalty, the coefficients get penalised by a L_1-penalty leading to a non-differentiable problem. As a consequence, many of the coefficients will be set to exactly zero, thus leading to the sparsity property. Due to the non-differentiable nature there is no closed form for the regression estimates (Hastie et al, 2009). Originally proposed as an enhancement for standard linear regression, the method is based on ordinary least squares. But the method has been made available for other regression contexts such as Cox regression (Tibshirani, 1997). In the later case, the estimation is based on maximising the log partial likelihood. As has been outlined above, the Cox model assumes $\lambda(t^*|x_i) = \lambda_0(t^*)exp(x_i^T\beta)$ leading to the partial likelihood

$$L(\beta) = \prod_{r\in E}\frac{exp(x_{i_r}^T\beta)}{\{\sum_{l\in R_r} exp(x_l^T\beta)\}}, \tag{16}$$

where E is the set of event indices, R_r the set of indices of individuals at risk and i_r indicates failure at time t_r (Tibshirani, 1997). The Lasso estimate is then given through (Simon et al, 2011)

$$\hat{\beta} = \arg\max_{\beta}\{log(L(\beta)) - \lambda \sum_{j=1}^{p} |\beta|\}. \tag{17}$$

The parameter λ steers the amount of penalisation, leading to the standard regression estimate if set to zero, and to increasingly spares models with increasing values. The parameter λ is important and usually determined through cross-validation (Simon et al, 2011).

Despite its excellent properties, the standard Lasso still suffers from some drawbacks. For example, if several features are strongly correlated, Lasso selects only one of them (Simon et al, 2011). Furthermore, it is not possible to incorporate any additional group structure information. This means that all features will be equally penalised regardless of their priorly assumed importance. In multi-omics settings it is often assumed that some of the groups are more important than others, especially regarding clinical features. Thus, for example, the small amount of clinical features might get lost in the huge amount of molecular features (Boulesteix and Sauerbrei, 2011).

Several adaptations of the Lasso method have been proposed to overcome this drawbacks, some of them being described in the next sections. We focus on Lasso variants that are suited to incorporate group structure information in a multi-omics setting. Besides these methods, there are other extensions tackling specific weak points of the standard Lasso. For example, the *elastic net* method combines Ridge and Lasso regression to overcome the problem of correlated features (Simon et al, 2011), and Zou (2006) proposed the *adaptive Lasso*, a consistent version of Lasso, yielding oracle properties. Furthermore, Yuan and Lin (2006) introduced the *group Lasso*, an enhancement for factor selection.

2.5.2 IPF-Lasso

IPF-Lasso is a method introduced by Boulesteix et al (2017a) specifically designed to incorporate (multi-)omics group structure information. It is an extension of the standard Lasso using different penalties for the specified groups according to their relevance. As with standard Lasso, the method can also be applied to other regression outcomes particularly survival time data.

Let there be G groups, then $x_{\cdot 1}^{(g)}, ..., x_{\cdot p_g}^{(g)}$ denote the features of group g (g $= 1, ..., G$), with p_g the number of features within group g. Furthermore $\beta_j^{(g)}$ indicates the corresponding coefficient of feature $x_{\cdot j}^{(g)}$ (Boulesteix et al, 2017a). To estimate the coefficients one has to minimise the expression

$$\sum_{i=1}^{n}(y_i - \sum_{g=1}^{G}\sum_{j=1}^{p_g} x_{ij}^{(g)}\beta_j^{(g)})^2 + \sum_{g=1}^{G}\lambda_g\|\beta^{(g)}\|_1, \tag{18}$$

with $\|.\|_1$ the L_1-Norm, $\lambda_g > 0$ the penalty of group g and $\beta^{(g)} = (\beta_1^{(g)}, ..., \beta_{p_g}^{(g)})^T$. Boulesteix et al (2017a) set the first group as reference. This leads to a penalty factor of 1 and the penalty λ_1. The other penalty factors follow as λ_g/λ_1, leading to a vector of penalty factors (1, λ_2/λ_1, ... , λ_G/λ_1). Applying this on the penalty term, the groups get penalised individually and according to the relevance of the included features. This leads to strong shrinkage and sparsity within a group when it is of low importance or only few features of that group are of relevance, while at the same time relevant features from other groups are preserved.

Similar to the single penalty parameter λ in the standard Lasso context, this vector must be set in advance, thus yielding hyper-parameter characteristics. As such, it is recommended to specify this vector via cross-validation (Boulesteix et al, 2017a).

Unfortunately, this also leads to the major drawback of the method. If there are more than three to four feature groups, the cross-validation will get computational infeasible, reducing the relevance of the method for multi-omics prediction problems. To overcome this problem, Schulze (2017) introduced the *two-step IPF-Lasso* (TS IPF-Lasso). In the first step, a single candidate vector for the penalty factors is determined. This is achieved by fitting a regression model on the data and computing the mean of the coefficients of each group. By inverting the means one obtains a single set of data-driven penalty factors, which can than be used in the second step for the IPF-Lasso method as described above. This reduces the extensive computation broadly, making it possible to use IPF-Lasso in situations with more than a few feature groups. Several strategies for the first step (using different regression techniques and means) may be used. We describe the strategy used for the study at hand later in detail.

2.5.3 Priority-Lasso

Another method designed for the incorporation of different feature groups is *priority-Lasso* (Klau et al, 2018). The principle idea is based on the fact that often clinical researchers and physicians have some prior knowledge concerning the importance of different feature groups. For example, they might know that copy number variation is more important for predicting the survival time of a specific cancer type than gene expression information. To include such knowledge, a priority order for the groups of interest is defined by the researchers. Priority-Lasso then successively fits regression models using the features in the order of their group's priority, until all groups have been considered. The resulting linear predictor of every step is used as an offset for the regression model fit to the features of the group with the next highest priority. Thus, the features of a group with lower priority only explain that part of variation that has not been explained by features of higher priority. In a standard linear regression context this means fitting the residuals of the preceding step (Klau et al, 2018).

Formally speaking, let G again be the number of groups under investigation. Let further $\pi = (\pi_1, ..., \pi_G)$ be a permutation of $(1, ..., G)$ indicating the priority order. As for IPF-Lasso $\beta_j^{(\pi_g)}$ indicates the coefficient of feature j of group π_g and p_{π_g} the number of features from group π_g. The coefficients of the first step are then estimated by applying standard Lasso on the features of the group with highest priority order. Thus, minimising

$$\sum_{i=1}^{n}(y_i - \sum_{j=1}^{p_{\pi_1}} x_{ij}^{(\pi_1)}\beta_j^{(\pi_1)})^2 + \lambda^{(\pi_1)}\sum_{j=1}^{p_{\pi_1}}|\beta_j^{(\pi_1)}| \tag{19}$$

leads to the linear predictor $\hat{\eta}_{1,i}(\pi) = \hat{\beta}_1^{(\pi_1)}x_{i1}^{(\pi_1)} + ... + \hat{\beta}_{p_{\pi_1}}^{(\pi_1)}x_{ip_{\pi_1}}^{(\pi_1)}$. This is used as an offset for the Lasso model fit to the next group in the following step (Klau et al, 2018). This procedure is iterated until all groups have been considered, using different offsets $\hat{\eta}_{\pi_g,i}(\pi)$ in each step.

Klau et al (2018) emphasise that the information used to produce the model of a subsequent step has already been used to compute the offsets of the previous steps. Therefore, they recommend to use cross-validation to estimate the offsets. Otherwise, variability that could be explained by lower-priority groups might be removed, although it is not explained by previous groups.

For the study at hand an issue arises concerning the prior group importance. As we do not have any substantial knowledge concerning the regarded cancer types, we cannot specify a priority order as described above. To circumvent this problem, we altered the method to a two-step procedure similar to the two-step IPF-Lasso. In the first step, we determine a vector of coefficient means for every group exactly as in the first step of the two-step IPF-Lasso. Inverting the means and ordering them increasingly results in a vector where the first element corresponds to the most important group and so on. We use this ordering as a surrogate for a knowledge based priority order. It has to be emphasised that this is only used due to the lack of substantial prior information (which should be favored if available). Klau et al (2018), in contrast, describe a cross-validation-based method to specify the priority order in the absence of prior knowledge. But as with IPF-Lasso, this has shown to be computational infeasible with the data at hand.

2.5.4 Sparse Group Lasso

The *group Lasso* mentioned above could be another method using group information. Since it is mainly designed for factor selection, it either selects all features of a group or none of them. This leads to a sparsity on group level but not within groups. That is useful for factor features, but does not equally apply to multi-omics settings, since often only few features of a group are relevant (Boulesteix et al, 2017a). Therefore, it is of less use in multi-omics settings. Simon et al (2013a) introduced the *sparse group Lasso* (SGL). It builds on the group Lasso, but additionally offers sparsity within groups, making it a very interesting candidate for the multi-omics benchmark study. SGL has already shown to be competitive in such settings (Simon et al, 2013a; Boulesteix et al, 2017a).

On the other hand, Schulze (2017) mentions that the method encounters substantial problems in the presence of very high-dimensional data. The method leads to a fatal error in the standard statistical software R, where it is made available via an add-on package (Simon et al, 2013b). Having tried many approaches to circumvent this problem, Schulze (2017) eventually drops the method from the study as no attempt is successful. Unfortunately, we encountered the same problem with data sets of around 100,000 features. Therefore, it was decided not to regard SGL neither, although this means dropping one very promising candidate method.

Concluding, three methods are used within the benchmark experiment which we subsume as Lasso-based methods. This comprises the standard Lasso, the two-step IPF-Lasso (TS IPF-Lasso) and the two-step priority-Lasso (TS priority-Lasso). The standard Lasso is not able to include group specific information and treats all features within a data set equally. The newer methods TS IPF-Lasso and TS priority-Lasso are based on the standard Lasso and additionally incorporate group structure in different ways.

So far, statistical boosting and Lasso-based methods have been described as candidates for multi-omics prediction methods. In the next section, we finally describe random forest as a third method.

2.6 Random forest

Random forest is a tree-based ensemble method introduced by Breiman (2001). Instead of growing a single classification or regression tree, it uses bootstrap aggregation (bagging) to grow several trees and average the results as outcome. Bagging means that out of the data set many bootstrap samples are drawn and on each of these sub-samples a tree is fit. This reduces the variance of the single tree results.

To additionally reduce the variance, only a fraction of the features is randomly drawn and used to build a single tree, resulting in decorrelated trees (Hastie et al, 2009). The following algorithm is based on Hastie et al (2009, p. 588).

To compute a random forest

```
1. For $b = 1$ to $B$:
    1.1 Draw a bootstrap sample of size $n$ from the training data
    1.2 Fit a single tree $T_b$ to the bootstrap sample by recursively repeating the following steps for each terminal node, until the predefined minimum node size
        a) Draw $mtry$ numbers of features randomly
        b) Choose the best feature and split-point combination
        c) Split the node into two daughter nodes
2. Output the tree ensemble
```

Predictions for new data can then be made for regression via $\hat{f}_{rf}^{B}(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$. For a classification problem the class of a new observation is chosen by majority vote over all B trees.

The main hyper-parameter of random forest is the number of features to be randomly chosen, often denoted as *mtry* (Couronné et al, 2018). Although specifically tuned values for a data set at hand may result in better performance, some standard default values have been established: *mtry* is standardly set to $\lfloor\sqrt{p}\rfloor$ and $\lfloor\frac{p}{3}\rfloor$ for classification and regression respectively (Hastie et al, 2009; Couronné et al, 2018). According to Couronné et al (2018), the number of trees to be fit, *ntree*, should be chosen as large as possible and not be treated as a hyper-parameter. They state that a reasonable number is about a few hundred trees.

Moreover, the tree depth is important. It is associated with the minimum

number of observations a terminal node should include, often denoted as *nodesize*. A larger value for *nodesize* leads to less deep trees, with just a few terminal nodes (Couronné et al, 2018).
The random forest method was expanded to survival time data by Ishwaran et al (2008). The feature maximising the difference in survival is chosen as best feature. The terminal *nodesize* is defined by the minimum number of deaths that should be included. Eventually, the cumulative hazard function is computed via the Nelson-Aalen estimator for every tree and averaged over all fitted trees to obtain the ensemble cumulative hazard function. Predicting a new observation means dropping it down the trees. According to its feature values it is passed through the nodes. Finally, according to the terminal nodes it reaches, the average of the Nelson-Aalen estimates (over all trees) is used for the cumulative hazard.

The random forest method is very competitive regarding prediction performance and does not need a lot of tuning (Hastie et al, 2009). In their large benchmark study, Couronné et al (2018) even show that random forest with default parameters outperforms logistic regression. So, it is not surprising that random forest gained a lot attention in the last years. Besides the good prediction performance, it offers further often desired properties. It naturally handles high-dimensional data, data of different types and missing data. Particularly, the ability to deal with high-dimensional data, where p is greater than n, is an important property for the study at hand.
Nevertheless, there is a major drawback. Random forest can be regarded as a black-box method yielding models that are hard to interpret. Although there are some approaches to gain insight into the resulting models, such as feature importance measures and partial dependence plots, random forest models are not as easily interpretable and transportable as models yielding coefficient estimates of few relevant features (such as the boosting and Lasso variants

described before) (Couronné et al, 2018). Furthermore, at the moment there is no adaptation to make use of group structure information. Despite this drawbacks, random forest is still an important candidate method for the prediction of multi-omics data.

Summarising, its strong prediction performance and ease of use have made random forest a widely used and competitive method that should not be neglected in a comparison study of prediction methods for survival data. Overall, there a three major prediction approaches regarded in the study at hand: random forest, boosting and Lasso. For the later two, different variants and adaptions exist. These lead to different possible Lasso-based methods and versions of boosting. The precise implementations of these methods and the data sets they are applied on are described in the next section.

3 Benchmark experiment

In this section, we outline the benchmark experiment conducted for this study in detail. This comprises the overall framework, the data sets, the specific method configurations, the dimensions the methods are assessed in and the software used. In the following, the specific model configurations will also be called *learners* to separate the specific configurations form the general *methods* described in the *Background* section.

3.1 Setup and *mlr* framework

The benchmark experiment is conducted using R 3.3.4 (R Core Team, 2018). To reproduce the results, all the code and data can be found in the electronic appendix. To further improve the reproducibility, the add-on package *checkpoint* (Microsoft Corporation, 2018) is used. Because the computations are extremely time demanding, but parallelisable, the package *batchtools* (Lang et al, 2017) is used for parallelisation.
To implement the benchmark experiment the package *mlr* (Bischl et al, 2016) is used. Since several learners have to be customized, we use a development version of *mlr*, which can be found in the electronic appendix. *mlr* offers a simple framework to conduct all necessary aspects in a unified way. The methods under investigation can be accessed via wrappers, different performance measures can be applied, and resampling for hyper-parameter tuning and performance assessment may be defined. Moreover, parallelisation is easily achieved.

In the benchmark study we apply 11 learners on 18 data sets. To assess their performance, we use a repeated cross-validation strategy and performance measures in three performance dimensions.
Several, but not all, of the methods under investigation can be accessed via

mlr wrappers, which call the necessary functions from the packages the methods are implemented in. Other methods have not been implemented in *mlr* yet by the developers, but can be customised by the users. This was done for (TS) priority-Lasso, (TS) IPF-Lasso, the Kaplan-Meier estimate, the clinical (only) reference model and parts of model-based boosting.

To assess the performance, a repeated k-fold cross-validation (CV) strategy is used. In a k-fold CV the data set is split into k subsets (folds) randomly. Each of these subsets is then used for testing once, the corresponding $k-1$ other subsets are used for training. In the end, the performance is averaged over all k testing folds (Couronné et al, 2018). In a repeated CV this is iterated several times. We use 10 x 5-fold CV for the smaller data sets and 5 x 5-fold CV for the larger data sets to keep computation times feasible. Furthermore, we stratify the subsets, so that in each fold a comparable amount of events is included, since the ratio of events and censorings is unbalanced for some data sets.

This resampling strategy is carried out on all data sets for all learners. As 7 larger and 11 smaller data sets and 11 learners are used, this leads to 7 x 25 x 11 + 11 x 50 x 11 = 7975 models to be fit.

Moreover, hyper-parameter tuning is performed. This could also be implemented via *mlr*, but in this study the tuning procedures provided by the underlying packages are used. We denote the resampling strategy used for hyper-parameter tuning *inner resampling* and the repeated CV used for performance assessment *outer resampling.* As specification of the inner resampling strategies we use the default settings of the underlying functions. The learners and the applied inner resampling are described in the *Methods and learners* subsection. In the following, the data sets are presented.

3.2 Data sets

Based on the cancer data that has been gathered by the TCGA Research Network: http://cancergenome.nih.gov/, for 26 cancer types (those with more than 100 samples and five different multi-omics groups) the according data sets were available, of which 18 could be used. Table 13 in the appendix lists all cancer types and the abbreviations used to reference them within the study.

For each of the cancer types there are five different raw data sets, four containing molecular data and one containing clinical data. The molecular data comprises *copy number variation* (cnv), *gene expression* (rna), *miRNA expression* (mirna), and *mutation*. The number of features the groups include is similar over data sets but strongly differs between groups. Figure 1 displays an overview. It is obvious that most features (about 60,000) belong to the cnv group, only a couple of hundred features to mirna, the smallest group. Overall, there are about 80,000 to 100,000 molecular features for every cancer type.

For the analysis these data sets are merged by patient-ID. Since not for all patients every molecular data type is observed, merging the molecular data sets reduces the number of observations. For three cancer types the reduction is severe and leaves no observation behind that does not have a any missing values. So the cancer types CESC, GBM and READ are excluded. The number of patients per group and cancer type is presented in Figure 2. The group noNA represents the number of observations with no missing values after merging the molecular data for that cancer type. One can see that except for the mentioned groups the reduction in observations can be neglected. Only for OV and KIRC the remaining number of observations is about the half of the maximum number of observations within a single group. For KIRC the reduction strongly depends on the fact that the mutation group does not include as many observations as the other groups.

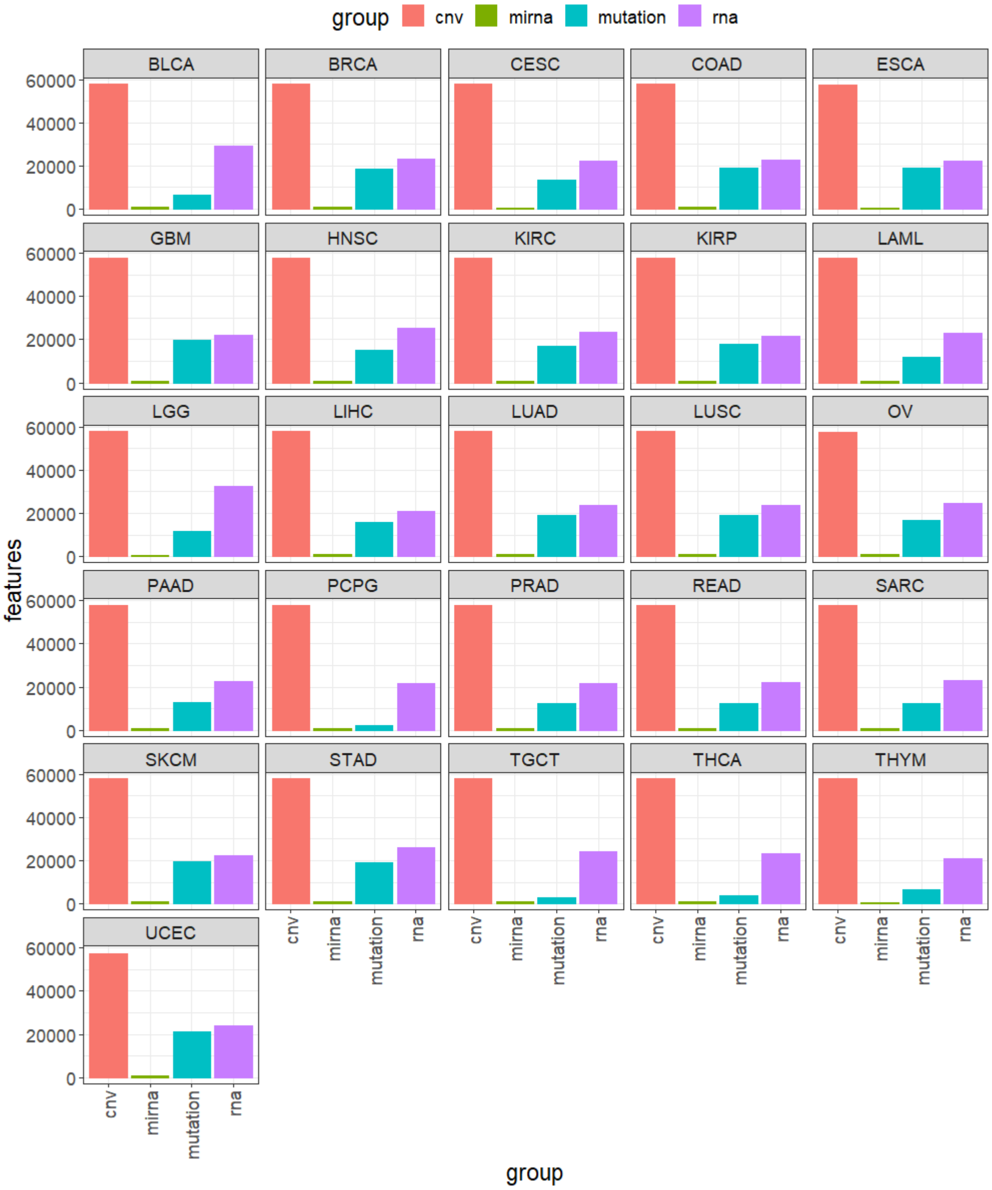


Figure 1: Number of features per molecular group by cancer type.

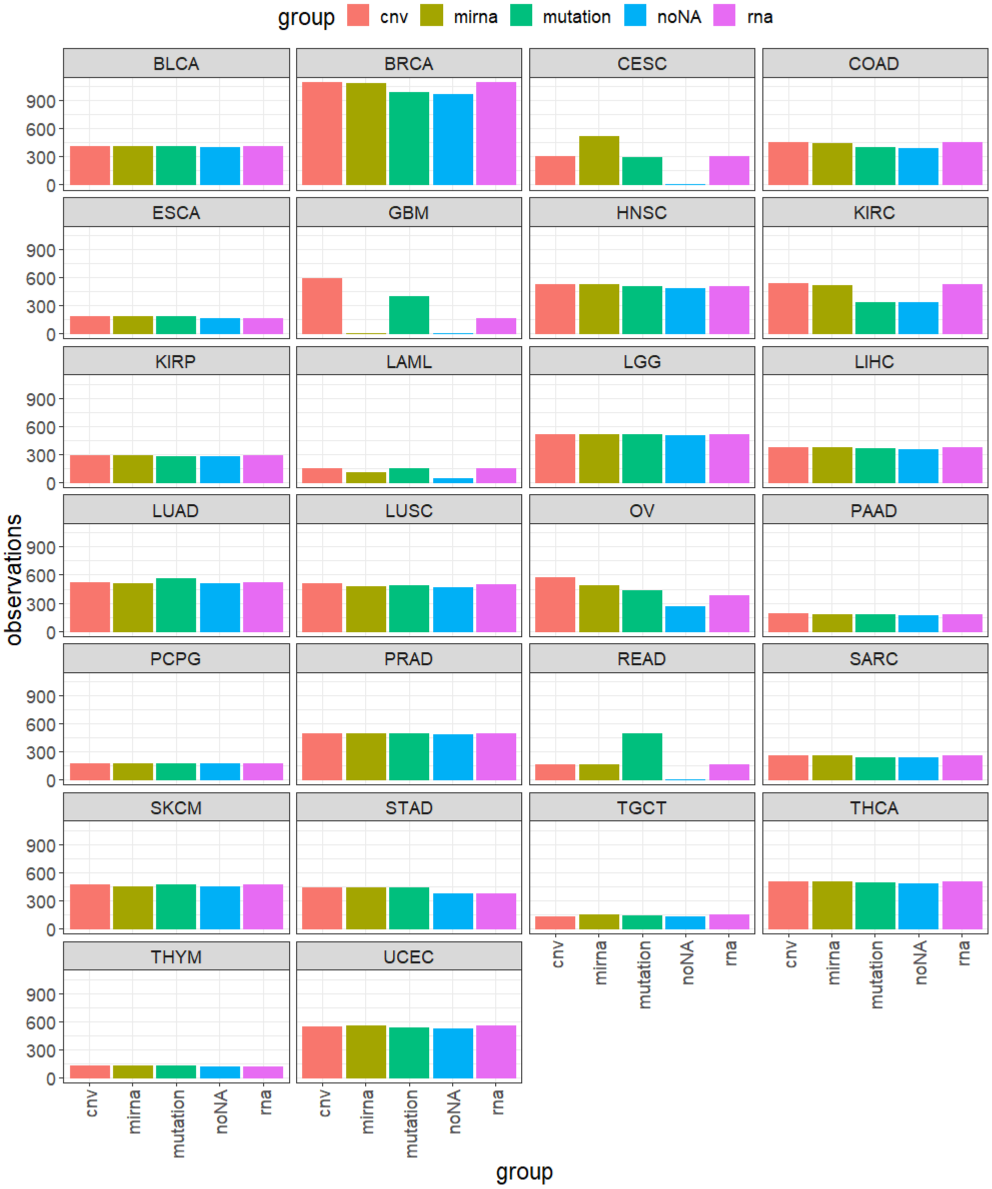


Figure 2: Number of observations per molecular group by cancer type. The *noNA* group refers to the number of observations without NA after merging the molecular data.

Since the outcome of interest is survival time, not (only) the number of observations is crucial, but particularly the number of events, i.e. deaths, which we call the number of *effective cases*. A ratio of 0.2 of effective cases is common (De Bin et al, 2014b). Unfortunately from a statistical point of view, there are some cancer types for which even this ratio is not reached, five of them having less then 5 % effective cases. Of these five cancer types, PCPG and TCGT have a number of effective cases too low to conduct 5-fold cross-validation. Therefore, these cancer types are excluded as well. Furthermore, it turned out that data sets with few number of effective cases lead to extremely long computation times (see *Computation time* section), so only data sets with more than 5 % events where included. That leaves behind data sets for 18 cancer types as basis of the benchmark experiment. For those, the clinical data is merged to the joined molecular data.

The raw clinical data sets contain a lot of features, making further preprocessing necessary. First of all, depending on the cancer type, many of them contain only NAs. Secondly, there are features holding administrative information such as *informed_consent_verified*, which are assumed not to be related to the outcome. Thirdly, there are many features that might be related to the outcome for one cancer type, but not for the other. As the majority of the clinical features has missing values, the question arises which to include for a specific data set while saving as much observations/effective cases as possible. As we do not have any domain knowledge, we decided by a two step strategy.

First, a literature search was conducted to find studies where the specific cancer type was under investigation. Features mentioned to be useful in this studies are defined as the ones to be necessary. But as either not all of them are included in the clinical data sets at hand, or they contain a lot of NAs, not all of them are available. Secondly, if for a cancer type none or only a few of the necessary features are available, we only or additionally use the features that are available for most of the cancer types. That comprises *sex*, *age*, *histological type* and *tumor stage*. They are standardly included, if available. Of course, sex was not included for the sex-specific cancer types OV, TGCT, PRAD, BRCA[1].

Furthermore, most of the clinical features are factors. To utilise them for the methods they are converted to dummy features. Many of the factor features have a lot of levels, for example *histological type* has mostly more than ten levels. As some of the levels only hold few observations, they have to be pooled to obtain computational stability. If, for example, *tumor stage* comprises the levels Stage IA, Stage IB, Stage IC, and IB and IC hold only few observations whereas IA many, they are pooled to Stage I, given that there are other levels such as Stage II and Stage III.

The clinical data set is merged with the joined molecular data by patient-ID, leading to 18 final multi-omics data sets that are used for the benchmark experiment. Since the selected clinical features also hold some NAs, the number of observations gets further reduced in some cases. As the clinical features are regarded as very important, this is accepted.

Table 1 summarises the most important cornerstones of the data sets used in the benchmark study. The first column represents the cancer type, the following five columns provide the number of features per multi-omics group. Moreover, p is the total number of features, N the number of observations, n_{eff} the number of effective cases (i.e. events) and r_{eff} the ratio n_{eff}/N.

[1] BRCA is actually not sex specific as it contains one male patient. Feature *sex* was nevertheless not used for that cancer type.

Cancer	cnv	mirna	mutation	rna	clin.	p	N	n_{eff}	r_{eff}
BLCA	57964	825	18650	23081	5	100525	382	103	0.27
BRCA	57964	835	18847	22694	8	100348	735	72	0.10
COAD	57964	802	19786	22210	7	100769	191	17	0.09
ESCA	57964	763	15162	25494	6	99389	106	37	0.35
HNSC	57964	793	17840	21520	11	98128	443	152	0.34
KIRC	57964	725	12017	22972	9	93687	249	62	0.25
KIRP	57964	593	11610	32525	6	102698	167	20	0.12
LAML	57964	882	6575	29132	7	94560	35	14	0.40
LGG	57964	645	13389	22297	10	94305	149	77	0.18
LIHC	57964	776	15924	20994	11	95669	159	35	0.22
LUAD	57964	799	18966	23681	9	101419	426	101	0.24
LUSC	57964	895	18832	23524	9	101224	418	132	0.32
OV	57447	975	16837	24508	6	99773	219	109	0.50
PAAD	57964	612	12882	22348	10	93816	124	52	0.42
SARC	57964	778	12478	22842	11	94073	126	38	0.30
SKCM	57964	1002	19488	22248	9	100711	249	87	0.35
STAD	57967	787	19141	26027	7	103929	295	62	0.21
UCEC	57447	866	21226	23978	11	103528	405	38	0.09

Table 1: Summary of data sets used for the benchmark experiment

In Tables 14, 15, 16, 17, and 18 in the appendix the included clinical features are listed. Also the reference, on which the information for useful clinical features is based, is included. Finally, the number of originally included effective cases is displayed.

3.3 Methods and learners

In this section, the methods and their specific configurations (which we call *learners*) used within this study are described. As the *mlr* methods are only wrappers for the underlying functions of the packages where the methods are implemented, we refer to this packages and functions in the following. The package names can be found in the parentheses after the method heading. If the method has been customised for this study, it is denoted with (c).

Lasso methods

standard Lasso (glmnet, Friedman et al (2010) and Simon et al (2011))
The penalty parameter λ is chosen via 10-fold CV (inner resampling). No group structure information is used.

Two-step IPF-Lasso (ipflasso, Boulesteix and Fuchs (2015)) (c)
To implement the two-step procedure, additionally code from Schulze (2017) is used. The penalty factors get specified in the first step by computing separate Ridge regression models for every feature group and averaging the coefficients within the groups by the arithmetic mean. These settings have shown to yield reasonable results (Schulze, 2017). Moreover, for inner resampling, 10-fold-CV is used in the first step, 5 x 5-fold CV in the second. Since it is a learner using group structure information, additionally the indices of the features according to group membership are used.

Two-step priority-Lasso (prioritylasso, Klau and Hornung (2017)) (c)
The first step is realised exactly as the first step of TS IPF-Lasso. The priority order is then defined as described in the *Priority-Lasso* section by ordering the groups according to their penalty factors. For the second step a 10-fold CV is used. Group structure information is also provided. Although

recommended otherwise (Klau et al, 2018), the offsets are not estimated via CV to not further increase the computation times.

Two-step priority-Lasso favoring clinical features (prioritylasso, Klau and Hornung (2017)) (c)
Another variant of priority-Lasso is examined. The settings are the same as before. Additionally, the clinical features are favored by assigning the highest priority always to the clinical group. Only the priority order among the molecular groups is specified by the data in the first step. The clinical features are used as an offset when fitting the model of the second block. Furthermore, the clinical features are not penalised (setting parameter *block1.penalization = FALSE*). Again, the offsets are not computed via CV for the same reason as mentioned above.

Boosting methods

Model-based boosting (mboost, Hothorn et al (2018))
Internally, *mlr* uses the function *glmboost* of package *mboost* and sets the family argument to *CoxPH()*. Furthermore the maximum number of boosting steps (m_{stop}) is set to 100. The actual m_{stop} is defined by a *25-boostrap* (BS) procedure via the function *cvrisk*, so the number of boosting steps might be less. The procedure *cvrisk* for inner resampling is originally not implemented in the *glmboost*-wrapper of *mlr*. This was manually added for this study (c). For the learning rate ν the default value of 0.1 is used. Group structure information is not supplied.

Likelihood-based boosting (CoxBoost, Binder (2013))
Again the maximum number of boosting steps is 100. Here the actual m_{stop} is determined by 10-fold CV. The penalty λ is set to default and thus com-

puted according to number of events. No group structure information is used.

Likelihood-based boosting favoring clinical features (CoxBoost, Binder (2013)) Similar to priority-Lasso a second version is considered. The settings are the same as before. Additionally, group structure information is used by specifying the clinical features as mandatory. These features are favored similar to priority-Lasso whilst setting them as an offset and not penalising them (s. the *Likelihood-based boosting* section for details). Further group information is not used, so the molecular data is not distinguished.

Random forest

Two versions of random forest are examined: *randomForestSRC (rfsrc)* (Ishwaran and Kogalur, 2018) and *ranger* (Wright and Ziegler, 2017). They share the same theoretical background, but differ in the implementation. Thus, for random forest rather two different implementations are compared, in contrast to the comparison of different (theoretical) frameworks like for boosting. As in the study of Couronné et al (2018), no hyper-parameter tuning is conducted and for both implementations their default settings are used. It follows that the number of trees (*ntrees*) is 500 for *ranger* and 1000 for *rfsrc*. Furthermore, the parameter *mtry* is set to $\lceil\sqrt(p)\rceil$ for *rfsrc* and $\lfloor\sqrt(p)\rfloor$ for *ranger*. The minimal node size is 3 for both.

Reference methods

To reference the complex methods, two base procedures are used, one of which is the *Kaplan-Meier estimate* which does not use any feature information to estimate the survival probability. Moreover, a *clinical reference model* that uses only the clinical features to predict the survival probabilities is fit for every data set. This clinical reference is a Cox proportional hazard model computed via the *coxph* function of the *survival* package (Therneau, 2015). The Kaplan-Meier estimate is computed via *survfit* from the same package. The two methods are customised in *mlr* for the study (c).

Table 2 summarises the specific learners described above. It displays the used R packages and functions, and the inner resampling strategy for tuning (if conducted). Furthermore, the use of group structure information is indicated in column *structure*. Some of the learners need standardized features. This is indicated in column *standardized*. If a method requires standardized features, the learner handles that by means of the underlying function itself, so the data is not standardized manually beforehand.

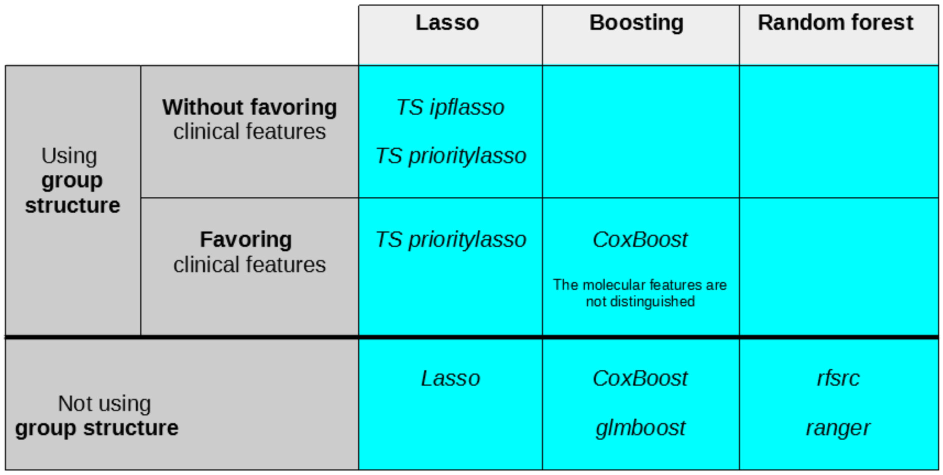
learner	method	package::function	tuning	structure	standardized
Lasso	Standard Lasso	glmnet::cv.glmnet	10-f-CV	no	yes
ipflasso	TS IPF-Lasso	glmnet::cv.glmnet	10-f-CV	yes	yes
		ipflasso::cvr.ipflasso	5x 5-f-CV	yes	yes
prioritylasso	TS priority-Lasso	glmnet::cv.glmnet	10-f-CV	yes	yes
		prioritylasso::prioritylasso	10-f-CV	yes	yes
prioritylasso favoring	TS priority-Lasso	glmnet::cv.glmnet	10-f-CV	yes	yes
		prioritylasso::prioritylasso	10-f-CV	yes	yes
glmboost	Model-based boosting	mboost::glmboost	25 BS	no	centered
CoxBoost	Likelihood-based boosting	CoxBoost::cv.CoxBoost	10-f-CV	no	yes
CoxBoost favoring	Likelihood-based boosting	CoxBoost::cv.CoxBoost	10-f-CV	yes	yes
rfsrc	Random forest	randomForestSRC:rfsrc	no	no	no
ranger	Random forest	ranger::ranger	no	no	no
Clinical only	Clinical reference model	survival::coxph	no	no	no
Kaplan-Meier	Kaplan-Meier estimate	survival::survival	no	no	no

Table 2: Summary of learners used for the benchmark experiment. Fixed priority 1 and no penalization via *block1.penalization = FALSE* for clinical features for prioritylasso favoring. Clinical features set to be mandatory via *unpen.index* in CoxBoost favoring. The first step of IPF-Lasso and priority-Lasso is conducted via the function *cv.glmnet* of the *glmnet* package.

3.4 Dimensions of assessment and comparability

The performance of the methods is evaluated in different dimensions. First of all, as prediction methods are examined, the prediction performance is assessed via the integrated Brier-score (ibrier) and the c-index based on Uno et al (2011) (in the following simply cindex). Concerning the ibrier, in general the methods under investigation may not be regarded as useful if they perform worse than the Kaplan-Meier estimate, which does not use any information contained in the features and may thus be regarded as null model. Concerning the cindex, the Kaplan-Meier estimate corresponds to a constant prediction of 0.5, which represents the null model prediction.
The second dimension is the sparsity of the resulting models, which has two aspects: sparsity on feature level and sparsity on group level. The later refers to the aspect whether features of only some groups, and not of all groups, are selected. The sparsity on feature level, in contrast, refers to an overall sparsity, i.e. the total amount of selected features in the resulting models. As random forest does not yield easily interpretable models, it is not assessed in this dimension. As computation times are a crucial aspect of model fitting, the computation times are used as third dimension.

Another important aspect is the different use of group structure information. Some of the methods don't use any information of such kind, some favor clinical data over molecular data, and some incorporate every multi-omics group individually. Thus, the differences in performance might not only result from using different prediction methods. The differences may also arise from the way in which the group structure information is included. So, the comparability with regard to prediction performance is only given within methods that use the same strategy to include group information. Figure 3 illustrates the comparability with respect to the different ways of using group structure.



		Lasso	**Boosting**	**Random forest**
Using **group structure**	**Without favoring** clinical features	*TS ipflasso* *TS prioritylasso*		
	Favoring clinical features	*TS prioritylasso*	*CoxBoost* The molecular features are not distinguished	
Not using **group structure**		*Lasso*	*CoxBoost* *glmboost*	*rfsrc* *ranger*

Figure 3: Comparability of learners with respect to the use of group structure information by overall modelling approach, where the learners within a row are comparable. CoxBoost favoring clinical features does not further distinguish the molecular data, thus making slightly different use of the structure as the priority-Lasso learner in the same row.

Concluding, there are three dimensions of assessment: prediction performance, computation time, and sparsity, whereby the later one is split into two sub-dimensions: feature level and group level. When assessing the performance, additionally the comparability of the methods based on the dimensions of group structure inclusion (naive, favoring clinical features, including multi-omics groups) has to be taken into account.

4 Results

In this section the benchmark results are presented. We first describe the assessment dimensions *computation time* and *sparsity*. Then an expansive description of the last dimension *prediction performance*, taking into account the comparability dimensions, is given.

For some CV iterations the model fitting was not successful, leading to NAs for the assessment measures for these iterations. This is common in benchmark experiments of larger scale (Bischl et al, 2013). To cope with such model failures we follow the strategies described by Probst et al (2018) and Bischl et al (2013). If a learner fails in more than 20 % of the CV iterations for a given data set, we assign (for the failing iterations) the data independent values of the prediction performance measures (0.25 for ibrier and 0.5 for cindex) and the mean of the other iterations for the computation time and the number of selected features. If a learner fails in less than 20 %, the performance means of the successful iterations of this learner are assigned for all measures.

4.1 Computation time

The learners' computation times are measured based on the time needed for fitting the model (training time). Table 3 shows the average computation time of each learner. The value in the middle column is the mean computation time, averaged over the CV iteration and then over the data sets.

Clearly, rfsrc is the fastest procedure, followed by glmboost and standard Lasso. The CoxBoost variants need about 3.5 times as much time as glmboost, and ranger about 2.6 times as much as rfsrc. The three Lasso variants using group structure are very time intensive with ipflasso being by far the most time consuming. Though, it has to be taken into account that the random forest learners are not subject to inner resampling and the default inner resampling for ipflasso is 5 x 5-fold CV, whereas for most of the other learners it is 10-fold CV.

Learner	computation time in hours	computation time per CV-fold in minutes
rfsrc	1.89	4.30
glmboost	2.32	4.14
Lasso	4.93	8.91
ranger	6.61	13.96
CoxBoost favoring	8.36	14.92
CoxBoost	8.37	15.01
prioritylasso	19.2	27.46
prioritylasso favoring	19.4	27.67
ipflasso	26.0	39.70

Table 3: In the middle column the mean computation times for the whole procedure including outer resampling is depicted. The right column shows the mean computation times for a single CV iteration (mean time needed to fit a single model).

Of course, the computation times depend on the size of the data sets, a reason for the fact that ranger ranks fourth. Figure 4 displays the mean computation times in seconds for one CV iteration for the different learners and data sets. The data sets are ordered from smallest (LAML) to largest (BRCA). It has to be emphasised that the data set size is not increasing linearly. rfsrc is the fastest algorithm for most of the data sets, followed by glmboost and ranger. Interestingly, ranger is second fastest for smaller data sets, but is increasingly slow compared to the other methods for data sets larger than KRIP. Eventually, it is outperformed by all but one of the other methods for the four greatest data sets (although no inner resampling is conducted).

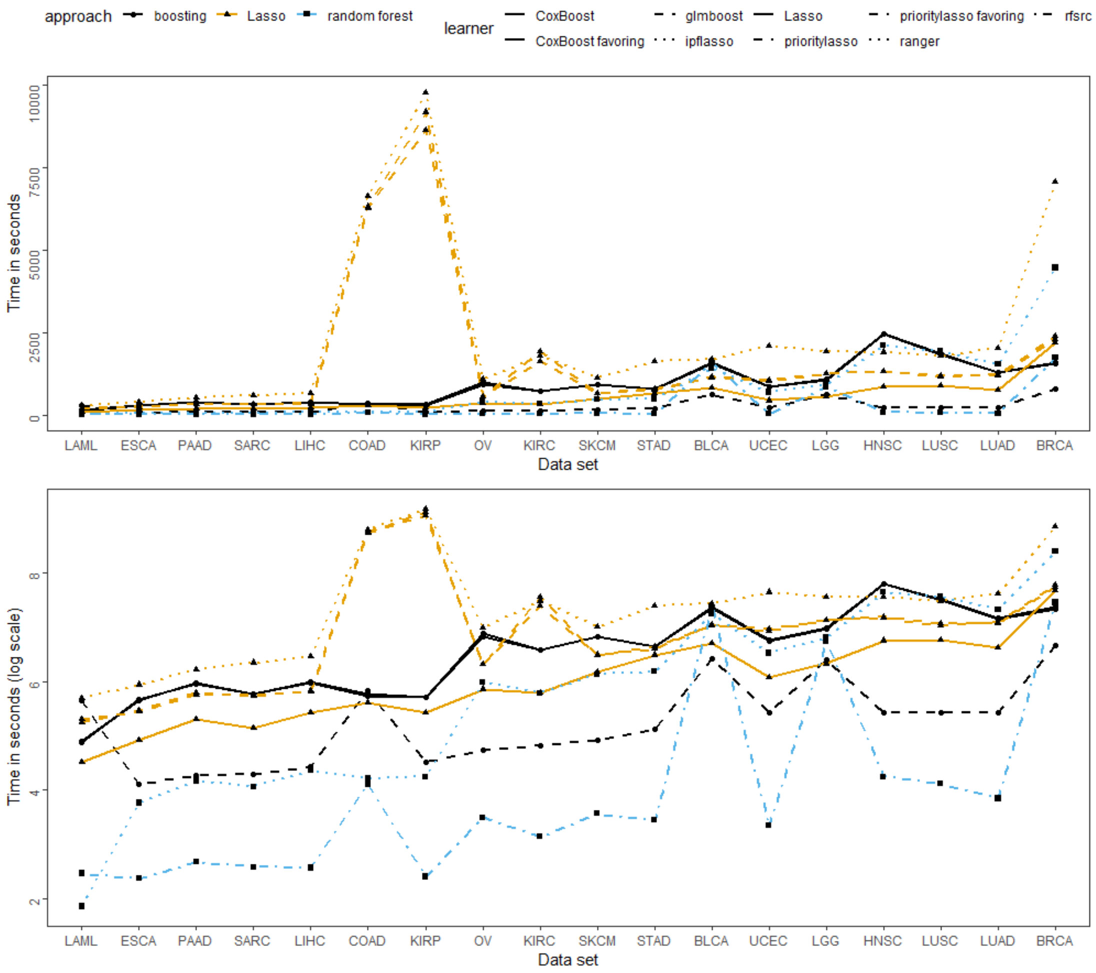


Figure 4: Average computation time for one CV iteration of the different learners on different data sets. The lower plot is depicted on logarithmic scale. The data sets are ordered from smallest (LAML) to largest size.

The Lasso methods taking group structure information into account are among the slowest methods, where IPF-Lasso is slower than priority-Lasso. Moreover, there is almost no difference when favoring clinical variables, regarding CoxBoost as well as prioritylasso.
Besides the data set size, the effective number of cases influences the computation time. COAD and KIRP are among the smaller data sets with 17 (9 %) and 20 (12 %) events respectively. Especially the Lasso variants taking group structure into account yield a grave increase in computation time for these data sets. Also glmboost and rfsrc are affected in the case of COAD. For the Lasso variants the computation times even exceed by far the times needed for the largest data set, a reason why data sets with even less events were excluded from the study.

4.2 Model sparsity

To assess sparsity, the number of non-zero coefficients of the resulting model of each CV iteration is counted. These values are averaged over the CV iterations and then averaged over all data sets (for every learner). As random forest models do not yield such coefficients, the two random forest variants are not considered within this dimension.

4.2.1 Sparsity on feature level

Considering overall sparsity, i.e. sparsity on feature level, is particularly interesting for practical purposes, since sparse models are easier to interpret and to communicate. On average, as Table 4 shows, ipflasso leads to the sparsest models, followed by the boosting variants with glmboost being sparser than the naive CoxBoost. Standard Lasso ranks in midfield and prioritylasso models are least sparse.

Learner	ipflasso	glmboost	CoxBoost	CoxBoost fav.
No. of features	5.56	6.45	9.93	13.39
Learner	Lasso	prioritylasso	prioritylasso fav.	
No. of features	15.74	25.74	30.14	

Table 4: Average number of selected features.

Moreover, ipflasso shows the lowest variability across data sets, whereas prioritylasso yields the greatest, see Figure 5. It also becomes obvious that glmboost is sparser than ipflasso for about half of the data sets.
Favoring clinical variables leads to less sparse models for CoxBoost as well as for prioritylasso, which also leads to lower variability across data sets. Summarising, boosting leads to sparser models when compared to the Lasso variants, except for IPF-Lasso which is the sparsest method.

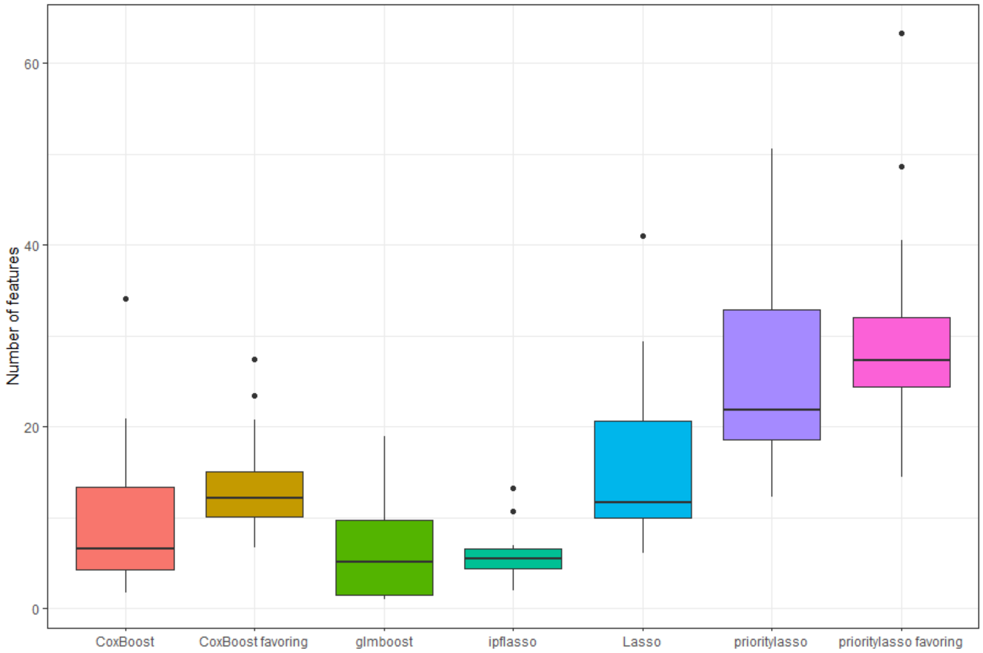


Figure 5: Number of selected features.

4.2.2 Sparsity on group level

Figure 6 displays the number of selected features by multi-omics group for all but the random forest and reference learners.

Regarding feature selection on group level, three aspects stick out. First of all, ipflasso yields strong sparsity on a group level. Except for some outliers, only clinical and mirna features are selected for most data sets. Furthermore, the boosting variants and the standard Lasso yield groups for which mostly no features are selected: this comprises the groups mirna and clinical, and, concerning CoxBoost favoring and glmboost, cnv.

This, secondly, exemplifies the problem of high- and low-dimensional feature groups treated equally. As has been pointed out before in the *Background* section, due to their low-dimensional character, clinical features get lost within the huge amount of molecular features. It becomes obvious that this does not only apply for clinical features. The mirna group is, in comparison to the other molecular groups, low-dimensional with sizes ranging from 585 to 1002 features. Learners not taking any group structure into account, CoxBoost, glmboost, and Lasso, fail to include clinical or mirna features. CoxBoost favoring, which only differentiates clinical and molecular features, does not select mirna features.

Thirdly, learners taking the multi-omics group structure into account include, in contrast, features of both low-dimensional groups. ipflasso even only selects features of these groups. Thus, including the multi-omics group structure saves low-dimensional groups from being discounted, which is very important, since the naive learners show, overall, a worse prediction performance than learners using group structure.

Interestingly, features of the largest group, cnv, are also often not included by the boosting methods, and Lasso variants using group structure select from this group the lowest amount. This indicates that this group is not very useful. Finally, priority-Lasso in both variants is not able to select groups.

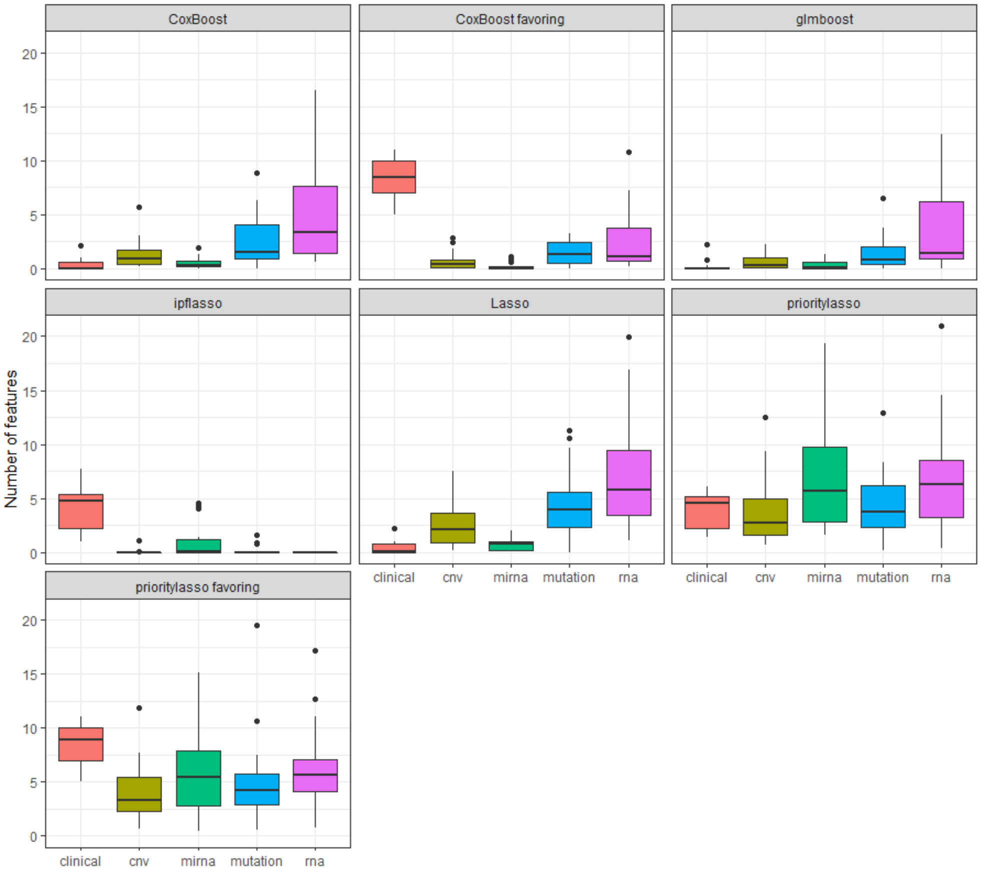


Figure 6: Number of selected features by multi-omics group.

Concluding, IPF-Lasso is the sparsest method on feature as well as group level. Instead, priority-Lasso is the less sparsest method, again on both levels. The naive methods, particularly naive boosting methods, also lead to sparse models on feature and group level, but to the disadvantage of low-dimensional feature groups.

4.3 Prediction performance

The main goal of the study at hand is to compare prediction methods using multi-omics data. In this section we first give an overview of the prediction performance. We then concentrate in the *Using multi-omics data* subsection on the differences in performance with respect to the use of the multi-omics data. First of all, we look upon the added predictive value of the molecular data. Furthermore, we present in general the differences resulting from the different forms of multi-omics data inclusion by comparing the naive strategy learners on the one side with the learners using group structure information (also called structured learners in the following, comprising learners favoring clinical features as well as learners using multi-omics structure) on the other side. In the subsequent *Comparing prediction methods* section, we compare the results of the different prediction methods/learners with respect to the comparability dimensions.

4.3.1 Overview of prediction performance

For every learner, Table 5 shows the average performance based on the cindex and the ibrier as well as the ranks based on these measures. To obtain the final values, the performance of each learner is averaged over all CV iterations and then averaged over all data sets. The ranks are computed similarly from 1 (best) to 11 (worst).

The main findings are, first of all, that there is no learner clearly outperforming the clinical learner on average over all data sets. In fact, only CoxBoost favoring performs slightly better based on the ibrier and according to the ranks. Secondly, regarding the cindex, the structured learners perform better than the naive learners, with the favoring learners ahead of the non-favoring structured learners.

	means		ranks	
	cindex	ibrier	cindex	ibrier
Clinical only	**0.617**	0.176	2.83	4.22
CoxBoost fav.	0.614	**0.175**	**2.78**	**4.17**
prioritylasso fav.	0.604	0.183	3.72	6.89
prioritylasso	0.588	0.182	5.44	6.61
ipflasso	0.570	0.182	5.44	4.72
ranger	0.567	0.178	6.56	6
rfsrc	0.566	0.184	6.67	8
Lasso	0.542	0.196	6.67	7.61
Kaplan-Meier	0.5	0.180	8.89	6.89
glmboost	0.485	0.184	7.89	6.06
CoxBoost	0.405	0.176	9.11	4.83

Table 5: Mean performance results and ranks for all learners. The best performances are indicated in bold.

Figure 7 shows the performance distributions. Obviously, all other learners perform worse than the clinical learner (red line). Only CoxBoost favoring yields a slightly better result for the ibrier and based on the ranks. Another important fact is that all learners (except CoxBoost for the cindex) perform better than the Kaplan-Meier reference (dashed lines; corresponds to 0.5 for the cindex). It also becomes obvious that the structured learners (four learners on the right) yield better performances than the naive learners, with ipflasso having highest variability, and that the learners favoring clinical features perform slightly better than the other structured learners.

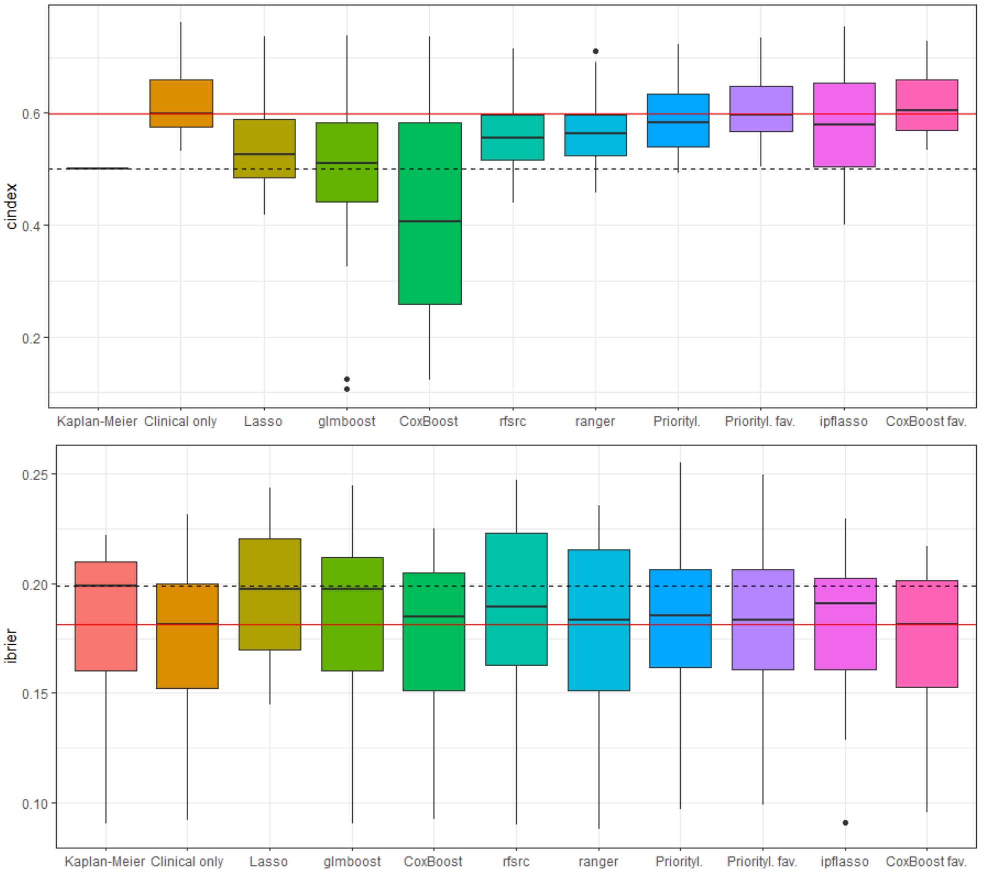


Figure 7: Prediction performance based on the cindex and the ibrier. For the cindex higher values are better, lower values are better for the ibrier.

Considering the naive learners, according to the cindex the random forest variants perform clearly better, but the results are not as clear regarding the ibrier. Moreover, the standard Lasso and the naive boosting learners (glmboost, CoxBoost) perform only marginally better or even worse than the Kaplan-Meier estimate (see also Table 5). This indicates that they are, on average, not very useful.

4.3.2 Using multi-omics data

Added predictive value To assess the added predictive value of the molecular data, we follow *approach A* proposed by Boulesteix and Sauerbrei (2011), thus comparing learners obtained by only using clinical features and combined learners, i.e. learners using clinical and molecular features. Since it is emphasised that for this validation approach the combined learners should not be derived by the naive strategy, these learners are not considered here. Despite that there is no learner outperforming the clinical learner on average over all data sets, for several data sets there is at least one structured learner outperforming the clinical learner. This indicates, according to the validation approach, that using additional molecular data is useful and leads to better prediction performance in these cases. However, as Table 6 and Figure 8 show, often the differences are small. Moreover, only for some data sets this finding is supported by both measures (see Table 6). For several data sets the structured learners perform better only based on one measure. Clearly better performance based on the cindex can be found for HNSC, KIRP, and LGG, based on the ibrier for LAML and SARC.

These findings, having data sets where molecular data adds predictive value and data sets where it does not, is consistent with findings by others mentioned in the *Added predictive value* section, which also show that using molecular data adds predictive value in some cases and doesn't in others.

In addition, the findings of the study at hand indicate that if there is additional predictive value in the molecular data, this does not automatically mean that this potential is used by every learner/method. See Figure 8 and, for example, LAML where prioritylasso favoring clinical features performs better than the reference model, whereas CoxBoost favoring does not.

	learner	cindex	ibrier	clin. cindex	clin. ibrier
BLCA	**CoxBoost fav.**	0.640	0.190	0.633	0.192
BRCA	CoxBoost fav.	0.643	0.149	0.637	0.147
COAD	CoxBoost fav.	0.553	0.107	0.541	0.101
HNSC	**CoxBoost fav.**	0.574	0.203	0.554	0.210
KIRC	ipflasso	0.755	0.144	0.761	0.146
KIRP	priorityl. fav.	0.610	0.146	0.572	0.140
LAML	**CoxBoost fav.**	0.607	0.215	0.596	0.231
LGG	**CoxBoost fav.**	0.712	0.155	0.652	0.168
LIHC	**CoxBoost fav.**	0.602	0.166	0.586	0.169
LUAD	Priorityl. fav	0.665	0.174	0.663	0.172
LUSC	Priorityl. fav.	0.537	0.216	0.531	0.216
OV	ipflasso	0.580	0.168	0.598	0.173
PAAD	Priorityl. fav	0.684	0.191	0.683	0.190
SARC	**ipflasso**	0.676	0.189	0.673	0.202
SKCM	CoxBoost fav.	0.590	0.192	0.581	0.191
UCEC	**ipflasso**	0.690	0.091	0.686	0.092

Table 6: Data sets with at least one structured learner outperforming the clinical learner. The performances of the best structured (second, third and fourth columns) and the clinical learner (last two columns) are depicted. If the structured learner outperforms the clinical learner on both measures, it is indicated in bold.

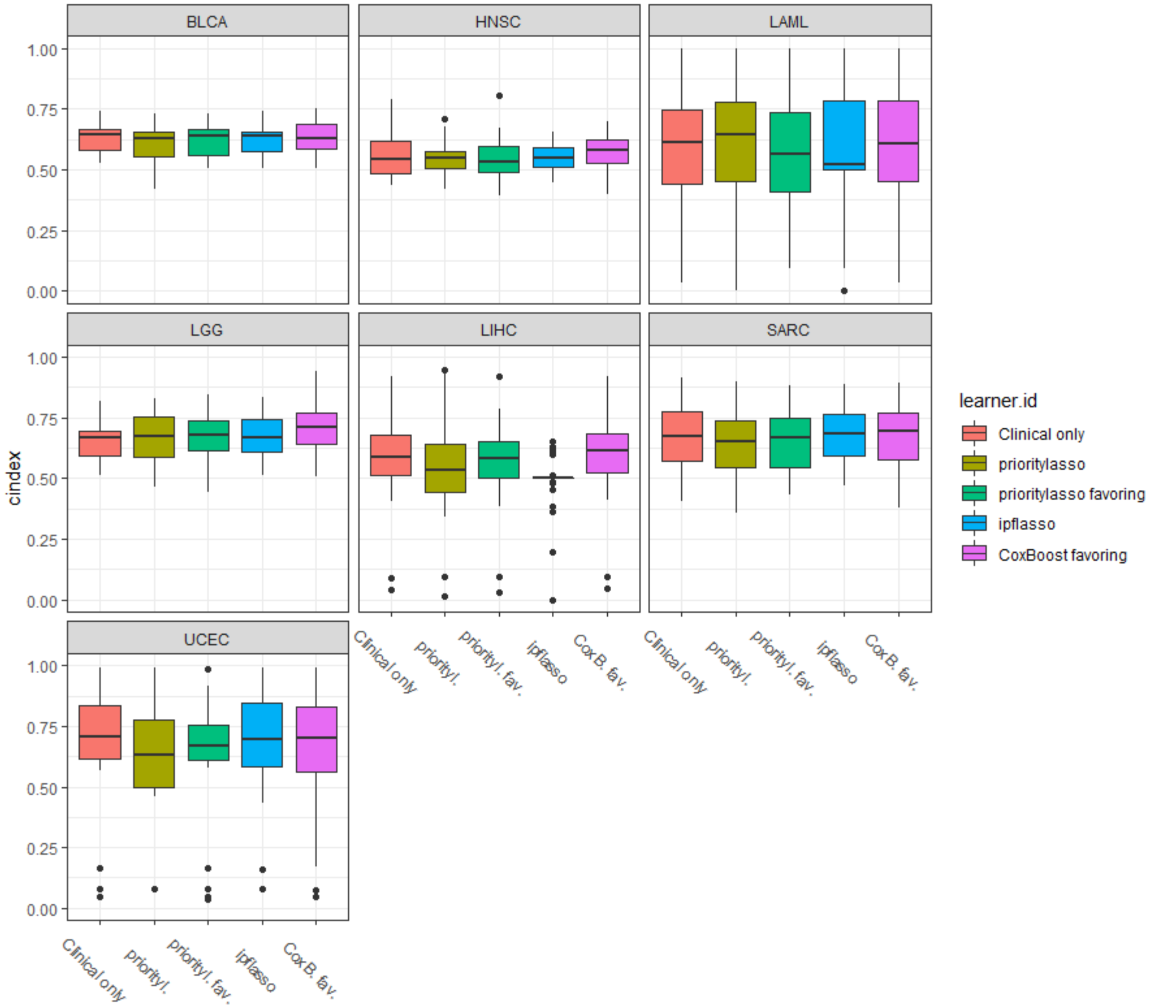


Figure 8: cindex for the structured learners and the clinical learner. Depicted are only those data sets for which the best learner performs better than the clinical learner on both measures (see Figure 10 in the appendix for the ibrier.)

Moreover, different configurations of a method with respect to the usage of group information affect the potential of using molecular data. Also for LAML, prioritylasso favoring clinical features performs better than the clinical model, prioritylasso not favoring clinical features performs equally. Similar findings are valid for BRCA (not depicted).
Figure 8 also shows that ipflasso could not be fit in several CV iterations for LIHC, since many iterations yield a value of 0.5.

Summarising, for some data sets the molecular data hold additional predictive value, although mostly it leads to only a small increase in performance. Furthermore, it does not only depend on the data/cancer type whether molecular features add predictive value, but also the method and the specific configuration used to build the model seem to be important.

Including group structure In general, the results affirm that using the naive strategy of treating clinical and molecular features equally, i.e. not taking the (two) different groups into account, leads to a worse performance in comparison to methods where the clinical and the molecular data are taken into account differently. The later comprises learners favoring clinical features as well as learners taking the whole multi-omics group structure into account. Table 7 shows the mean performance of the naive learners and the structured learners based on the cindex and by data set. Each value is computed as average over the naive respectively the structured learners' mean cindex values.
Only for LGG the mean of the naive learners is higher. It also becomes obvious that for eight data sets the naive learners perform equally or worse than the Kaplan-Meier estimate, whereas the same is true only for one data set for the structured learners. There are similar findings for the ibrier (see Table 8), yet the naive learners perform, on average, better in five cases.

	BLCA	BRCA	COAD	ESCA	HNSC	KIRC	KIRP
naive	0.578	0.394	0.396	0.407	0.544	0.673	0.525
structured	**0.626**	**0.591**	**0.496**	**0.511**	**0.554**	**0.735**	**0.545**
	LAML	LGG	LIHC	LUAD	LUSC	OV	PAAD
naive	0.535	**0.686**	0.534	0.511	0.406	0.382	0.574
structured	**0.594**	0.683	**0.551**	**0.663**	**0.518**	**0.588**	**0.654**
	SARC	SKCM	STAD	UCEC			
naive	0.618	0.481	0.497	0.515			
structured	**0.659**	**0.575**	**0.562**	**0.651**			

Table 7: Mean performance of naive learners and learners using group structure based on the **cindex**. The means are computed over the mean performance of each learner.

	BLCA	BRCA	COAD	ESCA	HNSC	KIRC	KIRP
naive	0.205	0.160	**0.103**	0.2344	0.216	0.159	**0.131**
structured	**0.197**	**0.154**	0.119	**0.2343**	**0.211**	**0.150**	0.136
	LAML	LGG	LIHC	LUAD	LUSC	OV	PAAD
naive	**0.203**	0.172	**0.163**	0.204	0.232	0.193	0.206
structured	0.227	**0.163**	0.184	**0.174**	**0.220**	**0.171**	**0.196**
	SARC	SKCM	STAD	UCEC			
naive	**0.181**	0.218	0.214	0.122			
structured	0.201	**0.195**	**0.195**	**0.106**			

Table 8: Mean performance of naive learners and learners using group structure based on the **ibrier**. The means are computed over the mean performance of each learner.

Individually considering the naive learners and learners using group structure, it becomes clear that the performance strongly varies across learners and across data sets. Figures 11 and 12 in the appendix show the performance distribution measured per cancer type and learner. The better performance of learners using group structure is obvious for cancer types LUAD, STAD an OV. For the other data sets the picture is not as obvious.

Still, there are exceptions from the rule, where this finding is reverted. It turns out that for a minority of the data sets the naive learners in general and the random forest variants in particular perform better than the learners using group structure. Taking a closer look, for several of these data sets the clinical learner performs only slightly better or even worse than the Kaplan-Meier estimate. This indicates that for these data sets the clinical data is not very useful for prediction. Inspecting Figure 9 this becomes obvious. It shows the cindex and the ibrier for data sets where at least some of the naive learners perform well.

Regarding the cindex it turns out that the random forest variants are the well performing naive learners and clearly outperform the clinical learner and the structured learners. Similar results arise when looking at the ibrier. Again, the random forest variants perform best (COAD, KRIP, LAML, LIHC), but in this case also the other naive learners often perform better than the clinical learner and some of the structured learners. Moreover, the clinical learner does not yield clearly better results than the Kaplan-Meier estimate, again indicating that the clinical features are not very useful in these cases.

Overall, for these data sets the random forest variants perform clearly best regarding both measures. The other naive learners perform well based on the ibrier. So, the naive learners, especially the random forest variants, may also show (very) good prediction performance, in some cases even better than the structured learners. Interestingly, this corresponds with data sets with less informative clinical features.

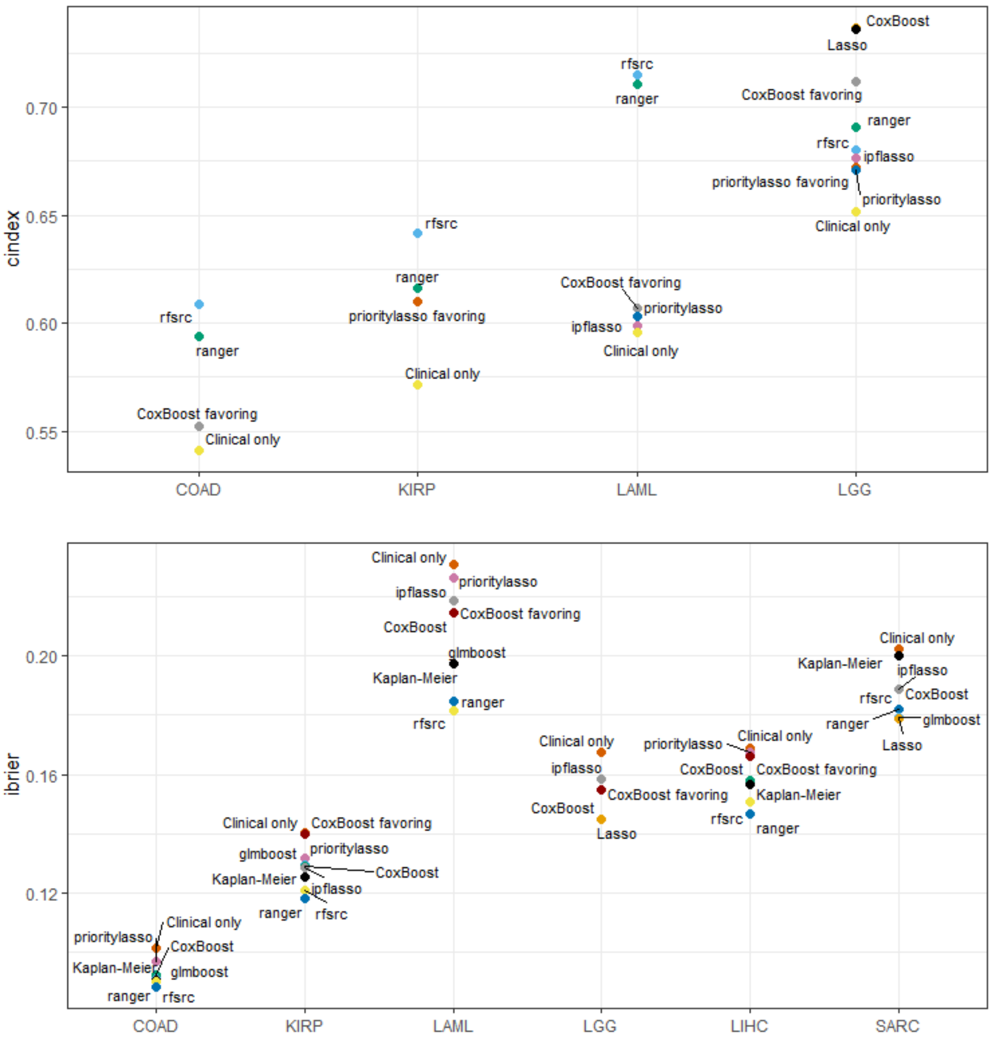


Figure 9: cindex and ibrier for the data sets where naive learners perform well. For each data set, only the learners better than the clinical learner are displayed, so the number of learners is not equal across data sets. The Kaplan-Meier estimate corresponds to a value of 0.5 for the cindex.

The LGG data set is a special case. First of all, the clinical features hold some useful information, since the clinical learner performs clearly better than the Kaplan-Meier estimate (cindex = 0.652, ibrier = 0.168 vs. ibrier = 0.200 for the Kaplan-Meier estimate). Furthermore, regarding the cindex, all other learners except glmboost perform markedly better than the clinical. Thus, not only the clinical features have predictive value, but also the molecular ones. Moreover, the naive learners perform better than the structured learners, with overall best performing learners being Lasso and CoxBoost.
Regarding the ibrier, the situation is slightly different. Still the naive methods, with Lasso and glmboost in lead, perform best. But here all structured learners except ipflasso perform worse than the clinical learner and the Kaplan-Meier estimate.
Overall, this is an interesting result, since clinical as well as molecular features seem to hold different information, but the naive learners are able to use it better.

4.3.3 Comparing prediction methods

As has been outlined in the *Dimensions of assessment and comparability* section, to obtain comparability it is necessary to take into account how the feature groups are treated. Thus, in this section we compare naive methods, methods favoring clinical features and methods using the multi-omics group structure separately and present the results of comparable methods individually.

Naive methods CoxBoost and glmboost, the two random forest learners and the standard Lasso learner are fit with the naive strategy. Thus, for every overall modelling approach (Lasso, boosting, random forest) there are representatives. The average results and ranks for these learners over all data sets can be found in Table 9.

	means		ranks	
	cindex	ibrier	cindex	ibrier
ranger	**0.567**	0.178	2.94	2.5
rfsrc	0.566	0.184	2.89	3.67
Lasso	0.542	0.196	**2.22**	3.61
glmboost	0.485	0.184	3.06	3.11
CoxBoost	0.405	**0.176**	3.89	**2.11**

Table 9: Mean performance results and ranks of the naive learners. The best performance is indicated in bold.

In general, the results are not consistent over the two measures. According to the cindex, random forest performs best, regardless which implementation is used, whereas CoxBoost performs best according to the ibrier. When considering the ranks based on the cindex, the picture is different, with Lasso ranking highest. The results based on the ibrier are consistent: here CoxBoost performs best and ranks highest.

Regarding the average performance based on the cindex, both random forest learners outperform the next best learner Lasso by about 0.025. These three learners perform slightly better than the Kaplan-Meier estimate. In contrast, the boosting variants perform worse than the the Kaplan-Meier estimate, with glmboost performing better than CoxBoost.

Regarding the ibrier, the situation is different. First of all, the difference between ranger and rfsrc is more noticeable, with ranger performing better by 0.006 in contrast to 0.001 for the cindex. Moreover, the boosting variants perform well compared to the other learners and Lasso performs worst. Yet, not every learner performs better than the Kaplan-Meier estimate (0.180).

Summarising, regarding the overall prediction approaches and the resulting methods, there is no obviously best performing approach or even method. Though by tendency, the random forest variants perform best: Regarding the cindex they outperform the other methods, regarding the ibrier they are among the best performing methods. In contrast, the boosting learners perform badly based on the cindex, but competitively based on the ibrier. Lasso performs worst based on the ibrier and worse than the random forest approaches based on the cindex, though it ranks best accoring to the cindex.

Using structure - favoring: priority-Lasso vs. likelihood-based boosting There are two methods for which models favoring clinical features have been fit: likelihood-based boosting and priority-Lasso (with learners CoxBoost favoring and prioritylasso favoring). Hence, only two of the overall modelling approaches (Lasso and boosting) are represented. Table 10 shows the average performance results and ranks.

	means		ranks	
	cindex	ibrier	cindex	ibrier
CoxBoost fav.	**0.614**	**0.175**	**1.39**	**1.11**
prioritylasso fav.	0.604	0.183	1.61	1.89

Table 10: Mean performance and ranks of learners favoring clinical features.

Here, the results are unambiguous with CoxBoost performing better than the prioritylasso, although the performance differences are small. Furthermore, both learners perform better than the Kaplan-Meier estimate based on the cindex, but only CoxBoost performs better than the Kaplan-Meier estimate (0.180) based on the ibrier.

Thus, according to these findings, likelihood-based boosting yields better results than priority-Lasso when clinical features are favored, even though priority-Lasso further distinguishes the molecular data.

Using structure - multi-omics: priority-Lasso vs. IPF-lasso We now consider the methods which take the whole multi-omics group structure into account without favoring the clinical features. This comprises IPF-Lasso and, again, priority-Lasso. Thus, only Lasso-based methods are considered. Boosting and random forest are not represented. Table 11 shows the average performance results and ranks.

	means		ranks	
	cindex	ibrier	cindex	ibrier
prioritylasso	**0.588**	0.182	1.5	1.83
ipflasso	0.570	0.182	1.5	**1.17**

Table 11: Mean performance results and ranks of the learners using multi-omics group structure.

Compared to the favoring methods/learners, the situation is not as obvious for this case. Overall, priority-Lasso shows by tendency better prediction performance than IPF-Lasso. It outperforms IPF-Lasso based on the cindex and yields equal results for the ibrier. But looking at the ranks, the findings are contradictory. On the basis of the ibrier, IPF-Lasso is ranked higher than priority-Lasso most of the times. On the basis of the cindex, they are equally often in the first place.

Thus, there is no clearly best performing method like in the case of the favoring methods. Moreover, both methods perform only slightly better than the Kaplan-Meier estimate based on the cindex and worse based on the ibrier. Since IPF-Lasso yields sparser models, it might be preferable when sparsity is important. Yet, priority-Lasso is markedly better according to the cindex.

To favor or not to favor clinical features As has been outlined in the *Background* section favoring clinical features over molecular features is preferable for several reasons, one of which is that they have often proved to be of value and practitioners want them to be included by all means.

According to the findings of the benchmark experiment at hand, favoring clinical features leads to clearly better prediction performance. For methods where only clinical and molecular features are treated differently (likehihood-based boosting), this is in line with the findings of others (see De Bin (2016) and the reference therein). Table 12 displays the results for prioritylasso and prioritylasso favoring as well as for CoxBoost and CoxBoost favoring. Differentiating the clinical features from the molecular features strongly increases the prediction performance of likelihood-based boosting (representend by learners CoxBoost and CoxBoost favoring) according to the average cindex and the ranks, though there is only slight improvement based on the average ibrier. Overall, favoring clinical features raises likelihood-based boosting from the worst to the best performing prediction method.

	means		ranks	
	cindex	ibrier	cindex	ibrier
prioritylasso	0.588	0.182	1.78	1.5
prioritylasso fav.	0.604	0.183	1.22	1.5
CoxBoost	0.405	0.176	1.94	1.61
CoxBoost fav.	0.614	0.175	1.06	1.39

Table 12: Mean performance results and ranks of methods with configurations favoring and not favoring clinical features. The ranks are computed only among the learners of one method, so 1 is best and 2 is worst.

According to our findings, this also holds true for methods using the multi-omics group structure. For priority-Lasso the increase is not as strong, but still notably when regarding the cindex. Therefore, favoring clinical features might also be an advantage when using multi-omics group structure. But since this finding is based on only one example (priority-Lasso), more research has to be conducted to confirm this.

Summarising, all this indicates that, whether or not the molecular data types are distinguished, favoring clinical features leads to better prediction performance. In the study at hand the two learners favoring clinical features are, overall, the best performing of all complex methods under investigation (see *Overview of prediction performance* section). In the next section, this and the other outlined results are discussed and conclusions are drawn.

5 Discussion and Conclusion

The study at hand provides a large-scale benchmark experiment for prediction methods using multi-omics data. Eleven prediction methods and variants of them are compared based on their prediction performance, sparsity, and computation time on 18 cancer data sets. Among the methods compared are three Lasso-based methods (standard Lasso, two-step IPF-Lasso, two-step priority-Lasso), two boosting methods (likelihood-based boosting and model-based boosting with the implementations CoxBoost and Coxboost favoring, and glmboost), and two random forest variants (rfsrc and ranger). These methods make use of the multi-omics group structure differently.

Taking into account all of the different assessment dimensions and performance measures, there is no clearly best performing method. But likelihood-based boosting with the configuration favoring clinical features performs best according to prediction accuracy. It also leads to reasonable sparse models on group as well as on feature level and ranks in midfield based on computation time.

Moreover, on average and based on the cindex, the structured learners (learners favoring clinical features or learners using the whole multi-omics group structure) show better prediction performance than the naive learners (not using group structure at all). But the picture is ambiguous when regarding the ibrier. Furthermore, there is no method/learner that, averaged over all data sets, clearly outperforms the clinical learner. Again, likelihood-based boosting favoring clinical features differs from that, being ahead of the clinical learner based on the ibrier and comparable based on the cindex. Yet, for several data sets there are learners which outperform the clinical learner. This indicates that whether the molecular data add predictive value depends on the data set/cancer type. Also the used method and its configuration seem to be important.

Regarding sparsity, IPF-Lasso yields the sparsed results, both on feature as well as on group level, whereas priority-Lasso leads to the less sparsest models on feature level and does not select groups at all. The methods model-based boosting, likelihood-based boosting, and standard Lasso also lead to sparse models on feature level and group level but at the expense of smaller feature groups. Since random forest does not yield easily interpretable models, it was not assessed in this dimension.

Moreover, since comparability is only given when regarding methods that include the group structure the same way, the methods were additionally compared within the three dimensions of comparability: naive, favoring clinical features, and using multi-omics group structure.

Among the naive learners and according to the cindex, the random forest variants perform notably better than the other methods (standard Lasso, model-based boosting and likelihood-based boosting). Again, the picture is different when regarding the ibrier, but still only one other method (likelihood-based boosting) performs better than the random forest variants. Moreover, rfsrc needs the least amount of computation time (though it has to be taken into account that no tuning is conducted for the random forest variants). Focusing on the other naive methods, standard Lasso performs better than the boosting methods based on the cindex and worse based on the ibrier.

There are two learners favoring clinical features based on two different prediction methods. The learner *CoxBoost favoring* refers to the *likelihood-based boosting* method and the learner *prioritylasso favoring* to the method *priority-Lasso*, which represent two different modelling approaches (boosting and Lasso). *prioritylasso favoring* additionally uses the remaining group structure by further distinguishing the molecular features. Here, likelihood-based boosting (i.e. CoxBoost favoring) shows notably better prediction performance than priority-Lasso, performing better based on both measures. Moreover, it leads to sparser models with around 10 features, whereas priorty-

Lasso selects around 30 features on average. Finally, likelihood-based boosting favoring clinical features needs far less computation time. Overall, the two favoring methods are the best performing complex methods based on the cindex, and likelihood-based boosting also based on the ibrier.

The methods using the whole multi-omics group structure and not favoring clinical features are priority-Lasso and IPF-Lasso. Thus, only methods from the Lasso approach are investigated. In contrast to the methods/learners favoring clinical features, the findings are ambiguous. Admittedly, priority-Lasso outperforms IPF-Lasso based on the cindex, but they perform equally based on the ibrier. Moreover, IPF-Lasso is ranked higher than priority-Lasso based on the ibrier. Also, IPF-Lasso leads to clearly sparser models on feature as well as on group level. However, priority-Lasso needs less computation time.

Finally, the results suggest that, whether or not the molecular data are distinguished, the clinical features should be favored, since for likelihood-based boosting as well as for prioirty-Lasso the learners favoring clinical features outperform the corresponding non-favoring learners.

Concluding, the findings indicate that using multi-omics data for prediction may lead to better prediction performance, but that depends on the used method and its configuration, the data set and the way the multi-omics data is used. Naive methods, not using the group structure, yield overall poor prediction performance (with exception for some data sets where especially the random forest variants yield some very good results).

One limitation of the study is that only few methods have been investigated that include the multi-omics group structure (due to only few of such methods have yet been proposed). For example, likelihood-based boosting shows good performance when distinguishing clinical and molecular features. This is promising that further group information might additionally raise the performance. The same can be said for the random forest variants. Since they show by tendency the best performance among the naive learners and for some data sets even better performance than the structured learners, it would be very interesting to investigate whether implementations of these methods, which are able to use multi-omics group structure, lead to better performances. Not including sparse group Lasso, due to the fact that it was not possible to use it in such a high-dimensional setting, further limits the study.

Moreover, of the 26 data sets available only 23 data sets were suited for our purpose. Of these 23 data sets, 5 had to be excluded due to stability and computation time issues, so that the study is based on 18 data sets. Expanding the study to more multi-omics data sets might result in more clear-cut findings, where the study at hand draws an ambiguous picture.

Finally, the proportional hazards assumption underlies all of the methods used. Although an implementation of the cindex was chosen which is not prone to the model assumption, other underlying model assumptions are conceivable and it would be interesting to investigate their influence.

In addition to that, a collection with the described and other multi-omics data sets could be gathered to further improve comparability and increase the scope this benchmark study covers. This would make it possible to use hypothesis testing to compare the performance of the methods, which for example Boulesteix et al (2015) describe and is conducted by Couronné et al (2018).

Thus, future research could focus on new methods that are able to include multi-omics data or to adjust established methods such as random forest to be able to include multi-omics group structure information. Furthermore, methods which have been proposed to include multi-omics data, but which were not published in time to be included in the study, for example the method proposed by Velten and Huber (2018), could be assessed. The same holds true for sparse group Lasso, if the problems get fixed. The benchmark experiment presented here is designed in that way that an expansion can easily be achieved with the provided code.

References

Azzola, M. F., Shaw, H. M., Thompson, J. F., Soong, S., Scolyer, R. A., Watson, G. F., Colman, M. H., and Zhang, Y. (2003). “Tumor mitotic rate is a more powerful prognostic indicator than ulceration in patients with primary cutaneous melanoma: an analysis of 3661 patients from a single center”. In: *Cancer: Interdisciplinary International Journal of the American Cancer Society* 97.6, pp. 1488–1498.

Binder, H. (2013). *CoxBoost: Cox models by likelihood based boosting for a single survival endpoint or competing risks*. R package version 1.4. URL: `https://CRAN.R-project.org/package=CoxBoost`.

Binder, H., Allignol, A., Schumacher, M., and Beyersmann, J. (2009). “Boosting for high-dimensional time-to-event data with competing risks”. In: *Bioinformatics* 25.7, pp. 890–896.

Binder, H. and Schumacher, M. (2008). “Allowing for mandatory covariates in boosting estimation of sparse high-dimensional survival models”. In: *BMC Bioinformatics* 9.1.

Bischl, B., Lang, M., Kotthoff, L., Schiffner, J., Richter, J., Studerus, E., Casalicchio, G., and Jones, Z. M. (2016). “mlr: Machine Learning in R”. In: *Journal of Machine Learning Research* 17.170, pp. 1–5.

Bischl, B., Mersmann, O., Trautmann, H., and Weihs, C. (2012). “Resampling methods for meta-model validation with recommendations for evolutionary computation”. In: *Evolutionary Computation* 20.2, pp. 249–275.

Bischl, B., Schiffner, J., and Weihs, C. (2013). “Benchmarking local classification methods”. In: *Computational Statistics* 28.6, pp. 2599–2619.

Blaszczak, W., Barczak, W., Wegner, A., Golusinski, W., and Suchorska, W. M. (2017). “Clinical value of monoclonal antibodies and tyrosine kinase inhibitors in the treatment of head and neck squamous cell carcinoma”. In: *Medical Oncology* 34.4, p. 60.

Boulesteix, A.-L. (2013). “On representative and illustrative comparisons with real data in bioinformatics: response to the letter to the editor by Smith et al.” In: *Bioinformatics* 29.20, pp. 2664–2666.

Boulesteix, A.-L., De Bin, R., Jiang, X., and Fuchs, M. (2017a). “IPF-LASSO: Integrative L1-Penalized Regression with Penalty Factors for Prediction Based on Multi-Omics Data”. In: *Computational and Mathematical Methods in Medicine* 2017, pp. 1–14.

Boulesteix, A.-L. and Fuchs, M. (2015). *ipflasso: Integrative Lasso with Penalty Factors*. R package version 0.1. URL: `https://CRAN.R-project.org/package=ipflasso`.

Boulesteix, A.-L., Hable, R., Lauer, S., and Eugster, M. J. A. (2015). “A Statistical Framework for Hypothesis Testing in Real Data Comparison Studies”. In: *The American Statistician* 69.3, pp. 201–212.

Boulesteix, A.-L., Lauer, S., and Eugster, M. J. A. (2013). “A Plea for Neutral Comparison Studies in Computational Sciences”. In: *PLoS ONE* 8.4, e61562.

Boulesteix, A.-L. and Sauerbrei, W. (2011). “Added predictive value of high-throughput molecular data to clinical data and its validation”. In: *Briefings in Bioinformatics* 12.3, pp. 215–229.

Boulesteix, A.-L., Wilson, R., and Hapfelmeier, A. (2017b). “Towards evidence-based computational statistics: lessons from clinical research on the role and design of real-data benchmark studies”. In: *BMC Medical Research Methodology* 17.1.

Bøvelstad, H. M., Nygård, S., and Borgan, Ø. (2009). “Survival prediction from clinico-genomic models - a comparative study”. In: *BMC Bioinformatics* 10.1.

Breiman, L. (2001). “Random Forests”. In: *Machine Learning* 45.1, pp. 5–32.

Bruix, J., Reig, M., and Sherman, M. (2016). “Evidence-Based Diagnosis, Staging, and Treatment of Patients With Hepatocellular Carcinoma”. In: *Gastroenterology* 150.4, pp. 835–853.

Brulé, S. Y., Jonker, D. J., Karapetis, C. S., O’Callaghan, C. J., Moore, M. J., Wong, R., Tebbutt, N. C., Underhill, C., Yip, D., Zalcberg, J. R., Tu, D., and Goodwin, R. A. (2015). “Location of colon cancer (right-sided versus left-sided) as a prognostic factor and a predictor of benefit from cetuximab in NCIC CO.17”. In: *European Journal of Cancer* 51.11, pp. 1405–1414.

Bühlmann, P. and Hothorn, T. (2007). “Boosting Algorithms: Regularization, Prediction and Model Fitting”. In: *Statistical Science* 22.4, pp. 477–505.

Cash, T., McIlvaine, E., Krailo, M. D., Lessnick, S. L., Lawlor, E. R., Laack, N., Sorger, J., Marina, N., Grier, H. E., Granowetter, L., Womer, R. B., and DuBois, S. G. (2016). “Comparison of clinical features and outcomes in patients with extraskeletal versus skeletal localized Ewing sarcoma: A report from the Children’s Oncology Group”. In: *Pediatric Blood & Cancer* 63.10, pp. 1771–1779.

Claus, E. B., Walsh, K. M., Wiencke, J. K., Molinaro, A. M., Wiemels, J. L., Schildkraut, J. M., Bondy, M. L., Berger, M., Jenkins, R., and Wrensch, M. (2015). “Survival and low-grade glioma: the emergence of genetic information”. In: *Neurosurgical Focus* 38.1, E6.

Coroller, T. P., Grossmann, P., Hou, Y., Rios Velazquez, E., Leijenaar, R. T. H., Hermann, G., Lambin, P., Haibe-Kains, B., Mak, R. H., and Aerts, H. J. W. L. (2015). “CT-based radiomic signature predicts distant metastasis in lung adenocarcinoma”. In: *Radiotherapy and Oncology* 114.3, pp. 345–350.

Couronné, R., Probst, P., and Boulesteix, A.-L. (2018). “Random forest versus logistic regression: a large-scale benchmark experiment”. In: *BMC Bioinformatics* 19.1.

Cox, D. R. (1972). “Regression Models and Life-Tables”. In: *Journal of the Royal Statistical Society. Series B (Methodological)* 34.2, pp. 187–220.

De Bin, R. (2016). “Boosting in Cox regression: a comparison between the likelihood-based and the model-based approaches with focus on the R-packages CoxBoost and mboost”. In: *Computational Statistics* 31.2, pp. 513–531.

De Bin, R., Herold, T., and Boulesteix, A.-L. (2014a). “Added predictive value of omics data: specific issues related to validation illustrated by two case studies”. In: *BMC Medical Research Methodology* 14.1.

De Bin, R., Sauerbrei, W., and Boulesteix, A.-L. (2014b). “Investigating the prediction ability of survival models based on both clinical and omics data: two case studies”. In: *Statistics in Medicine* 33.30, pp. 5310–5329.

Escudier, B., Porta, C., Schmidinger, M., Rioux-Leclercq, N., Bex, A., Khoo, V., Gruenvald, V., and Horwich, A. (2016). “Renal cell carcinoma: ESMO Clinical Practice Guidelines for diagnosis, treatment and follow-up†”. In: *Annals of Oncology* 27.suppl_5, pp. v58–v68.

Fakhry, C., Westra, W. H., J., Wang S., Zante, A. van, Zhang, Y., Rettig, R., Yin, L. X., Ryan, W. R., Ha, P. K., Wentz, A., Koch, W., Richmon, J. D., Eisele, D. W., and D’Souza, G. (2017). “The prognostic role of sex, race, and human papillomavirus in oropharyngeal and nonoropharyngeal head and neck squamous cell cancer”. In: *Cancer* 123.9, pp. 1566–1575.

Fraser, M., Berlin, A., Bristow, R. G., and Kwast, T. van der (2015). “Genomic, pathological, and clinical heterogeneity as drivers of personalized medicine in prostate cancer”. In: *Urologic Oncology: Seminars and Original Investigations* 33.2, pp. 85–94.

Friedman, J. H. (2001). “Greedy Function Approximation: A Gradient Boosting Machine”. In: *The Annals of Statistics* 29.5, pp. 1189–1232.

Friedman, J. H., Hastie, T., and Tibshirani, R. (2000). “Additive logistic regression: a statistical view of boosting (With discussion and a rejoinder by the authors)”. In: *The Annals of Statistics* 28.2, pp. 337–407.

— (2010). “Regularization Paths for Generalized Linear Models via Coordinate Descent”. In: *Journal of Statistical Software* 33.1, pp. 1–22.

Gerds, T. A., Cai, T., and Schumacher, M. (2008). “The Performance of Risk Prediction Models”. In: *Biometrical Journal* 50.4, pp. 457–479.

Gerds, T. A., Kattan, M. W., Schumacher, M., and Yu, C. (2013). “Estimating a time-dependent concordance index for survival prediction models with covariate dependent censoring”. In: *Statistics in Medicine* 32.13, pp. 2173–2184.

Graf, E., Schmoor, C., Sauerbrei, W., and Schumacher, M. (1999). “Assessment and comparison of prognostic classification schemes for survival data”. In: *Statistics in Medicine* 18.17-18, pp. 2529–2545.

Hasin, Y., Seldin, M., and Lusis, A. (2017). “Multi-omics approaches to disease”. In: *Genome Biology* 18.1, pp. 83–98.

Hastie, T., Tibshirani, R., and Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction, 2nd Edition.* Springer Series in Statistics. New York: Springer.

Hofner, B., Mayr, A., Robinzonov, N., and Schmid, M. (2014). “Model-based boosting in R: a hands-on tutorial using the R package mboost”. In: *Computational Statistics* 29.1-2, pp. 3–35.

Hong, H. G., Chen, X., Christiani, D. C., and Li, Y. (2018). “Integrated powered density: Screening ultrahigh dimensional covariates with survival outcomes”. In: *Biometrics* 74.2, pp. 421–429.

Hothorn, T., Buehlmann, P., Kneib, T., Schmid, M., and Hofner, B. (2018). *mboost: Model-Based Boosting.* R package version 2.9-0. URL: `https://CRAN.R-project.org/package=mboost`.

Ishwaran, H. and Kogalur, U. B. (2018). *Random Forests for Survival, Regression, and Classification (RF-SRC)*. R package version 2.6.1. manual. URL: `https://cran.r-project.org/package=randomForestSRC`.

Ishwaran, H., Kogalur, U. B., Blackstone, E. H., and Lauer, M. S. (2008). "Random survival forests". In: *The Annals of Applied Statistics* 2.3, pp. 841–860.

Joniau, S., Briganti, A., Gontero, P., Gandaglia, G., Tosco, L., Fieuws, S., Tombal, B., Marchioro, G., Walz, J., Kneitz, B., Bader, P., Frohneberg, D., Tizzani, A., Graefen, M., Cangh, P. van, Karnes, R. J., Montorsi, F., Van Poppel, H., and Spahn, M. (2015). "Stratification of High-risk Prostate Cancer into Prognostic Categories: A European Multi-institutional Study". In: *European Urology* 67.1, pp. 157–164.

Kim, K.-J., Kim, S.-M., Lee, Y. S., Chung, W. Y., Chang, H.-S., and Park, C. S. (2015). "Prognostic significance of tumor multifocality in papillary thyroid carcinoma and its relationship with primary tumor size: a retrospective study of 2,309 consecutive patients". In: *Annals of surgical oncology* 22.1, pp. 125–131.

Klau, S. and Hornung, R. (2017). *prioritylasso: Analyzing Multiple Omics Data with an Offset Approach*. R package version 0.2.1. URL: `https://CRAN.R-project.org/package=prioritylasso`.

Klau, S., Jurinovic, V., Hornung, R., Herold, T., and Boulesteix, A.-L. (2018). "Priority-Lasso: a simple hierarchical approach to the prediction of clinical outcome using multi-omics data". In: *BMC Bioinformatics* 19.1.

Lang, M., Bischl, B., and Surmann, D. (2017). "batchtools: Tools for R to work on batch systems". In: *The Journal of Open Source Software* 2.10.

Lang, M., Kotthaus, H., Marwedel, P., Weihs, C., Rahnenführer, J., and Bischl, B. (2015). "Automatic model selection for high-dimensional survival analysis". In: *Journal of Statistical Computation and Simulation* 85.1, pp. 62–76.

Mayr, A., Binder, H., Gefeller, O., and Schmid, M. (2014). "The Evolution of Boosting Algorithms". In: *Methods of Information in Medicine* 53.06, pp. 419–427.

Mayr, A., Hofner, B., Waldmann, E., Hepp, T., Meyer, S., and Gefeller, O. (2017). "An Update on Statistical Boosting in Biomedicine". In: *Computational and Mathematical Methods in Medicine* 2017, pp. 1–12.

Microsoft Corporation (2018). *checkpoint: Install Packages from Snapshots on the Checkpoint Server for Reproducibility.* R package version 0.4.5. URL: `https://CRAN.R-project.org/package=checkpoint`.

Morice, P., Leary, A., Creutzberg, C., Abu-Rustum, N., and Darai, E. (2016). "Endometrial cancer". In: *The Lancet* 387.10023, pp. 1094–1108.

Panici, P. B. et al (2014). "Secondary analyses from a randomized clinical trial: age as the key prognostic factor in endometrial carcinoma". In: *American Journal of Obstetrics and Gynecology* 210.4, 363.e1–363.e10.

Pignatti, F., Bent, M. van den, Curran, D., Debruyne, C., Sylvester, R., Therasse, P., Áfra, D., Cornu, P., Bolla, M., Vecht, C., and Karim, A. B. M. F. (2002). "Prognostic Factors for Survival in Adult Patients With Cerebral Low-Grade Glioma". In: *Journal of Clinical Oncology* 20.8, pp. 2076–2084.

Probst, P., Wright, M., and Boulesteix, A.-L. (2018). "Hyperparameters and Tuning Strategies for Random Forest". In: *arXiv preprint arXiv:1804.03515*.

R Core Team (2018). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing. Vienna, Austria. URL: `https://www.R-project.org/`.

Safieddine, N., Liu, G., Cuningham, K., Ming, T., Hwang, D., Brade, A., Bezjak, A., Fischer, S., Xu, W., Azad, S., Cypel, M., Darling, G., Yasufuku, K., Pierre, A., Perrot, M. de, Waddell, T., and Keshavjee, S. (2014). "Prognostic factors for cure, recurrence and long-term survival after surgical resection of thymoma". In: *Journal of Thoracic Oncology* 9.7, pp. 1018–1022.

Schnack, T. H., Høgdall, E., Nedergaard, L., and Høgdall, C. (2016). "Demographic Clinical and Prognostic Factors of Primary Ovarian Adenocarcinomas of Serous and Clear Cell Histology—A Comparative Study". In: *International Journal of Gynecological Cancer* 26.1, pp. 82–90.

Schulze, G. (2017). "Clinical Outcome Prediction Based on Multi-Omics Data: Extension of IPF-LASSO". MA thesis. Munich: Ludwig-Maximilians-University. Department of Statistics.

Seibold, H., Bernau, C., Boulesteix, A.-L., and De Bin, R. (2018). "On the choice and influence of the number of boosting steps for high-dimensional linear Cox-models". In: *Computational Statistics* 33.3, pp. 1195–1215.

Shapiro, J., Klaveren, D. van, Lagarde, S. M., Toxopeus, E. L. A., Gaast, A. van der, Hulshof, M. C. C. M., Wijnhoven, B. P. L., Berge Henegouwen, M. I. van, Steyerberg, E. W., and Lanschot, J. J. B. van (2016). "Prediction of survival in patients with oesophageal or junctional cancer receiving neoadjuvant chemoradiotherapy and surgery". In: *British Journal of Surgery* 103.8, pp. 1039–1047.

Simon, N., Friedman, J. H., Hastie, T., and Tibshirani, R. (2011). "Regularization Paths for Cox's Proportional Hazards Model via Coordinate Descent". In: *Journal of Statistical Software* 39.5, pp. 1–13.

— (2013a). "A Sparse-Group Lasso". In: *Journal of Computational and Graphical Statistics* 22.2, pp. 231–245.

— (2013b). *SGL: Fit a GLM (or cox model) with a combination of lasso and group lasso regularization.* R package version 1.1. URL: `https://CRAN.R-project.org/package=SGL`.

Smith, R., Ventura, D., and Prince, J. T. (2013). "Novel algorithms and the benefits of comparative validation". In: *Bioinformatics* 29.12, pp. 1583–1585.

Steyerberg, E. W., Vickers, A. J., Cook, N. R., Gerds, T., Gonen, M., Obuchowski, N., Pencina, M. J., and Kattan, M. W. (2010). "Assessing the Performance of Prediction Models". In: *Epidemiology* 21.1, pp. 128–138.

Sutton, M, Thiébaut, R., and Liquet, B. (2018). "Sparse partial least squares with group and subgroup structure". In: *Statistics in Medicine* 37.23, pp. 3338–3356.

Szász, A. M., Lánczky, A., Nagy, Á., Förster, S., Hark, K., Green, J. E., Boussioutas, A., Busuttil, R., Szabó, A., and Győrffy, B. (2016). "Cross-validation of survival associated biomarkers in gastric cancer using transcriptomic data of 1,065 patients". In: *Oncotarget* 7.31, pp. 49322–49333.

Teramoto, Y., Keim, U., Gesierich, A., Schuler, G., Fiedler, E., Tüting, T., Ulrich, C., Wollina, U., Hassel, J. C, Gutzmer, R., Goerdt, S., Zouboulis, C., Leiter, U., Eigentler, T. K., and Garbe, C. (2018). "Acral lentiginous melanoma: a skin cancer with unfavourable prognostic features. A study of the German central malignant melanoma registry (CMMR) in 2050 patients". In: *British Journal of Dermatology* 178.2, pp. 443–451.

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

Tibshirani, R. (1996). "Regression shrinkage and selection via the lasso". In: *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288.

— (1997). "The lasso method for variable selection in the Cox model". In: *Statistics in Medicine* 16.4, pp. 385–395.

Tutz, G. and Binder, H. (2006). "Generalized Additive Modeling with Implicit Variable Selection by Likelihood-Based Boosting". In: *Biometrics* 62.4, pp. 961–971.

Uno, H., Cai, T., Pencina, M. J., D'Agostino, R. B., and Wei, L. J. (2011). "On the C-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data". In: *Statistics in Medicine* 30.10, pp. 1105–1117.

Velten, B. and Huber, W. (2018). "Adaptive penalization in high-dimensional regression and classification with external covariates using variational Bayes". In: *arXiv preprint arXiv:1811.02962*.

Weis, C.-A., Yao, X., Deng, Y., Detterbeck, F. C., Marino, M., Nicholson, A. G., Huang, J., Ströbel, P., Antonicelli, A., and Marx, A. (2015). "The Impact of Thymoma Histotype on Prognosis in a Worldwide Database". In: *Journal of Thoracic Oncology* 10.2, pp. 367–372.

Wiel, M. A. van de, Lien, T. G., Verlaat, W., Wieringen, W. N. van, and Wilting, S. M. (2015). "Better prediction by use of co-data: adaptive group-regularized ridge regression". In: *Statistics in Medicine* 35.3, pp. 368–381.

Wright, M. N. and Ziegler, A. (2017). "ranger: A Fast Implementation of Random Forests for High Dimensional Data in C++ and R". In: *Journal of Statistical Software* 77.1, pp. 1–17.

Yang, C.-F. J., Wang, H., Kumar, A., Wang, X., Hartwig, M. G., D'Amico, T. A., and Berry, M. F. (2017). "Impact of Timing of Lobectomy on Survival for Clinical Stage IA Lung Squamous Cell Carcinoma". In: *Chest* 152.6, pp. 1239–1250.

Yokota, T., Ando, N., Igaki, H., Shinoda, M., Kato, K., Mizusawa, J., Katayama, H., Nakamura, K., Fukuda, H., and Kitagawa, Y. (2015). "Prognostic Factors in Patients Receiving Neoadjuvant 5-Fluorouracil plus Cisplatin for Advanced Esophageal Cancer (JCOG9907)". In: *Oncology* 89.3, pp. 143–151.

Yuan, M. and Lin, Y. (2006). "Model selection and estimation in regression with grouped variables". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 68.1, pp. 49–67.

Zhao, Q., Shi, X., Xie, Y., Huang, J., Shia, B., and Ma, S. (2014). "Combining multidimensional genomic measurements for predicting cancer prognosis:

observations from TCGA". In: *Briefings in Bioinformatics* 16.2, pp. 291–303.

Zou, H. (2006). "The Adaptive Lasso and Its Oracle Properties". In: *Journal of the American Statistical Association* 101.476, pp. 1418–1429.

A Figures

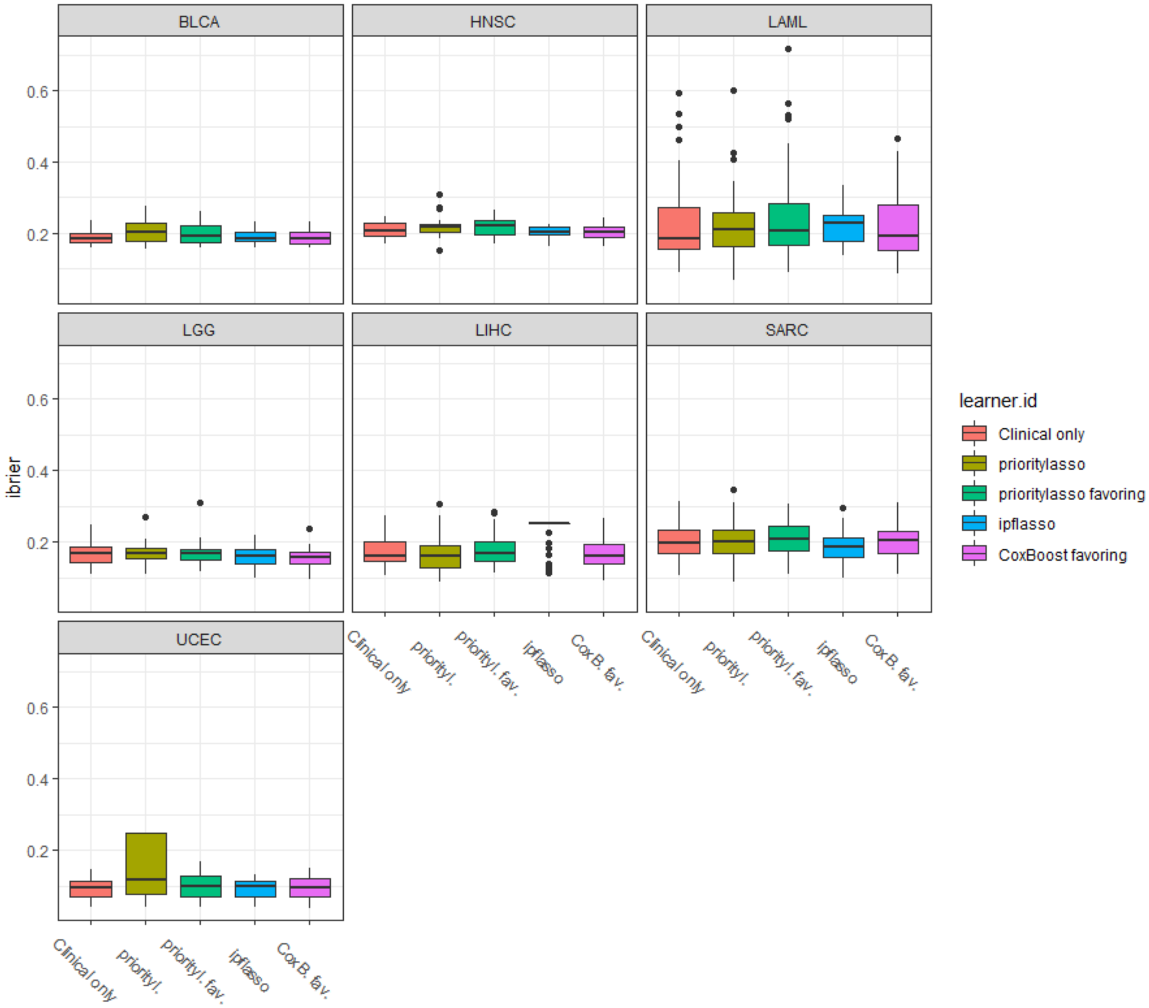


Figure 10: ibrier for the structured learners and the clinical learner as a reference. Depicted are only those data sets where the best learner performs better on both measures.

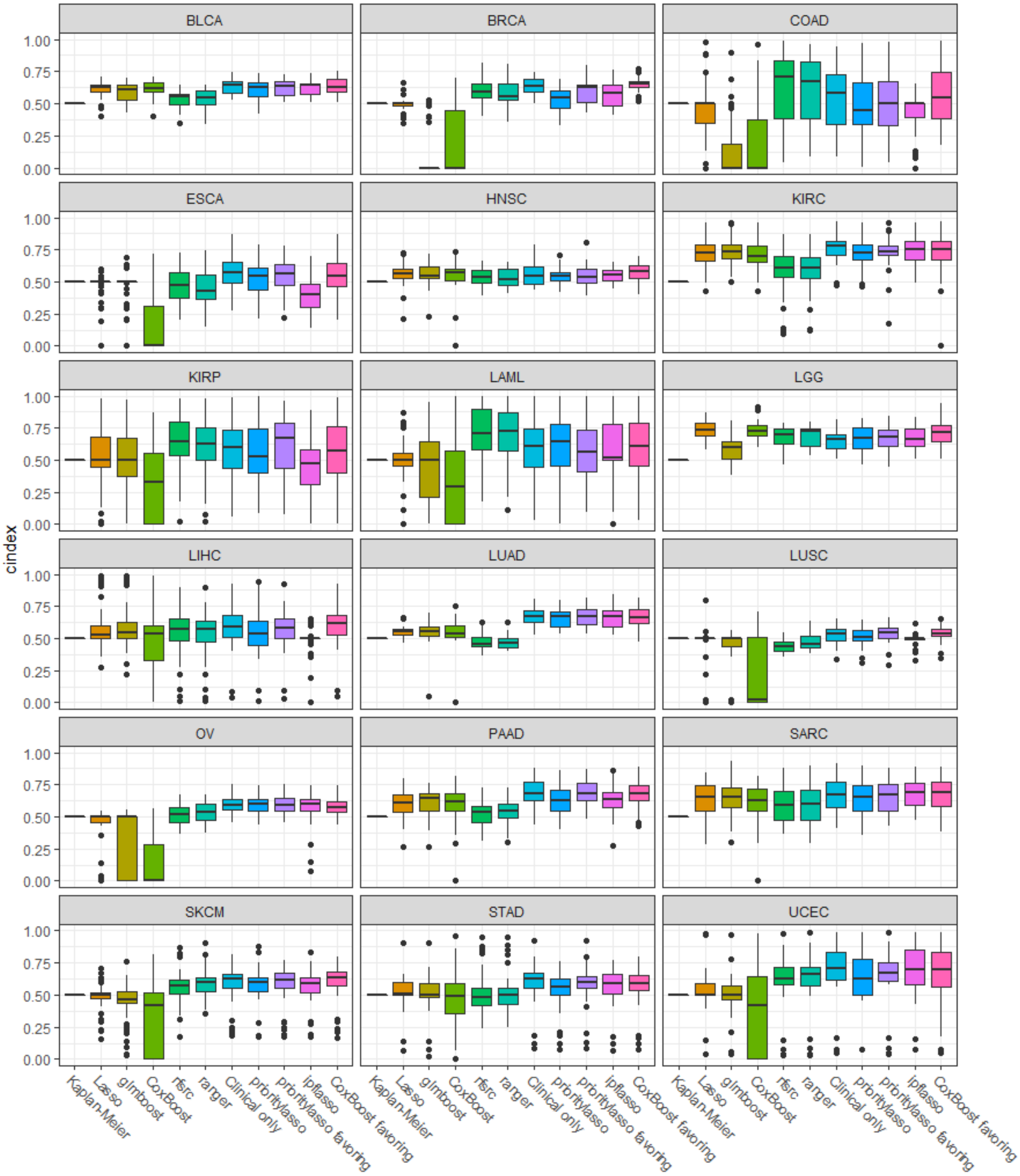


Figure 11: Prediction performance based on the cindex by data set.

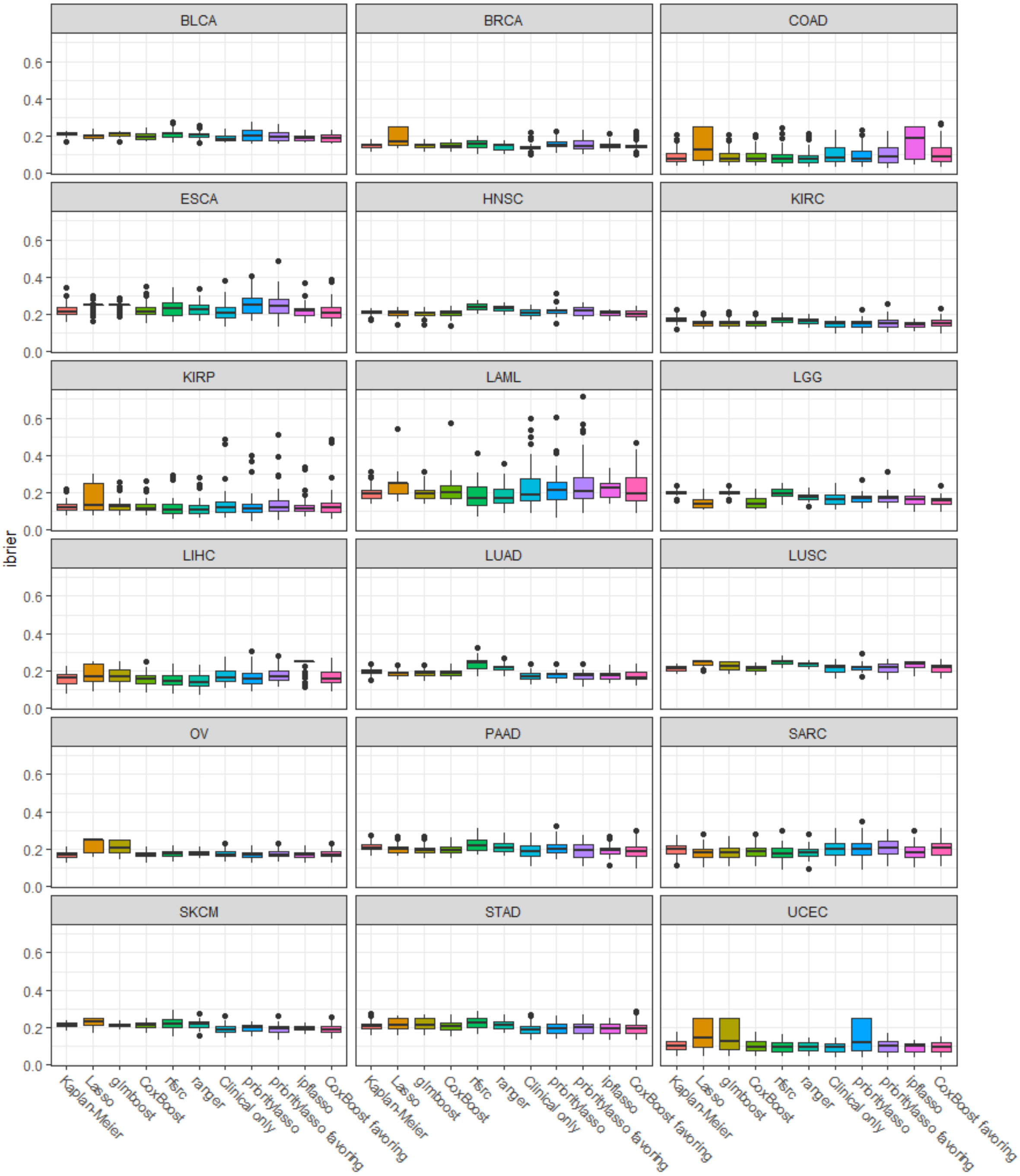


Figure 12: Prediction performance based on the ibrier by data set.

B Tables

BLCA	Bladder Urothelial
BRCA	Breast Invasive Carcinoma
CESC	Cervical Squamous Cell Carcinoma and Endocervical Adenocarcinoma
COAD	Colon Adenocarcinoma
ESCA	Esophageal Carcinoma
GBM	Glioblastoma Multiforme
HNSC	Head-Neck Squamous Cell Carcinoma
KIRC	Kidney Renal Clear Cell Carcinoma
KIRP	Cervical Kidney Renal Papillary Cell Carcinoma
LAML	Acute Myeloid Leukemia
LGG	Low Grade Glioma
LIHC	Liver Hepatocellular Carcinoma
LUAD	Lung Adenocarcinoma
LUSC	Lung Squamous Cell Carcinoma
OV	Ovarian Cancer
PAAD	Pancreatic Adenocarcinoma
PCPG	Pheochromocytoma and Paragangliom
PRAD	Prostate Adenocarcinoma
READ	Rectum Adenocarcinoma
SARC	Sarcoma
SKCM	Skin Cutaneous Melanoma
STAD	Stomach Adenocarcinoma
TGCT	Testicular Cell Tumor
THCA	Thyroid Cancer
THYM	Thymoma
UCEC	Uterine Corpus Endometrial Carcinoma

Table 13: Overview of cancer types and the reference abbreviations

The following tables provide details of the clinical features of each cancer type/data set. In the first column of the tables the cancer type and the corresponding reference used to identify relevant clinical features can be found. Furthermore, the first column shows the original number of effective cases before any preprocessing is conducted. The eventually resulting amount of effective cases after preprocessing and merging clinical and molecular data (in percentage of the original amount) is displayed in parentheses. The second and third column show the included and preprocessed features and the feature type (bin = binary, num = numerical, int = integer, fac = factor).

Cancer	Features	Type
BLCA	age	num
108 (95%)	sex	bin
Binder et al (2009)	diagnosis subtype Papillary	num
	pathologic stage	fac (3 levels)
BRCA	age	num
104 (69%)	no. of lymphnodes positive by he	int
Boulesteix et al (2017a)	histological type	bin
	estrogen receptor status	bin
	progesterone receptor status	bin
	surgical procedure	fac (4 levels)
COAD	age	num
56 (30%)	sex	bin
Brulé et al (2015)	no. of lymphnodes positive by he	int
	pathologic stage	fac (3 levels)
	venous invasion	bin
	lymphatic invasion	bin
ESCA	age	num
57 (65%)	sex	bin
Yokota et al (2015)	pathologic stage	fac (3 levels)
Shapiro et al (2016)	pathology histological type	bin
	pathology residual tumor	bin
HNSC	age	num
170 (89%)	sex	bin
Fakhry et al (2017)	clinical stage	fac (4 levels)
Blaszczak et al (2017)	histologic grade	fac (4 levels)
	alcohol history	bin
	lymphnode neck dissection	bin

Table 14: Clinical features: BLCA, BRCA, COAD, ESCA, HNSC

Cancer	Features	Type
KIRC	age	num
162 (38%)	sex	bin
Escudier et al (2016)	laterality	bin
	hemoglbin result	bin
	white cell count result	bin
	histologic grade	fac (4 levels)
	pathologic stage	bin
KIRP	age	num
32 (63%)	sex	bin
Schulze (2017)	laterality	bin
	white cell count result	bin
	hemoglobin result	bin
	pathologic stage	bin
LAML	age	num
120 (12%)	sex	bin
Boulesteix et al (2017a)	leukocyte result	fac (levels 3)
	morphology code	fac (levels 4)
LGG	age	num
92 (84%)	sex	bin
Pignatti et al (2002)	histological type	fac (levels 3)
Claus et al (2015)	laterality	bin
	visual changes	bin
	sensory changes	bin
	motor movement changes	bin
	tumor location	fac (levels 3)

Table 15: Clinical features: KIRC, KIRP, LAML, LGG

Cancer	Features	Type
LIHC	age	num
91 (38%)	sex	bin
Bruix et al (2016)	albumin value	num
	creatinine value	num
	fetoprotein value	int
	pathologic stage	fac (levels 3)
	vascular tumor cell type	fac (levels 3)
	fibrosis ishak score	fac (levels 3)
LUAD	age	num
123 (82%)	sex	bin
Coroller et al (2015)	tobacco smoking history	int
	pathologic stage	fac (4 levels)
	anatomic neoplasm subdivision	fac (4 levels)
LUSC	age	num
157 (84%)	sex	bin
Yang et al (2017)	tobacco smoking history	int
	pathologic stage	fac (4 levels)
	anatomic neoplasm subdivision	fac (4 levels)
OV	age	num
301 (36%)	clinical stage	fac (3 levels)
Schnack et al (2016)	tumor residual disease	fac (3 levels)
	anatomic neoplasm subdivision	bin

Table 16: Clinical features: LIHC, LUAD, LUSC, OV

Cancer	Features	Type
PAAD	age	num
66 (78%)	sex	bin
	maximum tumor dimension	num
	no. of lymphnodes positive by he	int
	tobacco smoking history	int
	anatomic neoplasm subdivision	bin
	histological type	bin
	histological grade	bin
	pathologic stage	bin
	surgery performed	bin
PRAD	age	num
8 (88%)	gleason grading	int
Joniau et al (2015)	psa	num
Fraser et al (2015)	residual tumor	bin
SARC	age	num
76 (50%)	sex	bin
Cash et al (2016)	histological type	fac (4 levels)
	metastatic diagnosis	bin
	radiation therapy	bin
	tumor tissue sites	fac (3 levels)
	tumor total necrosis percent	fac (3 levels)
SKCM	age	num
154 (56%)	sex	bin
Teramoto et al (2018)	breslow depth value	num
Azzola et al (2003)	melanoma ulceration	bin
	pathologic stage	fac (4 levels)
	tumor tissue site	fac (3 levels)

Table 17: Clinical features: PAAD, PRAD, SARC, SKCM

Cancer	Features	Type
STAD	age	num
85 (73%)	sex	bin
Szász et al (2016)	no. of lymphnodes positive by he	int
	histologic grade	bin
	pathologic stage	fac (4 levels)
THCA	age	num
14 (86%)	sex	bin
Kim et al (2015)	extrathyroid carcinoma present extension	bin
	histological type	bin
	thyroid gland neoplasm location	fac (3 levels)
	pathologic stage	bin
THYM	age	num
6 (100%)	sex	bin
Safieddine et al (2014)	histological type	fac (5 levels)
Weis et al (2015)	masaoka stage	fac (4 levels)
	history myasthenia gravis	bin
UCEC	age	num
45 (84%)	weight	int
Panici et al (2014)	pct tumor invasion	num
Morice et al (2016)	weight	int
	total aor lnr	int
	total pelv lnr	int
	histological type	bin
	histological grade	fac (3 levels)
	clinical stage	fac (3 levels)
	surgical approach	bin

Table 18: Clinical features: STAD, THCA, THYM, UCEC

C Electronic appendix

The electronic appendix comprises three folders, which contain all the code and data to reproduce the results. First of all, the most important folder is *Results* containing the file *ergebnis.RData*, which holds the benchmark results. All figures, tables and findings presented in the study are based on this file.

Run *reproduce_results_Tables_and_Figure_preprocessing.R* to reproduce the results. Make sure that all packages listed in *packages.R* are installed and loaded. Furthermore, it is necessary to set the *wd* variable to the directory where the *electronic appendix* folder is located. The script computes all necessary tables. Based on the tables, *reproduce_results_Figures.R* reproduces the figures. The file *reproduce_results_ancillary_code.R* holds additional code needed for the computations and gets sourced automatically.

Since the first two figures are based on the raw data, these are treated in an extra script: *produce_figures_1_and_2.R*. Set the *wd* variable as before. Since all molecular data sets must be in RAM simultaneously to produce the figures, computers with small RAM might not be able to do the computations.

If the *ergebnis.RData* file itself should be reproduced, i.e the benchmark experiment shall be repeated, the folder *Benchmark experiment* holds the code and preprocessed data to do so. Though, it must be emphasised that the computations are very time consuming. On 12 kernels and 32 GB RAM the computation lasted around two weeks.

Make sure all packages in *packages.R* in the *sources* sub-folder of the *bench_code* folder are installed and loaded. *sources* also holds the script *ancillary_code.R*, where the methods especially implemented for this study are defined. It is crucial that the right version of the *mlr* package is installed (which can either be downloaded from GitHub (s. code) or is contained in the *mlr* folder). To

reproduce the experiment run the *bench_experiment.R* file in the *bench_code* folder. It is necessary to set the cluster function for parallelisation according to the system (default is Linux). Moreover, the *dir* variable must be set to the path leading to the *bench_code* folder.
The folder *data* in *bench_code* contains the data sets and the folder *resample_instances* the indices of each cancer type's repeated CV instances. With this it is possible to reproduce single method results or to expand the experiment to not used methods. For that to happen, *mlr* can be used by supplying the resample instance objects to a new benchmark experiment (via the *benchmark* function). It is also possible to use methods not covered by *mlr* by splitting the data sets according to the resample instance objects by hand.

Finally, the *Raw data and preprocessing* folder contains the raw data and the R scripts which can be used to produce the final data sets and to comprehend the preprocessing of the clinical data. The most important file is the *create_final_data_sets.R* script. By running this script (after adjusting the directory) it reproduces the final data sets, which will be stored in the empty folder *final_data*. The other scripts necessary to preprocess the clinical data and merge clinical and molecular data get sourced in the right order. The *function_* scripts hold help functions, which carry out specific tasks in this process. If there is special interest to comprehend a specific step, run *create_final_data_sets.R* until that step and execute the rest manually:
The *select_clinical_features.R* script selects the defined clinical features, *preprocess_clinical_data_pre_merge.R* and *preprocess_clinical_data_post_merge.R* execute the preprocessing of the clinical data before and after merging the clinical and the molecular data (e.g. merging factor levels). The clinical and the joined molecular data sets get merged with *merge_clinicals_and_moleculars.R*. *TCGA_Datasets* contains all raw data, *TCGA_Datasets_joined* the joined molecular data and *TCGA_Datasets_clinical* the raw clinical data.