MASTER'S THESIS

Penalized regression approaches for prognostic modelling using multi-omics data with block-wise missing values

Author: Jonas Hagenberg

Supervisor: Prof. Dr. rer. nat. HDR Anne-Laure Boulesteix

Institut für Statistik

Fakultät für Mathematik, Informatik und Statistik

Ludwig-Maximilians-Universität München

02.06.2020

Abstract

XXX

Contents

List of Figures

List of Tables

1 Introduction

Since the beginning of this century, the cost of analysing genomic data has dramatically fallen [Caulfield et al., 2013]. At the same time, new methods have been developed to not only decipher DNA, but also other information stored in organisms. The advent of the so-called next generation sequencing (NGS) led to the development of cost-effective methods to analyse e.g. how strongly a gene is active in cells (RNA-Seq), if a gene is mutated at a specific position (single nucleotide polymorphism (SNP)) or if a specific DNA base is methylated (a modification that does not alter the genomic information, which is part of the field called epigenetics). Together with methods to measure the protein content of cells or metabolism products, the results of these methods are all called "omics" data. Omics data is usually high-dimensional as higher organisms have thousands of genes and proteins, respectively.

As omics data is a good representation of the biological state of an organism, it is not only used in research but increasingly in the diagnosis and treatment of diseases, too. Even though the same disease leads to similar symptoms in all patients, the underlying disease mechanisms can be different, for example in colorectal cancer [Markowitz and Bertagnolli, 2009]. Using the molecular information from omics data might allow for the tailoring of a personalised treatment for every patient, based on the actual disease mechanism. To achieve this, it is common to use more than one omics method. The resulting data is called multi-omics data.

Multi-omics data leads to some specific challenges. As already mentioned, the data is high-dimensional. Since generating this data is still more expensive than established routine diagnostics (e.g. blood parameters), usually the number of cases for which the data is measured is small compared to the number of covariates. This requires some sort of regularisation when using the data for predictive modeling. Additionally, the correlation structure can be rather complex. Several omics methods can measure the same biological process and therefore contain overlapping information. This leads to the question which methods are actually necessary for a good prediction.

Klau et al. [2018] introduced a penalized regression method called priority-Lasso to address these problems. It uses a Lasso regression fitted to each data set – called block – from the different methods. To include prior knowledge about the importance of each block in regard to the prediction task, the user can choose the order of the fitted blocks. The prediction of each block is then used as the offset for the following block. This means that the user selects what they think is the most important method for the prediction. The following blocks are then only used to improve the prediction.

Another problem of multi-omics data is missing data. One category of missing data is when single values are missing. However, a more prominent problem of multi-omics data is

that the values of complete blocks are missing for some observations (block-wise missing data), which can be due to several reasons. Often, studies are conducted at multiple sites. The technology or funding for one or some of the methods can be missing at some sites, leading to missing blocks. Another reason is that over time, new technologies are developed and hence these blocks are missing in older samples.

In this thesis, I extended priority-Lasso to be able to deal with block-wise missing data. In section 2, I give an overview about multi-omcis data and explain how priority-Lasso works. Then I describe the problem of missing data, especially block-wise missing values and how existing methods deal with this type of missingness. In section 3, I explain how I adapted existing approaches of how to deal with block-wise missing data to priority-Lasso and improved its functionality. In section 4, I describe how I conducted a simulation experiment to investigate priority-Lasso's ability to deal with block-wise missing values. Then I applied priority-Lasso to a real world data set from asthma research, the results are given in section 5. Lastly I discuss and summarise the results in section 6.

2 Background and existing methods

In this section I discuss how multi-omics data are used in predictive modeling, with a focus on priority-Lasso. After explaining the Lasso-based method, I outline the problem of missing data, especially block-wise missing values. Then I give an overview about existing methods to deal with this type of missingness and how they can be adapted to priority-Lasso.

2.1 Multi-omics data for prediction modeling

A living organism is a complex system. The building block of this system is a cell. All cells follow the central dogma of molecular biology: the genetic information is stored in the DNA, and if a gene is active, it is transcribed into RNA. This RNA is then translated into a protein which has an effect in the cell [Crick, 1970]. All these steps are heavily regulated, interact with each other and have additional layers of modifications [Huang et al., 2017]. Due to the complex nature of this system, errors in different steps or layers can lead to a disease like cancer [Markowitz and Bertagnolli, 2009].

Therefore, it is paramount to analyse data sets from different sources to better understand the biology of a disease and to find biomarkers [Huang et al., 2017]. These different sources can comprise the complete genomic information (genomics), the information which genes are active in the cell (transcriptomics) or which genes are methylated in the DNA (epigenomics). Because of their suffix, these different data types are called "omics". When different omics data sets are analysed together, this is called multi-omics. Analyses can be unsupervised, for example to find different subgroups of disease mechanisms or immune reactions to a cancer [Thorsson et al., 2018]. Another application is supervised predictive modelling to e.g. predict individual survival for patients [Klau et al., 2018].

One common problem of different omics data sets is their high dimensionality. For example, the transcriptomics can comprise the complete expressed human genes, estimated at around 20000 different genes [Pertea and Salzberg, 2010]. As studies often consist of fewer patients than the number of covariates, one needs appropriate methods to deal with this dimensionality problem. Even though the costs of generating omics data are decreasing, multi-omics based tests are seldom used in the clinic [Karczewski and Snyder, 2018]. One important aspect is the easy interpretability of the results for physicians [Karczewski and Snyder, 2018], which can be facilitated by supplying sparse models. Another problem is that it is still not clear how to best combine possibly correlated multi-omics data sets [Boulesteix et al., 2017]. So far, a plethora of different methods of how to deal with this integration [Huang et al., 2017] has been developed. In the following section, I explain how the Lasso-based method priority-Lasso developed by Klau et al. [2018] can provide

multi-omics integration with a parsimonious model.

2.2 Priority-Lasso

The method used in this work to make predictions with multi-omics data is called priority-Lasso. In section 2.2.1 I explain the underlying Lasso methodology, in section 2.2.2 I detail the extensions made to Lasso which lead to priority-Lasso.

2.2.1 Lasso method

Lasso stands for *least absolute shrinkage and selection operator* and was introduced in 1996 by Tibshirani [1996]. It is an extension of the least squares estimator in linear regression and constitutes one type of a technique called regularisation [Fahrmeir et al., 2013]. The following description of the classical linear model and its problems is based on Fahrmeir et al. [2013]. In the classical linear regression model, one tries to solve the problem

$$\hat{\beta}_0, \hat{\boldsymbol{\beta}} = \underset{\beta_0, \boldsymbol{\beta}}{\arg\min} \|\mathbf{y} - \beta_0 \mathbb{1} - \mathbf{X}\boldsymbol{\beta}\|_2^2 . \tag{1}$$

Here, $\mathbf{X}$ is a $n \times p$ matrix that represents n observations with p covariates. Every observation has a continuous outcome, depicted in the $n \times 1$ vector $\mathbf{y}$. The goal is to find an estimate for $\boldsymbol{\beta}$, a $p \times 1$ vector, and β_0. $\boldsymbol{\beta}$ are the parameters of a linear function that uses the covariates to explain the systematic component in the outcome. β_0 is the intercept and $\mathbb{1}$ is a $n \times 1$ vector of ones. The goodness of fit – how well the linear function $\mathbf{X}\hat{\boldsymbol{\beta}} + \beta_0 \mathbb{1}$ approximates the outcome $\mathbf{y}$ – is measured by the squared distance between the true outcome and the approximated outcome.

However, an estimation is not possible in all cases. If the number of observations is smaller than the number of covariates, $\boldsymbol{\beta}$ cannot be estimated. This is due to the fact that the matrix $\mathbf{X}^\intercal\mathbf{X}$ has no inverse, which is needed for the estimation of $\boldsymbol{\beta}$. Another case is if there is collinearity in the data, which means that several covariates are (strongly) correlated. This can lead to numerical problems when solving for $\hat{\boldsymbol{\beta}}$.

Also, in some cases it is beneficial to have variable selection, i.e. only the parameters of the important variables are estimated with a value different from 0.

To address these problems, Tibshirani [1996] introduced the Lasso which can be written as

$$\hat{\beta}_0, \hat{\boldsymbol{\beta}} = \underset{\beta_0, \boldsymbol{\beta}}{\arg\min} \|\mathbf{y} - \beta_0 \mathbb{1} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \sum_{j=1}^{p} |\beta_j| \tag{2}$$

[see Fahrmeir et al., 2013]. Here, β_j is the jth element of $\boldsymbol{\beta}$ and the absolute values of the parameters (except from the intercept) are added as an additional penalty to the

measure how well the model fits the data. Because having (absolute) big parameter values is penalised, the parameters are shrunk towards 0 and some of the parameters are estimated as 0. Also, it is possible to get estimations for $\hat{\boldsymbol{\beta}}$ in the case of $p > n$ [Tibshirani, 1996].

In equation 2, λ is a hyperparameter with $\lambda \in [0, \infty)$ [Friedman et al., 2010]. A hyperparameter is a parameter that is not learned from the data, but has to be chosen before the model is run [Probst et al., 2019]. This necessitates that an optimal λ is chosen. The coefficients of the model are shrunken the more the bigger λ is chosen. Also the sparsity of the model increases [Hastie et al., 2009]. One possibility to choose λ is that this λ is chosen which minimises the prediction error as evaluated using the test set – data not used for fitting the model. In order to maximise the data usage for fitting the model, a cross validation scheme can be applied to estimate the prediction error and find the best λ [Friedman et al., 2010]. For k-fold cross validation, the data is divided into k equally big folds and the model is fitted with $k - 1$ folds. The other fold is used as the test set for evaluation. This process is repeated so that every fold is the test set once and the test error is averaged over all folds [Hastie et al., 2009].

Besides helping to deal with high dimensional data sets, the Lasso can provide a sparse model. This facilitates the interpretation of the model. However, a sparser model usually leads to an increased bias of the prediction, which results in a reduced accuracy [Hastie et al., 2009].

2.2.2 Priority-Lasso as extension of Lasso

Klau et al. [2018] introduced the priority-Lasso as an extension of the classical Lasso. Having a clinical setting in mind, it was especially developed to deal with data situations as described in section 2.1. The following description of priority-Lasso in this section is based on Klau et al. [2018]. They describe their method as follows: "Priority-Lasso is a hierarchical regression method which builds prediction rules for patient outcomes (e.g., a time-to-event, a response status or a continuous outcome) from different blocks of variables including high-throughput molecular data while taking clinicians' preference into account." Their approach is to sequentially fit a Lasso model to different *blocks* of data in such a manner that later blocks are only used to improve the prediction from the previous blocks. In this context, a block can consist of high dimensional omics data, but also established clinical parameters.

In order to assure that later blocks only improve the prediction of the previous blocks, the fitted linear predictor of block m is used as the offset when fitting block $m + 1$. An offset is always included in the linear predictor with a coefficient of 1. As the offset for block $m + 1$ is the prediction from the model for block m, "block $[m + 1]$ is fitted to the

outcome conditional on all blocks with higher priority" [Klau et al., 2018]. Therefore, the model is only trained on variability not explained so far. The concept of priority-Lasso can be incorporated into the equation 2 for Lasso regression by taking into account the block number m. The coefficients for block 1 are estimated by

$$\hat{\beta_0}^{(1)}, \hat{\boldsymbol{\beta}}^{(1)} = \underset{\beta_0^{(1)}, \boldsymbol{\beta}^{(1)}}{\arg\min} \left\| \mathbf{y} - \beta_0^{(1)} \mathbb{1} - \mathbf{X}^{(1)} \boldsymbol{\beta}^{(1)} \right\|_2^2 + \lambda^{(1)} \sum_{j=1}^{p_1} |\beta_j^{(1)}|. \tag{3}$$

The resulting linear predictor fitted in step 1 is then

$$\hat{\boldsymbol{\eta}}_1 = \hat{\beta_0}^{(1)} \mathbb{1} + \mathbf{X}^{(1)} \hat{\boldsymbol{\beta}}^{(1)}. \tag{4}$$

Now, this linear predictor is used as the offset for the next block, and the coefficients for block 2 are given as

$$\hat{\beta_0}^{(2)}, \hat{\boldsymbol{\beta}}^{(2)} = \underset{\beta_0^{(2)}, \boldsymbol{\beta}^{(2)}}{\arg\min} \left\| \mathbf{y} - \hat{\boldsymbol{\eta}}_1 - \beta_0^{(2)} \mathbb{1} - \mathbf{X}^{(2)} \boldsymbol{\beta}^{(2)} \right\|_2^2 + \lambda^{(2)} \sum_{j=1}^{p_2} |\beta_j^{(2)}|. \tag{5}$$

Please note that also in block 2 (and all further blocks) an intercept is fitted. Ideally, this intercept is estimated as 0 since an offset is already included. However, in reality the estimate can differ from 0. In the implementation of priority-Lasso in the programming language `R` [Klau et al., 2018], these intercepts are used to calculate the offsets for the next blocks during the model training. However, for the prediction of new data only the intercept of the first block is used. Therefore, I included the intercept in equation 5 contrary to the notation in Klau et al. [2018].

One obtains the final linear predictor,

$$\hat{\boldsymbol{\eta}}_M = \sum_{m=1}^{M} \hat{\beta_0}^{(m)} \mathbb{1} + \mathbf{X}^{(m)} \hat{\boldsymbol{\beta}}^{(m)}, \tag{6}$$

by summing up the results for all M blocks which are calculated analogously to equation 5.

2.3 Missing data

Missing data is a common problem in statistics. In the real world, there are many reasons why a value is missing in the observations. In section 2.3.1 I briefly describe single missing values. Then, I explain the problem of block-wise missing values considered in this work in detail in section 2.3.2.

2.3.1 Single missing values

In data sets, single values are often missing. In the following, the case of missing covariates is considered. This can be due to people not answering certain questions in a survey, participants in a clinical trial missing an appointment or a measurement device being faulty. An obvious way to handle missing values is to only analyse the observations with no missing values. This strategy is called *complete case analysis*. However, this approach may lack power due to the reduced data set. Additionally, it only leads to unbiased results if the data is *missing completely at random (MCAR)* [Janssen et al., 2010]. MCAR means that the probability of the events that lead to the missing values is independent of both the observed and the unobserved (missing) values [Heitjan and Basu, 1996]. A less strict assumption is *missing at random (MAR)*. For MAR, the probability of the observed missingness pattern depends on the observed values, but is independent of the unobserved values [Heitjan and Basu, 1996]. A third category is *missing not at random (MNAR)*, where the probability of the missingness pattern depends on the unobserved values itself [Sterne et al., 2009].

A common way to deal with MAR (and MCAR) data is *multiple imputation*. The idea is to use the observed values to create a predictive distribution of the missing values (so that the observed values are the covariates). By drawing the missing values from this distribution (called imputing), a complete data set is obtained. To account for the uncertainty that the imputed values are not the true observed ones, one not only generates one imputed data set, but multiple data sets [Sterne et al., 2009]. On these complete data sets the standard statistical analysis can be performed. Afterwards, the results are combined by Rubin's rules to obtain an overall estimate that takes the variation introduced by the imputation into account [Rubin, 2004].

2.3.2 Block-wise missing values

A more recently emerged problem is block-wise missing values. In this case, a block is defined as all these covariates that belong together, e.g. because they were all measured by the same methodology. In the life sciences and in clinical settings, the different omics data sets represent such blocks. Typically, one methodology is used to measure hundreds or thousands of targets. For example, RNA-Seq can be used to determine the expression level of all (roughly) 20000 protein-coding genes in the human body [Fagerberg et al., 2014].

As described in section 2.1, the state of a biological system can be measured by different methods. In a clinical setting, this means that the diagnostics of a disease can lead to several different data sets or blocks. However, not all hospitals and laboratories have the technical or financial means to generate all possible data sets. Especially in studies that

are carried out in different hospitals, this can lead to an overall data set with block-wise missing values. A similar situation can arise if over time more or less funding for more expensive methods is available or new technologies are developed.

2.3.3 Existing methods for dealing with block-wise missing values

The approach for single missing values, as mentioned in section 2.3.1, to impute all missing values, can be challenging applied to a complete missing block. For the complete block, there is no information available that could be used in an imputation model. As these imputation methods are not so effective when a high percentage of the data is missing, they do not work well with the block-wise missingness pattern [Yuan et al., 2012]. Also, the assumption that the data is missing at random is not fulfilled, as a complete block is missing, which is not taken into account in the classical imputation methods [Xiang et al., 2014]. Another point is that the dimensionality of the data can be so high that it leads to computational problems for these methods. Therefore, in the recent years several methods were developed to deal with block-wise missing data. Two main reoccurring aspects are to divide the data into subsets based on the missingness pattern of the blocks and to make use of the block-wise missingness structure for the imputation. A lot of these methods originate from the domain of (brain) imaging data.

Ingalhalikar et al. [2012] were one of the first to propose a method, which is based on ensemble learning. For this method, the data is divided into subsets so that every subset has complete observations for a specific combination of blocks. One observation can be contained in several subsets. At least as many subsets are used so that all observations from the original data set are included. Then, on each of these subsets a model is trained. The model can be chosen freely from the available model classes. To make a prediction for an observation, the individual predictions of this ensemble of models are combined. For this, the models trained on the different subsets are weighted by their expected error [Ingalhalikar et al., 2012].

The method proposed by Krautenbacher [2018] also uses weighting to combine several models, in his case classifiers. In contrast to Ingalhalikar et al. [2012], a classifier is trained on every block and uses all observations that have values for a given block. Then, for the overall prediction the different classifiers are weighted by a measure for their prediction quality.

In 2012, Yuan et al. [2012] proposed the incomplete Multi-Source Feature (iMSF) learning method. In a similar manner as Ingalhalikar et al. [2012], the data is divided into subsets depending on the missingness structure. However, the subsets are disjoint so that an observation can only be contained in one subset. Then, on every subset a regularised regression model is fitted. However, the models are fitted jointly and have the additional

constraint that in all models (for a subset) that share a block, the same features within this block have to be selected. However, the coefficients for one block do not need to have the same values. This leads to a regularisation structure similar to the group lasso [Yuan and Lin, 2006]. For a new prediction (with a certain combination of blocks), the model that was trained on the same combination is used.

The iMFS method was developed further by Xiang et al. [2014] to the incomplete Source-Feature Selection (iSFS) model. In contrast to iMSF, one observation can be contained in several subsets, creating a subset for every missingness combination of the blocks. Then, a penalised regression model for every block is learned jointly over all subsets. This leads to the identical model for a given block, no matter in which subset an observation with this block is included. Additionally, in every subset for every block a parameter is learned that determines how much influence the prediction from a given block has in the overall prediction for this subset. Hence, this parameter takes the different combination of blocks in the subsets into account. To do a prediction on test data, the same model is used for a block irrespective of the observation's missingness pattern, but its influence weighted differently depending on the missingness pattern.

Liu et al. [2017] proposed a method that also divides the observations into subsets depending on the missingness pattern. An observation can be contained in several subsets. Then, on every subset a hypergraph is learned. The structural information of the different hypergraphs are combined and used to train a classifier. Because the hypergraphs are learned jointly on both the train and the test data, the classifier can be used to label the test data.

Thung et al. [2014] developed a method based on matrix completion. In contrast to the aforementioned methods, Thung et al. [2014] actually impute values. In order to reduce the number of variables and samples that have to be imputed, they first shrink the data set. This is done in two steps. First, like in the iSFS method, different subsets are generated. Then, on every subset a penalised regression model is fitted and verified by cross validation. Every feature that is not at least selected once in the different models is discarded. In the second step, the sample size of the data for which the outcome is known (training data) is reduced. For this, the covariates of the training data are used to predict the covariates of the test data (for which the outcome is not known). In order to do this, a penalised regression model in a multi-task learning setting is used. For every missingness pattern subset that is shared by training and test set observations, a regression is performed. Observations that are not selected in any subset are excluded from the training data set. Afterwards, the shrunken training data set and the test data are stacked together in one matrix, with both the covariates and the outcome. The missing values in the matrix are then imputed by either a trace norm minimisation algorithm or

a regularised expectation maximisation algorithm.

Cai et al. [2016] proposed a new structured matrix completion algorithm which makes use of the block-wise missingness structure to impute the missing values. It assumes that the matrix is approximately low rank. This paper is the basis of the work of Linder and Zhang [2019]. In contrast to Cai et al. [2016], which can only handle block-wise missing values, the method from Linder and Zhang [2019] can also deal with single missing values. It reorders the samples and features of the data set to always obtain a block-wise missing structure. On that data set the structured matrix completion algorithm from Cai et al. [2016] is applied.

Another approach to impute block-wise missing data is made by Zhu et al. [2020]. They assume that every block has an individual structure but also a shared joint structure that is exploited for imputation. Zhu et al. [2020] consider the data as realisations from exponential families. The underlying parameters of these exponential families are then estimated and the parameters for missing values imputed. In order to do this, the parameters are decomposed into their individual and joint structure in a principal component wise fashion. To impute the missing values, the joint likelihood is maximised.

Xue and Qu [2020] introduced the Multiple Block-wise Imputation (MBI) approach. Similar to Yuan et al. [2012], they divide the observations according to their missingness pattern into disjoint subsets. Then, every missing value in a given subset is imputed multiple times. For a missing value of a given subset, an imputation model each is built with all other subsets that contain the missing block and at least one other block that is also observed in the given subset. The imputation is done with a penalised generalised linear model. Afterwards, the multiple imputed data sets are used to generate estimation equations in a penalised fashion and the different estimators are combined to yield one prediction for the outcome per observation.

Lorenzo et al. [2019b] proposed a method that is based on partial least squares (PLS) called mdd-sPLS. In PLS, a regression is made between projections of the dependent and independent variables. The projections are constructed in such way that the covariance between them is maximised. Lorenzo et al. [2019b] perform a PLS independently for every block and the outcome. The predictors from the individual PLS are combined to yield one overall prediction. Missing values are mean imputed and then the PLS procedure is applied in an expectation maximisation fashion to generate better imputations. If new observations contain missing values, the PLS approach is used to impute the missing blocks from the observed blocks. Then, the complete (imputed) data set can be used for prediction.

Another imputation approach was introduced by Hieke et al. [2016]. In their case, they have data with three different blocks and one block is observed across all observa-

tions. Most observations are missing one of the other two blocks, and only a subset of observations have values for all blocks. Hieke et al. [2016] first built a penalised regression model with block one and two and used the prediction of this model as the offset for a next model. In the second model, the offset and block one and three are used. For observations that have missing values for block two, the offset (which depends on this missing information) is imputed. The imputation model for the offset uses the observations with values for both block two and three. So in contrast to other imputation methods, Hieke et al. [2016] do not impute the complete covariates, but only an aggegrated measure of it.

2.3.4 mdd-sPLS method

Because of its flexibility, good results and implementation for the programming language `R`, the method mdd-sPLS developed by Lorenzo et al. [2019b] was used for comparison with priority-Lasso. In this work, version 1.1.7 of the package `ddsPLS` was downloaded on 24.03.2020 from `https://github.com/hlorenzo/ddsPLS` and used. In the following, I give a more detailed description of mdd-sPLS. The method description is based on Lorenzo et al. [2019b].

At its core, the method uses PLS which is based on singular value decomposition (SVD) of the covariance matrix between the outcome $\mathbf{Y}$ and the covariates $\mathbf{X}$. This leads to the generation of latent variables for the outcome and the covariates. They are constructed in such way that they maximise the covariance between $\mathbf{X}$ and $\mathbf{Y}$. The latent variables can then be used to approximate $\mathbf{X}$ and $\mathbf{Y}$ and regress $\hat{\mathbf{Y}}$ onto $\hat{\mathbf{X}}$ [Abdi and Williams, 2013]. In contrast to the aforementioned method, Lorenzo et al. [2019b] uses a soft-thresholded covariance matrix for the SVD. Soft-thresholded means that only entries whose absolute value are bigger than a threshold are kept, otherwise they are set to 0. He calls this method covariance-thresholding sparse PLS (ct-sPLS). ct-sPLS is applied separately to all blocks. In order to combine the information from the different blocks, the weights from the SVD used to generate the latent variables are pooled. The pooling is performed in such way that the subsequent regression is done in a common subspace over all blocks. This method is called Multi-Data-Driven-sPLS (mdd-sPLS).

When values are missing in the training data, they are mean imputed. Afterwards, the mdd-sPLS algorithm is performed. Only originally missing variables that are selected in this model get a better imputation. In order to do this, another mdd-sPLS model is built. This model predicts the variables used in the first model from information about the outcome and is used to impute the missing values. The process of building a model to predict the outcome and then another model to impute the missing variables is repeated until convergence of the latent variables of the covariates.

If values are missing in the test data, a mdd-sPLS model is used to impute the missing

values from the observed data. As in the imputation model for the training data, only variables that are used in the prediction model are imputed. The complete data set can then be used to make a prediction for the test data.

3 Extensions of priority-Lasso to handle missing values

One goal of this work is to adapt priority-Lasso in such way that it can deal with block-wise missing values. Due to the sequential fitting of separate models for every block, one simple approach is to just ignore the observations with missing data for a given block when fitting the model. This means that for the ignored observations no offset is calculated for the next block. Instead, the offset (including the intercept) from the previous block is used for these observations. I call this approach priority-Lasso-ignore, abbreviated as pL-ign, which is detailed in section 3.1.

The second approach applies the idea of Hieke et al. [2016] to priority-Lasso. Instead of imputing all missing covariates, only the offset that is calculated from the covariates is imputed for the observations with missing values. I term this approach priority-Lasso-impute, abbreviated as pL-imp. Section 3.2 describes two different ways of doing this.

The extensions of priority-Lasso are based on the package `prioritylasso` version 0.2.3 from The Comprehensive R Archive Network [Klau et al., 2018]. Before working on the extension, I refactored the code, removed an error when using binary outcomes and enabled the possibility to include the intercepts from the models of all blocks in the prediction of new data. I additionally implemented the option that for the prediction only a subset of the blocks are used.

3.1 priority-Lasso-ignore

The basic idea of this approach is that the Lasso model for every block is only fitted with the observations that have no missing values for this block. In this way, no data has to be imputed and all the available data is used. However, for observations with the current block missing, no offset for the next block can be calculated. To circumvent this problem, for these observations the offset (including the intercept) from the previous block is carried forward. If observations lack the values for the first block, the offset for the second block is either set to 0 or to the estimated intercept of the first block (pL-ign (zero/intercept)). In the following, a formal notation is provided.

Let the offset of a block m for an observation i be denoted as $\delta_{m,i}$, defined as

$$
\begin{aligned}
&\delta_{1,i} = 0 \\
&\delta_{2,i} = \begin{cases} \hat{\eta}_{1,i}, & \text{if } x_{i1}^{(1)}, ..., x_{ip_1}^{(1)} \text{ are not missing} \\ 0 \text{ or } \hat{\beta}_0^{(1)}, & \text{if } x_{i1}^{(1)}, ..., x_{ip_1}^{(1)} \text{ are missing} \end{cases} \\
&\delta_{m,i} = \begin{cases} \hat{\eta}_{m-1,i}, & \text{if } x_{i1}^{(m-1)}, ..., x_{ip_{m-1}}^{(m-1)} \text{ are not missing} \\ \delta_{m-1,i}, & \text{if } x_{i1}^{(m-1)}, ..., x_{ip_{m-1}}^{(m-1)} \text{ are missing} \end{cases} \qquad m = 3, ..., M \\
&\forall i,
\end{aligned}
\tag{7}
$$

where $\hat{\eta}_{m,i}$ is the prediction from block m for the ith observation.

3.2 priority-Lasso-impute

As in the previous approach, in the second approach the Lasso models fitted to the separate blocks only use the available data and no covariates are imputed. However, the offset for the observations with missing values is not carried forward from the previous block, but imputed. This has the advantage that instead of imputing a possibly very high number of covariates, only one value is imputed. The idea for this approach comes from Hieke et al. [2016] where also only an offset is imputed. In this paper, they only use three blocks and have therefore a fixed setting which blocks are used for the imputation model. I generalised this approach and implemented two different strategies which blocks are used. Either all other blocks are used (pL-imp (complete)), which is detailed in section 3.2.1, or it is tried to use as much information as possible for more complex missingness patterns (pL-imp (available)). This approach is described in section 3.2.2. In general, any model that can predict a continuous outcome can be used as the imputation model. In the following sections, this is represented by only referencing a general regression model $F(\cdot)$. As priority-Lasso is a Lasso based method, for the imputation model I also resorted to a Lasso model in the implementation.

3.2.1 Complete cases

The first approach uses all other blocks than the current one to impute the offset derived from the current block. This requires that the data set contains some observations that have no missing values (complete cases). One has to decide on a sensible lower threshold for the number of complete cases. Based on Hieke et al. [2016] that used 26 complete cases, the default is set to 30 cases. Another restriction of this approach is that in every observation with missing data, at most one block can be missing (because all other blocks

are used in the imputation model). In the following, a more formal notation is provided.

If only the complete cases are used for the imputation model and $I = \{1, ..., n\}$ is the set of observation indices, the observations used for the imputation model are defined as

$$I_{imp}^{comp} = \{i \in I \mid x_{i1}^{(m)}, ..., x_{ip_m}^{(m)} \text{ are not missing } \forall m\}, \tag{8}$$

where $x_{i1}^{(m)}, ..., x_{ip_m}^{(m)}$ are the covariates of the mth block of the ith observation. Let $F(\cdot)$ be a general regression model that can be trained by providing the dependent variable and the independent variables. Then $\widetilde{F}(\cdot)$ is a trained regression model that returns its prediction when provided with the independent variables.

Algorithm 1 describes how for an observation j that misses block m the linear predictor is imputed as $\hat{\kappa}_{m,j}^{comp}$.

Algorithm 1: Impute missing offsets using complete cases

```
determine the set of all observations without missing values I_imp^comp
foreach block m in number of blocks M do
    train imputation model F_m^comp with η̂_{m,i} as the dependent variable and the
     covariates from all blocks except block m as the independent variables using
     all observations i ∈ I_imp^comp to generate F̃_m^comp
    foreach observation j that misses block m do
        use the imputation model F̃_m^comp to impute the result for block m as κ̂_{m,j}^comp
         with all other blocks than m as the independent variables
    end
end
```

Subsequently, the offset for a given observation i in block m is given as

$$\delta_{1,i} = 0$$

$$\delta_{m,i} = \begin{cases} \hat{\eta}_{m-1,i}, & \text{if } x_{i1}^{(m-1)}, ..., x_{ip_{m-1}}^{(m-1)} \text{ are not missing} \\ \hat{\kappa}_{m-1,i}^{comp}, & \text{if } x_{i1}^{(m-1)}, ..., x_{ip_{m-1}}^{(m-1)} \text{ are missing} \end{cases} \qquad m = 2, ..., M \tag{9}$$

$$\forall i \ s.t. \text{ every observation } i \text{ misses at most 1 block } m.$$

3.2.2 Available cases

The previous approach always uses all other blocks to impute the offset for a missing block. However, often data sets do not contain observations with complete cases or observations lack more than only one block. For these cases I developed a more flexible approach. It divides the observations into different subsets, an idea a lot of the existing methods described in section 2.3.3 resort to. All observations that miss block m are grouped by

their missingness pattern. A missingness pattern describes which combination of blocks are missing for a given observation. Then, for every unique pattern an imputation model is fitted. An overview about this approach is given in algorithm 2.

First, for all observations that miss block m all unique missingness patterns Q_m are determined. A missingness pattern q is defined as the information for every block if its values are observed (✓) or missing (✗). For a model that uses 3 blocks, observations that miss block 1 can have the following missingness patterns: {✗, ✓, ✓}, {✗, ✗, ✓}, {✗, ✓, ✗}. Not all patterns need to be observed in the training data.

In the next step, for every missingness pattern q all observations that can be used for the imputation model are found by `FindAllSuitableObservations` and the indices of these observations are stored in I_{imp}^{avail}. This step is explained in detail later on. Similar to the algorithm using the complete cases, these observations are used to generate the trained imputation model $\widetilde{F}_{m,q}^{avail}$ which in turn is used to impute the linear predictor of block m for every observation j that misses this block. This imputation is denoted as $\hat{\kappa}_{m,j}^{avail}$.

`FindAllSuitableObservations` searches for observations from the training data that can be used for an imputation model for observations that miss block m and have the specific missingness pattern q. Possible observations have to fulfil two criteria: (1) block m has to be observed and (2) at least one other block that is observed in the pattern q has to be observed. For example, for $m = 1$ and $q = \{✗, ✓, ✓, ✗\}$ observations with the pattern {✓, ✗, ✗, ✓} are not considered, as block 4 is not observed by observations with the pattern q and therefore cannot be used in an imputation model.

The pseudocode is given in algorithm 3. For a given missingness pattern q for block m, `FindAllSuitableObservations` generates all possible patterns P that satisfy the above mentioned criteria. In the example, these are {✓, ✓, ✓, ★}, {✓, ✓, ✗, ★} and {✓, ✗, ✓, ★}, where ★ denotes that the block can be either observed or missing. Blocks denoted with ★ are not of interest because they cannot be used for the imputation model. Then, for every pattern $p = 1, ..., P$ it is counted how many observations have observed values for all blocks marked with ✓ in the pattern p (and therefore can be used for an imputation model using these blocks).

Afterwards, one pattern p is chosen for the imputation model. For this, two strategies exist. Strategy *max. observations* or *max. n* chooses the pattern that has the most observations that can be used for the imputation model (pL-imp (available, max. n)). Strategy *maximise blocks* or *max. blocks* chooses the pattern that uses the most high priority blocks (pL-imp (available, max. blocks)). The pattern is determined as follows: If the pattern with the most observed (✓) blocks (the start pattern p_{start}) has more or equal observations than a threshold (the default is 30), this pattern is used. If this is not

the case, the observed block with the lowest priority is set to missing (✗) and this new pattern is considered and so on. If the new pattern only consists of missing blocks except block m, then the start pattern is again considered but now with the observed block with the highest priority set to missing and the procedure is repeated.

Let us consider the missingness pattern $q = \{✗, ✓, ✓, ✓\}$ for block $m = 1$ as an example. Then, the patterns P for the imputation model are checked in the following order for strategy *max.blocks*: $\{✓, ✓, ✓, ✓\}$, $\{✓, ✓, ✓, ✗\}$, $\{✓, ✓, ✗, ✗\}$, $\{✓, ✗, ✓, ✓\}$, $\{✓, ✗, ✓, ✗\}$ and $\{✓, ✗, ✗, ✓\}$.

Algorithm 2: Impute missing offsets using available cases

foreach *block m in number of blocks M* **do**
  determine all Q_m unique missingness patterns
  for *missingness pattern* $q \leftarrow 1$ **to** Q_m **do**
    $I_{imp}^{avail} \leftarrow$ `FindAllSuitableObservations`
    train imputation model $F_{m,q}^{avail}$ using all observations $i \in I_{imp}^{avail}$ with $\hat{\eta}_{m,i}$ as the dependent variable and the covariates from the blocks determined by `FindAllSuitableObservations` as the independent variables to generate $\widetilde{F}_{m,q}^{avail}$
    foreach *observation j that misses block m and has the pattern q* **do**
      use the imputation model $\widetilde{F}_{m,q}^{avail}$ to impute the result for block m as $\hat{\kappa}_{m,j}^{avail}$
    end
  end
end

Algorithm 3: `FindAllSuitableObservations`

input : missingness pattern q for block m

determine all patterns $p = 1, ..., P$ that satisfy the criteria that (1) block m is observed and (2) at least one block that is observed in q is also observed in p

foreach *pattern p in number of pattern P* **do**

  $c_p \leftarrow$ number of observations that have observed values for the blocks marked with ✓ in p

end

choose a pattern p which observed blocks are used for the imputation model

if *strategy == max. n* **then**

  choose that pattern p_{use} from patterns $1, ..., P$ that has the most observations ($\text{argmax}_p(c_p)$)

end

if *strategy == max. blocks* **then**

  choose that pattern p_{use} that uses the most blocks (with high priority) for the imputation model

  determine a threshold t of the minimum number of observations used for the imputation model

  $p_{temp} \leftarrow p_{start} \leftarrow$ that pattern p with the most observed (✓) blocks

  while *no pattern p_{use} chosen yet* **do**

    if *number of observations for pattern $p_{temp} \geq t$* **then**

      $p_{use} \leftarrow p_{temp}$

    else

      $p_{temp} \leftarrow$ pattern p_{temp} but with the least important observed (✓) block set to missing (✗)

      if p_{temp} now only consists of missing blocks (except block m) or blocks that are of no interest for the imputation model (★), set $p_{temp} \leftarrow p_{start}$

      set the block with the hightest priority of p_{temp} to missing (✗)

      $p_{start} \leftarrow p_{temp}$

    end

  end

end

return: index of all observations that have observed values for the blocks marked with ✓ in pattern p_{use} and which blocks to use for the imputation model

Subsequently, the offset for a given observation i in block m is given as

$$\begin{aligned}
&\delta_{1,i} = 0 \\
&\delta_{m,i} = \begin{cases} \hat{\eta}_{m-1,i}, & \text{if } x_{i1}^{(m-1)}, ..., x_{ip_{m-1}}^{(m-1)} \text{ are not missing} \\ \hat{\kappa}_{m-1,i}^{avail}, & \text{if } x_{i1}^{(m-1)}, ..., x_{ip_{m-1}}^{(m-1)} \text{ are missing} \end{cases} \qquad m = 2, ..., M \qquad (10) \\
&\forall i.
\end{aligned}$$

3.3 Prediction of new data

Depending on the method how to deal with block-wise missing values that was used with the training data, it differs how priority-Lasso can deal with missing values in new (test) data. If the training data contains block-wise missing values and was fitted with priority-Lasso-ignore, this enables to build a model for the training data, but it does not provide a method to deal with missing data in test data per se. Therefore I enabled the ad hoc solution to set block-wise missing values in the test data to 0. In general, this means that besides the estimated intercept of this block, the information from this block is not included. In the special case that the data is centered before the application of priority-Lasso, it corresponds to mean imputation of the missing values.

If the training data was fitted with priority-Lasso-impute (complete), new observations can have missing values in at most one block. Irrespective of the actual missingness situation in the training data, the imputation models are always calculated. When there are no observations with missing values in the training data, the imputation models are not needed for the model estimation, but can be used if new test data has block-wise missing values.

When using priority-Lasso-impute (available), in contrast, the missingness patterns of the new observations have to be already included in the training data, otherwise no prediction is possible.

Apart from dealing with missing values, I implemented two further options for the prediction using priority-Lasso. Firstly, it is now possible to include the intercepts of all blocks in the prediction. As mentioned in section 2.2.2, ideally all intercepts except the first are estimated as 0. However, in practise this is not the case and the intercepts are used to calculate the offsets of the priority-Lasso model, which influence the overall prediction. Therefore, they should also be used for the prediction of new data and it is recommended to use this option.

Secondly, I implemented the feature that one can chose which blocks are used for the overall prediction. This enables to determine how important the contribution of individual blocks are to the overall prediction.

4 Simulation experiment

In order to assess the performance of the different approaches for priority-Lasso to deal with block-wise missing data, a simulation experiment was conducted. ddsPLS was used used as the comparison with an existing method. Simulating the data allows to precisely control for the underlying structure of the data (e.g. the correlation), perform many repetitions of the experiment and a large test set to determine the generalisation error of the methods. I first describe how the data is simulated, then which different settings were tested and present the results.

4.1 Data generation

The aim of the simulation is to generate data that closely resembles data sets from real world scenarios. Priority-Lasso was designed to especially deal with clinical data sets that comprise of blocks from different diagnostic methods. Therefore, I adapted the approach from De Bin et al. [2019] that simulates many small correlated blocks of clinical and gene expression data and afterwards splits it into a clinical and a molecular block. I adjusted it so that every block is simulated on its own but one can control the intra- and inter-block correlation. First, a complete data set is simulated and then different missingness patterns introduced.

4.1.1 Generation of complete data sets

In total, three blocks are simulated. One block of clinical variables with 20 covariates $\mathbf{C}$, a block of molecular variables with 500 covariates $\mathbf{M}_1$ and another block of molecular variables with 2000 covariates $\mathbf{M}_2$. This reflects different technologies that can lead to varying dimensionality of their results. The size of the clinical block is based on a real data set in Klau et al. [2018]. Because of computational restrictions, only a part of the covariates of each molecular block are simulated with intra- and inter-block correlation. The data that is modelled in such way can be seen as information generated by a gene pathway, i.e. a biological process that has connected several molecular outcomes. This data is generated from a multivariate Gaussian distribution,

$$(\mathbf{C}, \mathbf{M}_1^c, \mathbf{M}_2^c) \sim \mathcal{N}_{k_{C+M_1^c+M_2^c}}((\boldsymbol{\mu}_C, \boldsymbol{\mu}_{M_1}, \boldsymbol{\mu}_{M_2})^\intercal, \boldsymbol{\Sigma}), \tag{11}$$

where $k_{C+M_1^c+M_2^c}$ denotes the number of variables in the combined blocks and $\mathbf{M}_1^c$ and $\mathbf{M}_2^c$ are the parts of the molecular blocks that are modelled with correlation. Because of its small size, the clinical block $\mathbf{C}$ is completely modelled with correlation. $\boldsymbol{\mu}_C$, $\boldsymbol{\mu}_{M_1}$ and $\boldsymbol{\mu}_{M_2}$ are the means of the respective blocks. Here, for all clinical variables it is set to 1 (so

$\boldsymbol{\mu_C}$ is a vector of ones) and for the molecular variables it is set to 6. $\boldsymbol{\Sigma}$ is the symmetric covariance matrix, which is constructed in a block-wise fashion,

$$\boldsymbol{\Sigma} = \left(\begin{array}{ccc|ccc|ccc} \sigma_C^2 & \cdots & \rho_C\sigma_C & \rho_{CM_1}\sigma_C\sigma_{M_1} & \cdots & \rho_{CM_1}\sigma_C\sigma_{M_1} & \rho_{CM_2}\sigma_C\sigma_{M_2} & \cdots & \rho_{CM_2}\sigma_C\sigma_{M_2} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \rho_C\sigma_C & \cdots & \sigma_C^2 & \rho_{CM_1}\sigma_C\sigma_{M_1} & \cdots & \rho_{CM_1}\sigma_C\sigma_{M_1} & \rho_{CM_2}\sigma_C\sigma_{M_2} & \cdots & \rho_{CM_2}\sigma_C\sigma_{M_2} \\ \hline & & & \sigma_{M_1}^2 & \cdots & \rho_{M_1}\sigma_{M_1}^2 & \rho_{M_1M_2}\sigma_{M_1}\sigma_{M_2} & \cdots & \rho_{M_1M_2}\sigma_{M_1}\sigma_{M_2} \\ & & & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ & & & \rho_{M_1}\sigma_{M_1}^2 & \cdots & \sigma_{M_1}^2 & \rho_{M_1M_2}\sigma_{M_1}\sigma_{M_2} & \cdots & \rho_{M_1M_2}\sigma_{M_1}\sigma_{M_2} \\ \hline & & & & & & \sigma_{M_2}^2 & \cdots & \rho_{M_2}\sigma_{M_2}^2 \\ & & & & & & \vdots & \ddots & \vdots \\ & & & & & & \rho_{M_2}\sigma_{M_2}^2 & \cdots & \sigma_{M_2}^2 \end{array}\right), \tag{12}$$

where σ_C^2 is the variance of the clinical data, $\sigma_{M_1}^2$ the variance of the first molecular block and $\sigma_{M_2}^2$ the variance of the second molecular block. ρ_C is the intra-block correlation of the clinical data, ρ_{M_1} and ρ_{M_2} the intra-block correlations of the first and second molecular block, respectively. ρ_{CM_1} is the inter-block correlation between the clinical and the first molecular block, ρ_{CM_2} the inter-block correlation between the clinical and the second molecular block and $\rho_{M_1M_2}$ the inter-block correlation between the two molecular blocks. Following De Bin et al. [2019], the standard deviations are set to $\sigma_C = 0.5$ and $\sigma_{M_1} = \sigma_{M_2} = 0.65$. The correlations are varied throughout the simulation study. In case the generated covariance matrix is nonpositive definite, Higham's algorithm is applied to calculate the closest positive definite matrix and the resulting matrix used instead.

In every molecular block, the amount of correlated variables are set to 70, roughly representing the size of a gene network as presented in Carro et al. [2010]. The other variables of the molecular blocks, $\mathbf{M}_1^u$ and $\mathbf{M}_2^u$, are independently drawn from a Gaussian distribution as

$$(\mathbf{M}_1^u, \mathbf{M}_2^u) \sim \mathcal{N}_{k_{M_1^u+M_2^u}}((\boldsymbol{\mu}_{M_1}, \boldsymbol{\mu}_{M_2})^\intercal, diag(\sigma_{M_1}^2 \mathbb{1}_{k_{M_1^u}}, \sigma_{M_2}^2 \mathbb{1}_{k_{M_2^u}})). \tag{13}$$

Here, $k_{M_1^u+M_2^u}$ denotes the number of variables in the combined molecular blocks that are modelled independently, and $\mathbb{1}_{k_{M_i^u}}$ is a vector of ones with length $k_{M_i^u}$. The same means and standard deviations as in equations 11 and 12 are used. The complete molecular blocks are given as the concatenation of the two parts as

$$\mathbf{M}_i = (\mathbf{M}_i^c, \mathbf{M}_i^u) \quad i = 1, 2, \tag{14}$$

and the number of variables per block is denoted as

$$k_{M_i} = k_{M_i^c} + k_{M_i^u} \quad i = 1, 2. \tag{15}$$

To further differentiate the two different data types, noise is added to the molecular blocks. The noise is separated into two parts,

$$\mathbf{G}_i = exp\{\mathbf{M}_i + \mathbf{B}_i\} + \mathbf{A}_i, \quad i = 1, 2, \tag{16}$$

where $\mathbf{B}_i$ is multiplicative noise and $\mathbf{A}_i$ additive noise. The multiplicative noise can result from "variation between the pixel affecting gene expression measurements" [De Bin et al., 2019] and is modelled as

$$\mathbf{B}_i \sim \mathcal{N}_{k_{M_i}}((0, \cdots, 0)^\intercal, diag(\boldsymbol{\phi})) \quad i = 1, 2. \tag{17}$$

Here, $\boldsymbol{\phi}$ is a vector of length k_{M_i} with all elements of value 0.1. The additive noise represents technical noise and is modelled as

$$\mathbf{A}_i \sim \mathcal{N}_{k_{M_i}}(\boldsymbol{\nu}, diag(\boldsymbol{\tau})) \quad i = 1, 2, \tag{18}$$

where $\boldsymbol{\nu}$ and $\boldsymbol{\tau}$ are vectors with length k_{M_i} and all elements have the value 10 and 20, respectively. Afterwards, a normalisation step was performed by assigning values smaller than 10 the threshold of 10 and values bigger than 16000 the threshold of 16000 and transforming the molecular blocks with the natural logarithm.

This data can be used to generate a continuous outcome with a linear model,

$$y = \mathbf{C}\boldsymbol{\beta}_C + \mathbf{M}_1\boldsymbol{\beta}_{M_1} + \mathbf{M}_2\boldsymbol{\beta}_{M_2} + \boldsymbol{e}, \tag{19}$$

where $\boldsymbol{\beta}_C$, $\boldsymbol{\beta}_{M_1}$ and $\boldsymbol{\beta}_{M_2}$ are the true parameters for the respective blocks. $\boldsymbol{e}$ is additive Gaussian noise modelled with

$$\boldsymbol{e} \sim \mathcal{N}_n((0, \cdots, 0)^\intercal, \sigma_{outcome}\mathbb{I}_n), \tag{20}$$

where n is the number of observations, $\mathbb{I}_n$ a $n \times n$ identity matrix and $\sigma_{outcome} = 6$ the standard deviation of the noise.

Based on the real data set results in Klau et al. [2018], in the clinical block all 20 variables have an influence. In this simulation study, the same effect sizes as in De Bin et al. [2019] are used. A weak effect size is randomly chosen as -1 or 1, a medium effect size as -2 or 2. In the clinical block, 10 variables have a weak effect and 10 a medium

effect. In each molecular block, 30 variables are modelled with a weak effect, reflecting the situation that several genes show a cumulated effect on the prediction. The variables that have an influence (where $\boldsymbol{\beta}_C$, $\boldsymbol{\beta}_{M_1}$ and $\boldsymbol{\beta}_{M_2}$ are not 0) are all taken from the variables that are modelled with a correlation structure.

4.1.2 Introduction of missingness patterns

After the complete data set is generated, a missingness pattern with block-wise missing values is introduced. In every block, all covariates of that block are deleted for a certain percentage of the observations. The observations are chosen randomly and independently for every block. In the simulation study, 0%, 5%, 10%, 25%, 50% and 75% missing observations are used. All 216 permutations of these missingness fractions (in the following called missingness pattern) for the three blocks are tested in every experiment. All different missingness patterns are introduced into the same complete data set.

4.2 Simulation setup

In the following, I describe how the simulation was set up and how the results were evaluated.

4.2.1 General setup

All models were trained on a simulated data set with 200 observations, reflecting often rather small sample sizes in biomedical studies. In total, for every experiment (with a certain parameter setting) 100 different training sets were simulated. Then, the models of each repetition were evaluated on the same test set consisting of 1000 observations. The evaluation was performed both on complete test data and test data that has the same missingness pattern as the training data. In case that a model did not yield a result in every repetition, the worst respective metric observed for this method in this experiment was used for the repetitions with no results. If a method failed for all repetitions (e.g. impute missing offsets with complete cases when there are no complete cases), this model was excluded from the analysis. When the missing test data included observations with a missingness pattern not observed in the training data, these observations were excluded from the predictions made by the model imputing missing offsets using the available cases.

4.2.2 Used methods

In every experiment, priority-Lasso was used in five different ways. On the training data with missing observations, priority-Lasso was trained with (1) the option to ignore the missing data, (2) the option to impute the missing offsets using complete observations and

(3) the option to impute the missing offsets using available observations. Additionally, (4) a priority-Lasso model was fitted on the remaining complete observations after discarding observations with missing values and (5) priority-Lasso was applied to the original complete data set. For all models, the default parameters from the existing priority-Lasso package [Klau et al., 2019] were used.

For comparison with an existing method, the mdd-sPLS method from the ddsPLS package [Lorenzo et al., 2019a] was used. After consultation with the method developer, the following options were used: A 10 fold cross validation was performed to find the best model with respect to the penalisation parameter λ. The 10 different λ values to test were automatically computed (`n_lambda = 10`) and the number of compenents fixed to one (`R = 1`).

As a baseline, the mean response of the training data was calculated and used as the prediction for all observations in the test data.

4.2.3 Metrics

To assess the accuracy of the different methods and compare them, the mean squared error (MSE) of the predictions on the test data was reported. The MSE is defined as the mean of the squared differences between the predictions and the true values. Also, the ability to select the correct variables by priority-Lasso was compared to evaluate the robustness of the models.

4.2.4 Prediction settings

For the prediction on new test data, several situations were included. Predictions were made on (1) complete test data and (2) on complete test data only using the first block or the first block plus the second or third block to assess the importance of additional blocks for the prediction. To reflect situations where the new data also has missing observations, the priority-Lasso models that can deal with block-wise missing data and the mdd-sPLS model were used to make predictions on (3) test data with the same missingness pattern as the training data and (4) test data where only a subset of the blocks have missing observations (with the same percentage as the training data).

4.3 Results

4.3.1 Comparison of prediction with and without all intercepts

As explained in sections 2.2.2 and 3, I implemented the option to use the intercepts of all blocks for the predictions. This was compared to the option to only use the intercept of the first block for ignoring missing data and imputing missing offsets using the available

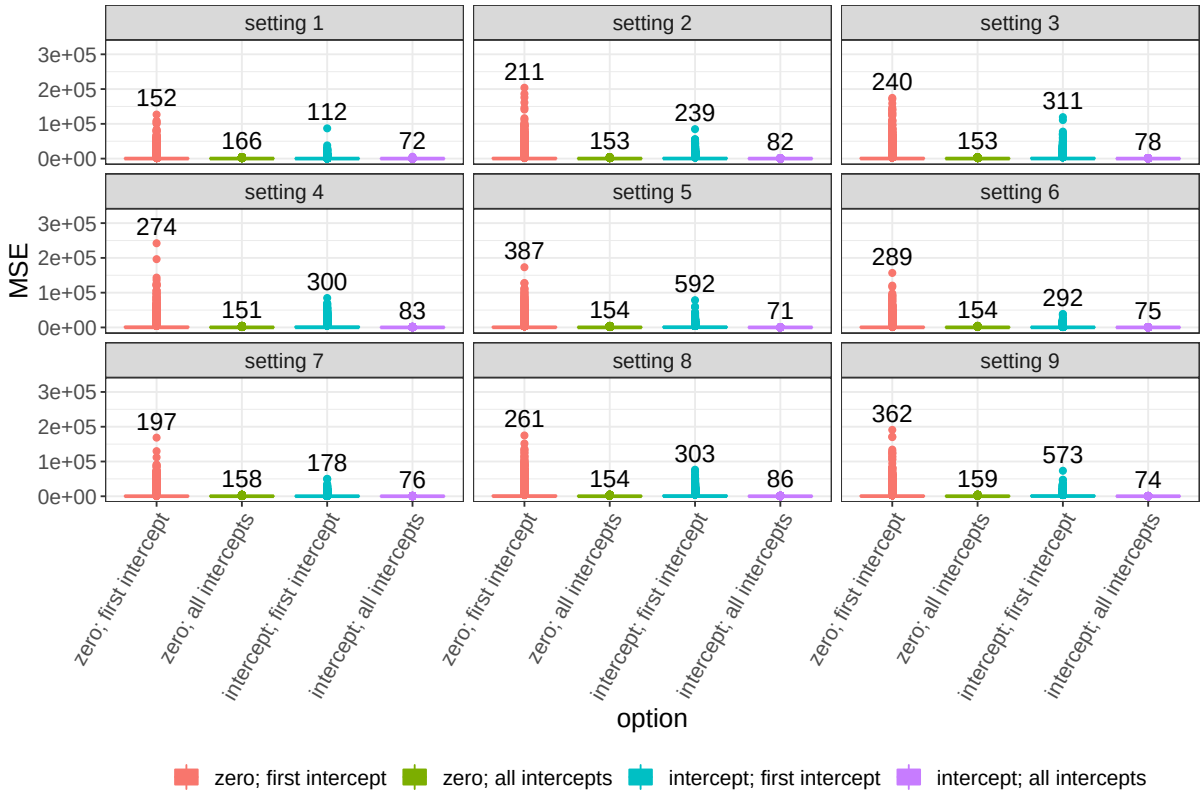

Figure 1: For every setting the MSE of predictions on complete test data with the two options for ignore missing data (use either zero or the intercept of the first block as the offset for missing observations of the first block) is shown; the MSE is either based on predictions that only use the intercept of the first block (red and blue, respectively) or that use the intercepts of all blocks (green and purple, respectively). Every box-plot summarises the results of the 100 repetitions for each missingness pattern. The median of the MSE is denoted above the box-plots.

cases. The results are shown in figures 1 and 2. For ignoring the missing data, when using the intercepts of all blocks compared to the first intercepts leads to a lower MSE in all settings expect one. Also, the variance of the MSE is considerably reduced (figure 1). The same results can be observed for the approach of imputing missing offsets using the available cases (figure 2). Therefore, in the following only predictions that use the intercepts of all blocks are considered.

4.3.2 Comparison of hyperparameters for ignore missing data and impute missing offsets with available cases

When using the approach to ignore missing data, one can chose if the offset for the second block should be zero or the intercept of the first block (given there are missing values in the first block). For the approach to impute the missing offsets using the available cases, one can determine if the blocks used for the imputation model should be chosen in a way to

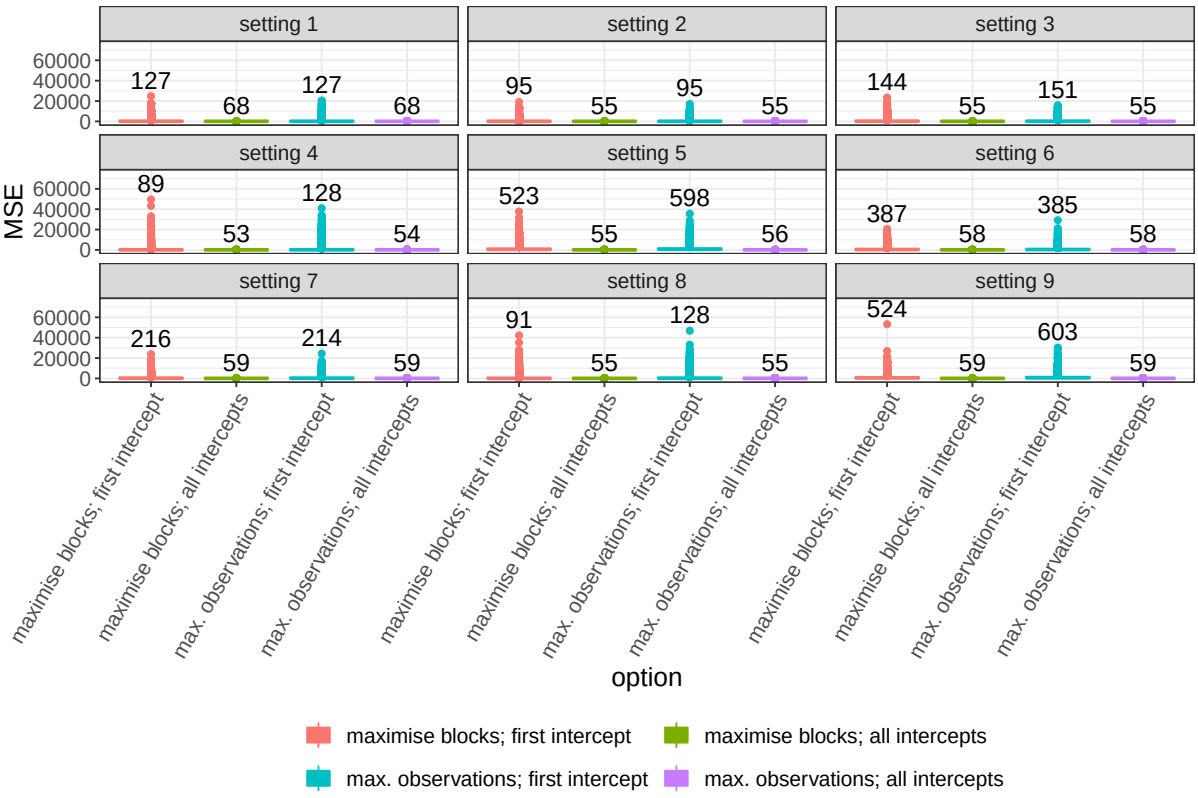

Figure 2: For every setting the MSE of predictions on complete test data with the two options for imputing missing offsets (either maximise the number of blocks used or use the maximum amount of observations) is shown; the MSE is either based on predictions that only use the intercept of the first block (red and blue, respectively) or that use the intercepts of all blocks (green and purple, respectively). Every box-plot summarises the results of the 100 repetitions for each missingness pattern. The median of the MSE is denoted above the box-plots.

maximise the block usage or to include the most observations. In order to determine which option gives better results for the respective approach, the two options were compared each in nine different settings. The correlation parameters for the settings are listed in table 1. All different missingness patterns as described in section 4.1.2 were used.

Figure 3 shows the results for predictions with ignoring missing data on complete data. For every setting, the MSE of all missingness patterns with 100 repetitions each are shown for the two different options. For all settings, the median MSE is lower when the intercept of the first block is used as the offset for the second block for missing observations. Also, the variance of the MSE for this option is lower than compared to using zero as the offset. With the exception of the setting where there is no correlation at all in the data, the range of the MSE is also smaller for the option to use the intercept as the offset. The same holds if the predictions are made with test data that has the same missingness pattern as the training data (the case where the training data is complete is not included). However, the median MSE and variance is higher compared to the prediction on complete data (data not shown).

Based on these results, the intercept of the first block is used as the offset for the second block for the ignore missing data approach in the rest of the study.

Figure 4 shows the results for predictions with imputing missing data on test data with the same missingness pattern as the training data. For every setting, the MSE of all missingness patterns with 100 repetitions each are shown for the two different options. In case that a repetition did not yield a result, the worst reported MSE for this missingness pattern was used. For all settings, the median MSE is lower or the same when the maximum number of observation option is used compared to trying to maximise the number of blocks. When the prediction is made on complete test data, the median MSE is the same for both options except for two settings, where the option to maximise the block usage is slightly better (see figure 2).

Based on these results, the option to use the maximum number of observations is used for the impute missing offsets approach when using the available cases in the rest of the study.

4.3.3 Varying the intrablock correlation

4.3.4 Influence of interblock correlation pattern

4.3.5 Permuting the block order

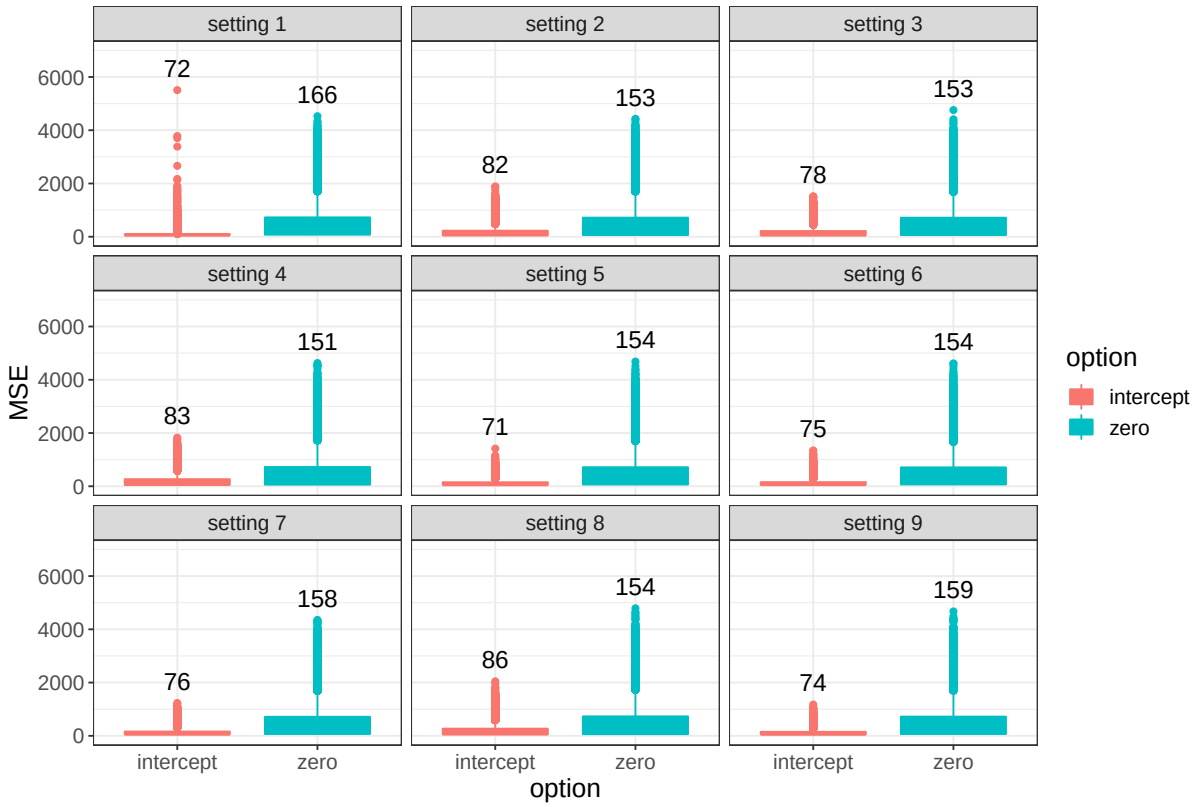

Figure 3: For every setting the MSE of predictions on complete test data with the two options for ignore missing data is shown; every box-plot summarises the results of the 100 repetitions for each missingness pattern. The predictions include the intercepts of all blocks. The median of the MSE is denoted above the box-plots.

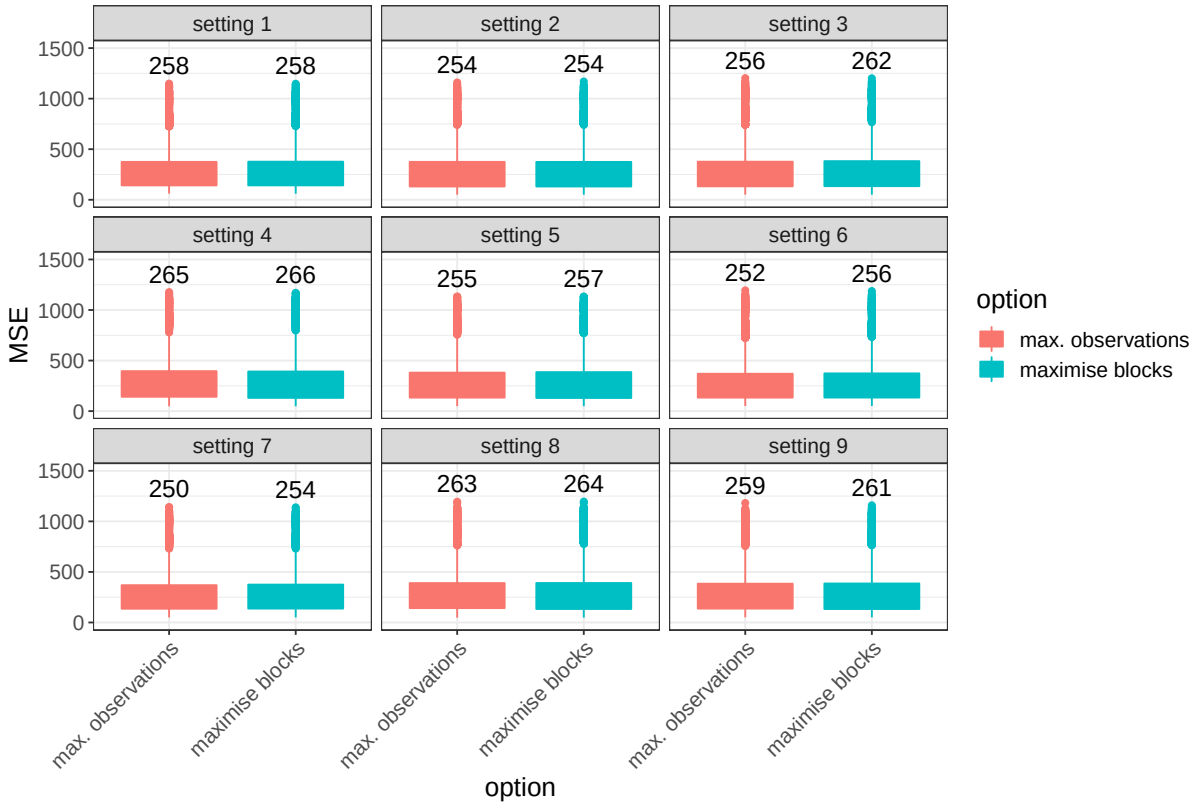

Figure 4: For every setting the MSE of predictions on test data with the same missingness structure as the training data with the two options for impute missing offsets using available data is shown; every box-plot summarises the results of the 100 repetition for each missingness pattern. In case that a repetition did not yield a result, the worst reported MSE for this missingness pattern was used. The predictions include the intercepts of all blocks. The median of the MSE is denoted above the box-plots.

Table 1: The correlation parameters for the different settings for comparing the two ignore missing data approaches

parameter	setting								
	1	2	3	4	5	6	7	8	9
ρ_C	0.0	0.5	0.8	0.8	0.8	0.5	0.5	0.5	0.5
ρ_{M_1}	0.0	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
ρ_{M_2}	0.0	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
ρ_{CM_1}	0.0	0.5	0.5	0.8	0.0	0.0	0.0	0.8	0.0
ρ_{CM_2}	0.0	0.5	0.5	0.0	0.8	0.0	0.0	0.0	0.8
$\rho_{M_1M_2}$	0.0	0.5	0.5	0.0	0.0	0.0	0.8	0.0	0.0

5 Real data application

I applied priority-Lasso to a real world data set with block-wise missing values provided by the group of Prof. Dr. med. Bianca Schaub at the paediatric clinic Dr. von Haunersches Kinderspital. The data collection was conducted in a clinical case-control study for asthma research and includes different data sources. This reflects a multi-omics situation as described in section 1. The outcome was defined as the presence of asthma in the examined children. The aim of the data analysis is to assess the suitability of these different data sources to predict an asthma diagnosis.

5.1 Data description

The data set contains 521 patients (age 5-16) with six different blocks with block-wise missing values. The block structure is detailed in table 2. Except the questionnaire, all other data sources are not present for all patients. The fraction of missing patients per block reflects the effort of generating the data, i.e. the more observations per block the easier it is to obtain this data type. The gene expression data is divided into two blocks as the data results from two different panels measured in different patients (mutually exclusive). The data set contains 256 cases and 265 controls.

From all data sources, all numeric and binary variables were included. Variables that directly include the outcome were removed from the data set (e.g. questions regarding a recent asthma diagnosis in the questionnaire). Variables containing more than 30% missing values were excluded, too. The remaining single missing values were imputed by the missForest package. The data cleaning was performed by the group of Bianca Schaub and the data set ready for the analysis provided to me.

5.2 Analysis setup

In order to get a good estimate of the generalisation error, a cross validation was performed. The data was split randomly into five folds with the same distribution of cases and controls in all folds (approximately 50:50). In the first fold, two observations were excluded from the test data because their missingness pattern was not already included in the training data. For every method, the receiver operation characteristic curve (ROC curve) and the area under the curve (AUC) are reported. A higher AUC indicates a better classification, an AUC of 0.5 is the lower boundary and can be reached by random guessing.

Priority-Lasso was used with two different approaches how to deal with block-wise missing values, priority-Lasso-ignore and priority-Lasso-impute (available). Both approaches were used with their respective two different hyperparameters and default parameters. As type measure "auc" was chosen. For the predictions of the priority-Lasso methods, the intercepts of all blocks were used. However, the prediction approach for priority-Lasso-ignore was changed. In its original form as described in section 3.3, blocks that are missing in the test data are imputed with the estimated intercept of this block. First results from the real data showed a bad performance compared to also leaving out the intercept. Therefore, the latter approach was used. A comparison is given in section 6.1.5. Additionally, mdd-sPLS with the same parameters as in section 4.2.2 was used as comparison, with the difference that when the scores from the different blocks are combined, they are weighted.

At the time of writing this thesis, Frederik Ludwigs developed approaches to deal with block-wise missing values in random forest-based models [Ludwigs, 2020]. In short, random forests combine the results from several decision trees trained on a random subset of the data. For a detailed description see Hastie et al. [2009]. I used three of his approaches as a comparison to the priority-Lasso models. (1) only use a single complete block to train the random forest and make predictions on the test data. Here, block 1 is used. (2) the block-wise approach trains a random forest on every block separately with the available observations. To make a prediction, the results of the different models are averaged and weighted by their F1 score. (3) the fold-wise approach trains a model on every missingness combination in the training data. All trees are used during the prediction of test data, however, if trees make a decision based on a variable not contained in the test data, the tree is cut and the prediction from before this decision used. The results of the different models are averaged and weighted by their F1 score. For a more detailed description see Ludwigs [2020]. All analyses were performed on the same cross validation folds as the priority-Lasso methods.

As for the random forest-based methods no AUC were recorded, the methods are

compared via their F1 score. The F1 score is the harmonic mean of the positive predictive value and the sensitivity and a value of 1 means perfect positive predictive value and sensitivity. All probabilities greater than 0.5 are predicted as positive (asthma).

Table 2: Overview of the different blocks of the real world data set. In total, 521 patients are included for which six different data sources (blocks) were recorded. The number of patients for which a block was recorded, the corresponding percentage of patients missing this block and the number of variables per block as used in the analysis are given.

block	**ID**	**number of observations**	**missing observations in %**	**number of variables**
questionnaire	1	521	0.0	44
clinical routine diagnostics	2	516	1.0	16
allergen sensitisation	3	472	9.4	19
cytokine expression data	4	149	71.4	29
gene expression data I	5	66	87.3	82
gene expression data II	6	46	91.2	84

5.3 Results

In the first step, the data analysis was performed with a block order that gives higher priority to blocks with less missing observations. The different methods are compared and the addition of blocks analysed. Afterwards, a second analysis was performed that takes the priority of the blocks according to the need of the clinicians into account.

5.3.1 Comparison of different approaches for handling missing data

Both approaches of priority-Lasso how to deal with block-wise missing data were applied to the real data set and their respective hyperparameters varied. This lead to four different approaches, for which the ROC curves are shown in figure 5. The results for priority-Lasso-impute (available) (AUC = 0.88 for maximising the blocks and AUC = 0.87 for maximal number of observations) are very similar to priority-Lasso-ignore (AUC = 0.87 for both zero as offset for missing values in the first block and the estimated intercept of the first block as offset). As a comparison, the result from the mdd-sPLS method is shown. Its AUC of 0.88 is comparable to the results from the priority-Lasso approaches.

5.3.2 Influence of using different block combinations for the prediction

Next, I examined the influence of using different block combinations for the prediction. For this, the model was trained on all blocks, but the predictions were based on a subset

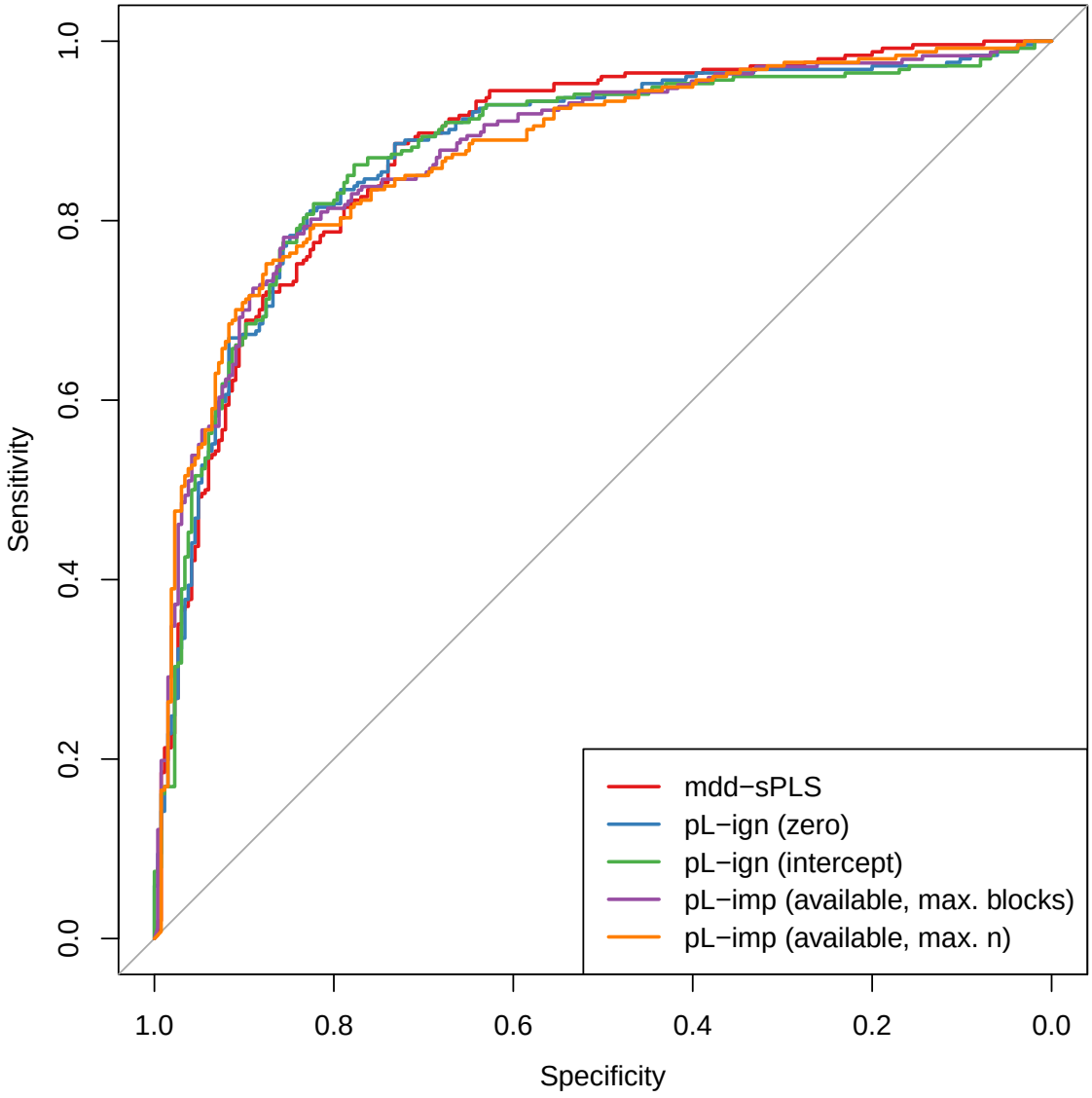

Figure 5: Receiver operation curves (ROC curves) for priority-Lasso and mdd-sPLS applied to the real data set in a five fold cross validation. The missing data was handled by priority-Lasso in four different ways: priority-Lasso-ignore (zero) (blue), area under the curve (AUC) = 0.87; priority-Lasso-ignore (intercept) (green), AUC = 0.87; priority-Lasso-impute (available, max. blocks) (purple), AUC = 0.88; priority-Lasso-impute (available, max. n) (orange), AUC = 0.87. mdd-sPLS (red) has an AUC of 0.88.

of the blocks, adding one after the other. The AUC for the different combinations are described in table 3 and are shown in figure 6.

The first block is complete and yields an AUC of 0.89 across all approaches. The addition of the next block, which only has five missing observations, improves the prediction slightly for all methods. Similarly, the addition of the third block, which has approximately 10% missing values, improves the prediction for all methods (except for priority-Lasso-impute (available, max. n) it stays the same). However, the addition of block four does not lead to an improvement for most methods, and the addition of blocks five and six, which approximately have 90% missing values, even lead to a decrease of the AUC.

In figure 7, the ROC curves of different block combinations during the prediction are exemplary shown for priority-Lasso-impute (available, max. n). As described before, the addition of the third block to blocks one and two improves the predictions (AUC of 0.91 versus 0.90). It is evident that when besides blocks one and two also block six is used for the prediction, this leads to worse predictions (AUC of 0.85). However, this effect is not as pronounced when also using all other blocks for the predictions (AUC of 0.87).

Table 3: The table shows the AUC for different priority-Lasso methods and predictions with different block combinations in a five fold cross validation. For the predictions, the results from different blocks were used. Added block 1 only uses the first block, added block 2 additionally uses block 2, so the prediction comprises the results from blocks 1 and 2, and so on. During the training, the model either used all blocks or only the blocks also used in the prediction (these methods are marked with *). Additionally, the AUC of mdd-sPLS when all blocks are used is shown.

method	**added block**					
	1	2	3	4	5	6
pL-ign (zero)	0.89	0.91	0.92	0.92	0.89	0.87
pL-ign (zero) *	0.89	0.91	0.92	0.92	0.89	0.87
pL-ign (intercept)	0.89	0.91	0.92	0.92	0.89	0.87
pL-ign (intercept) *	0.90	0.91	0.92	0.93	0.89	0.87
pL-imp (available, max. blocks)	0.89	0.90	0.92	0.92	0.89	0.88
pL-imp (available, max. blocks) *	0.89	0.90	0.92	0.91	0.89	0.88
pL-imp (available, max. n)	0.89	0.90	0.91	0.91	0.89	0.87
pL-imp (available, max. n) *	0.89	0.91	0.91	0.91	0.88	0.87
mdd-sPLS	0.88					

5.3.3 Influence of using all or a subset of blocks for training the model

When the prediction on test data is made only with a subset of all blocks, the training of the model can be done on all blocks or only on these blocks that are later used for the

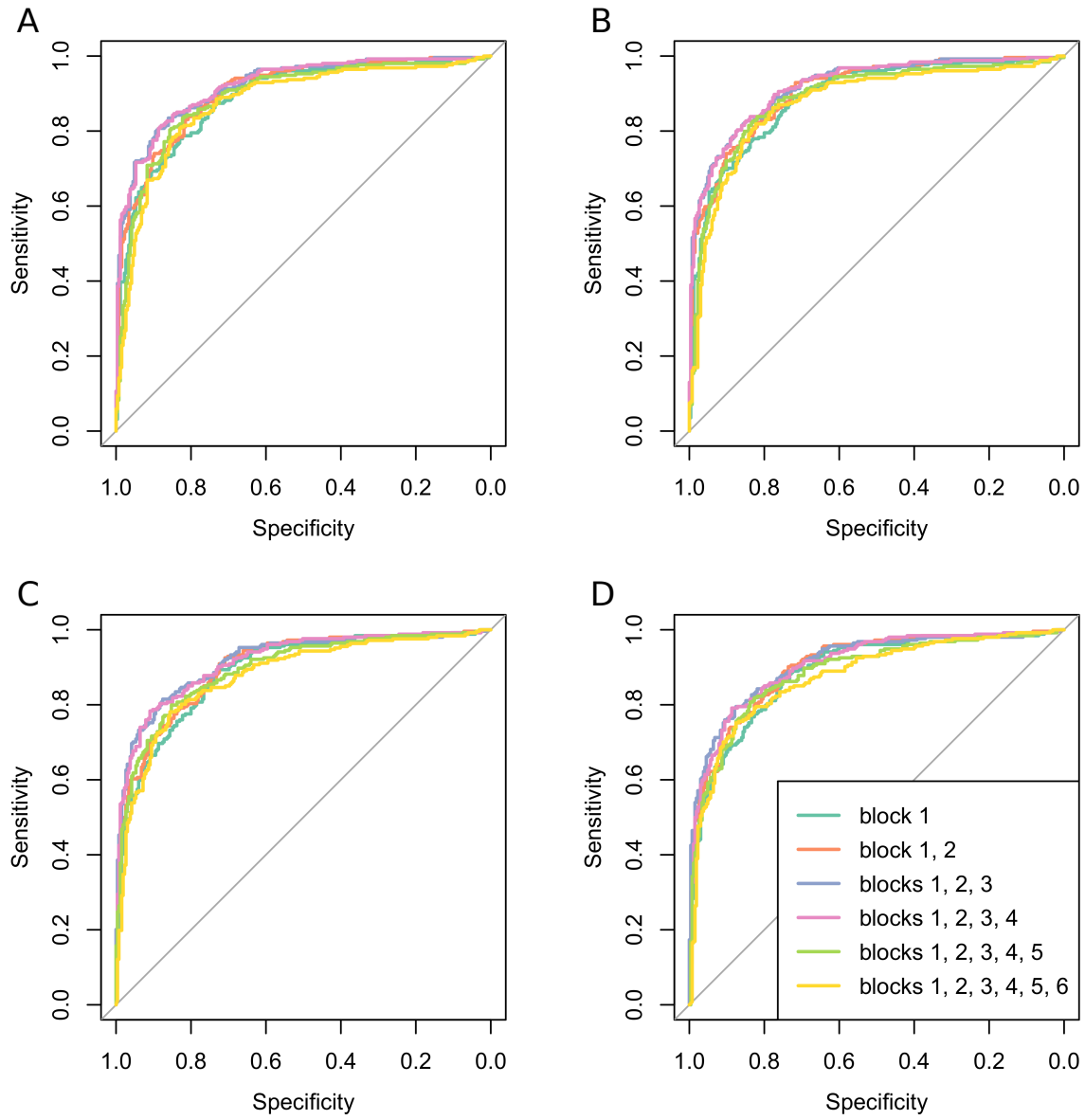

Figure 6: ROC curves for priority-Lasso applied to the real data set in a five fold cross validation. The models were trained on all blocks, but the prediction is only based on part of the blocks. In general, the prediction quality improves from only using the first block until using the first three blocks, adding more blocks leads to a decreasing AUC. A: priority-Lasso-ignore (zero); B: priority-Lasso-ignore (intercept); C: priority-Lasso-impute (available, max. blocks); D: priority-Lasso-impute (available, max. n)

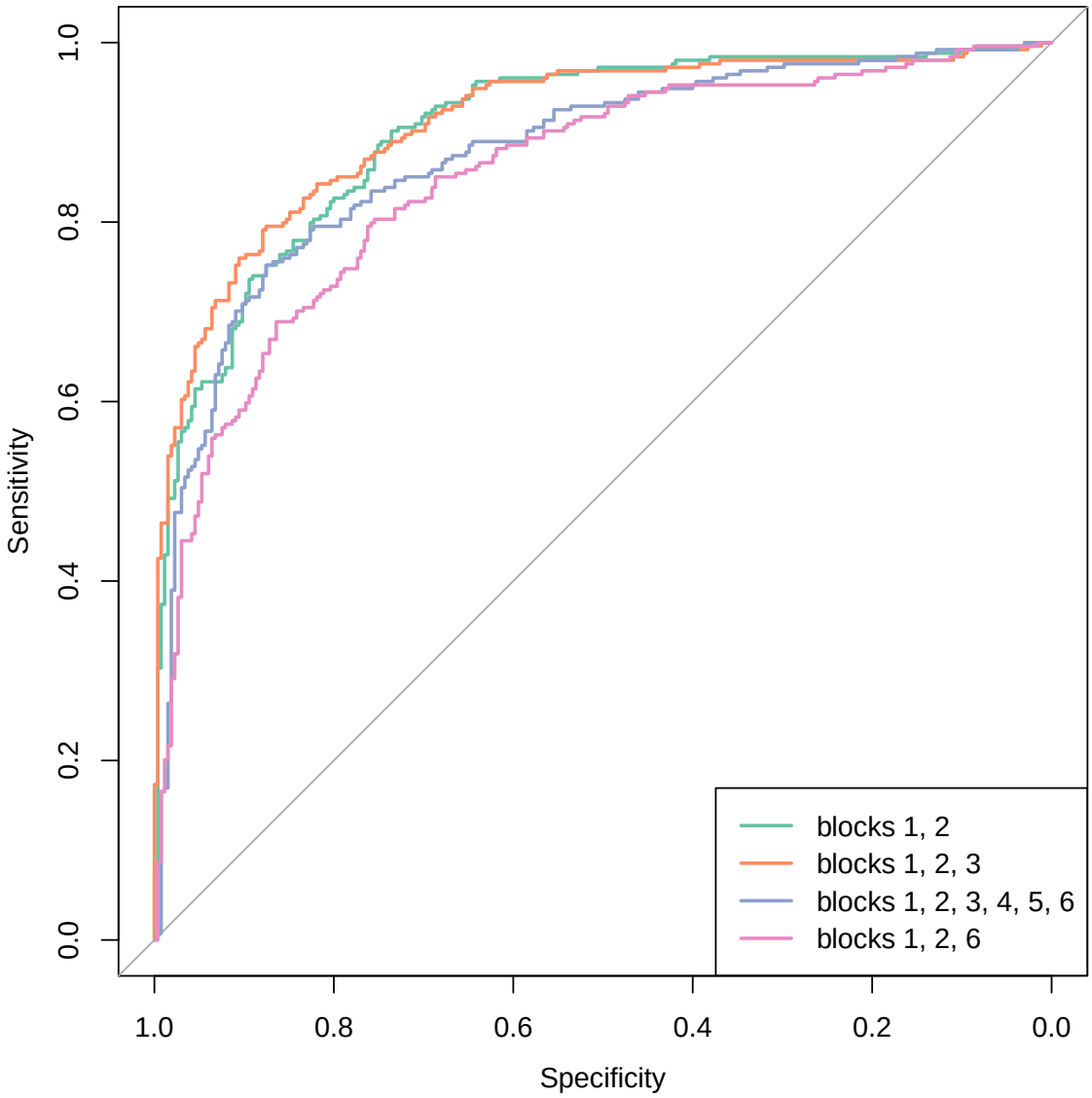

Figure 7: ROC curves for priority-Lasso applied to the real data set in a five fold cross validation. The models were trained on all blocks and missing offsets imputed using available cases and the maximum number of observations, but the prediction is only based on part of the blocks. When using blocks 1 and 2 for the prediction, which have only very few missing values, the AUC is 0.90 (green). Adding the next block, which has approximately 10% missing values, improves the prediction to an AUC of 0.91 (orange). This is better than using all blocks, which leads to an AUC of 0.87 (blue). When adding block 6, which has approximately 90% missing values, to blocks 1 and 2, the AUC decreases to 0.85 (pink).

prediction of test data. A comparison of these two different approaches is shown in table 3. The methods marked with * use only the blocks also used for the predictions of test data, the methods not marked with * use all blocks. There are no differences or at most a difference of 0.01 between the two approaches.

5.3.4 Influence of different block priorities

One key feature of priority-Lasso is that the user can set the priorities of the blocks according to their needs. In the analysis in section 5.3.1, the priorities were set according to the missingness of the data. However, this does not reflect the needs of the group that provides the data. Therefore, the analysis was repeated with the block priorities as follows (in descending order): cytokine expression data, clinical routine diagnostics, questionnaire, and lastly the allergen sensitisation, the gene expression data I and II have the same priority. With this block order, not all models with both gene expression data blocks could be estimated. Therefore, only the four block combinations where only one gene expression data block is included were analysed.

The resulting AUC from these combinations are presented in tables 4 and 5. For the different priority-Lasso methods, the predictions are shown when the predictions are based only on the block with the highest priority, based on the blocks with the two highest priorities and so on. The blocks are indicated by the ID given in table 2, the priority of the blocks during model training decreases from left to right. Additionally, the AUC for mdd-sPLS is shown. Every table shows the results from two different block combinations during training.

For all four block combinations, the predictions based on the cytokine expression data are better for the priority-Lasso-impute approaches compared to the priority-Lasso-ignore approaches. However, this difference nearly decreases when the predictions are additionally based on the clinical routine diagnostics. The same pattern is observed when the questionnaire is added to the predictions. Adding these two blocks each leads to an improvement in the prediction quality. For the combinations where the allergen sensitisation data has the next highest priority, the AUC increases slightly. In contrast, for the combinations where the gene expression data has the next highest priority the AUC decreases. For these combinations, adding the allergen sensitisation data to the prediction as the last block slightly increases the AUC, in case of using gene expression data II, the AUC for the imputing approach decreases slightly.

For the combinations where the gene expression data has the lowest priority, adding these gene expression blocks to the prediction leads to a decreased AUC, a bit more pronounced for priority-Lasso-impute than for priority-Lasso-ignore approach. When all blocks are used for the prediction, the AUC within each priority-Lasso approach is ap-

Table 4: The table shows the AUC for different priority-Lasso methods and predictions with different block combinations in a five fold cross validation. For the predictions, the results from different blocks were used. Added block 4 only uses block 4 with priority 1, added block 2 additionally uses block 2 with priority 2, so the prediction comprises the results from blocks 4 and 2, and so on. Additionally, the AUC for mdd-sPLS is shown (using all blocks). In the left part, the gene expression data I had the lowest priority during model training, in the right part the second lowest priority.

method	**added block**					**added block**				
	4	2	1	3	5	4	2	1	5	3
pL-ign (zero)	0.56	0.78	0.91	0.92	0.89	0.56	0.77	0.91	0.88	0.89
pL-ign (intercept)	0.56	0.77	0.91	0.92	0.89	0.55	0.76	0.90	0.88	0.89
pL-imp (available, max. blocks)	0.67	0.77	0.90	0.91	0.87	0.70	0.79	0.91	0.87	0.88
pL-imp (available, max. n)	0.71	0.79	0.90	0.91	0.87	0.71	0.78	0.90	0.86	0.88
mdd-sPLS			0.87					0.88		

Table 5: The table shows the AUC for different priority-Lasso methods and predictions with different block combinations in a five fold cross validation. For the predictions, the results from different blocks were used. Added block 4 only uses block 4 with priority 1, added block 2 additionally uses block 2 with priority 2, so the prediction comprises the results from blocks 4 and 2, and so on. Additionally, the AUC for mdd-sPLS is shown (using all blocks). In the left part, the gene expression data II had the lowest priority during model training, in the right part the second lowest priority.

method	**added block**					**added block**				
	4	2	1	3	6	4	2	1	6	3
pL-ign (zero)	0.56	0.78	0.91	0.92	0.90	0.55	0.78	0.91	0.88	0.89
pL-ign (intercept)	0.56	0.78	0.91	0.92	0.89	0.55	0.77	0.90	0.87	0.89
pL-imp (available, max. blocks)	0.70	0.78	0.91	0.92	0.86	0.70	0.78	0.90	0.85	0.84
pL-imp (available, max. n)	0.71	0.78	0.90	0.91	0.87	0.72	0.78	0.90	0.87	0.86
mdd-sPLS			0.90					0.89		

proximately the same not matter which priority the gene expression data has. This is true for both gene expression data sets (I, see table 4 and II, see table 5).

mdd-sPLS was trained on all five blocks and yields a comparable performance to the priority-Lasso methods. For the models with gene expression data I, the AUC is the same for the impute missing offsets approaches when all blocks are used for the prediction, but slightly worse than all priority-Lasso approaches when the gene expression data is not used for the prediction. For the models with gene expression data II, mdd-sPLS outperforms priority-Lasso-impute when all blocks are used for the prediction and is comparable to priority-Lasso-ignore. It again is slightly outperformed by all priority-Lasso methods when the gene expression data is not used for the prediction.

5.3.5 Comparison to random forest based method

The results presented in sections 5.3.2 and 5.3.4 are compared with three random forest based methods from Ludwigs [2020] in terms of the F1 score. Frederik Ludwigs provided me the results of his analysis, they are presented in table 6. Similar to the results based on the AUC, when using all six blocks for the prediction, the priority-Lasso-impute performs comparably to priority-Lasso-ignore (F1 score of 0.82). Also, the F1 score is higher when only the first three blocks are used for the prediction. Additionally, the F1 scores from the models with the block order 4, 2, 1 and 3 (descending priorities) are comparable for all methods, leading to F1 scores of 0.84-0.86.

A random forest only fitted on the complete block 1 reaches an F1 score of 0.83. The fold-wise random forest approach (using all blocks) leads to an F1 score of 0.84, which is similar to the results from priority-Lasso when the gene expression data (blocks 5 and 6) are not used for the prediction. The block-wise approach performs a bit worse (F1 score of 0.78) and gets outperformed by all priority-Lasso approaches, not matter which blocks are used for the prediction. mdd-sPLS yields an F1 score of 0.81 which is comparable to the results of the priority-Lasso methods when the predictions are based on all blocks.

Table 6: The table shows the F1 score for different priority-Lasso methods, mdd-sPLS and random forest (RF) based approaches. The block numbers are ordered with decreasing priority, blocks in brackets were used to train the model but not for the predictions. mdd-sPLS was fitted on all blocks, the RF single block approach used block 1.

method	**F1 score**	**method**	**F1 score**
blocks 1-6		blocks 4, 2, 1, 3 (and 5)	
pL-ign (zero)	0.82	pL-ign (zero)	0.85
pL-ign (intercept)	0.82	pL-ign (intercept)	0.86
pL-imp (available, max. blocks)	0.81	pL-imp (available, max. blocks)	0.85
pL-imp (available, max. n)	0.82	pL-imp (available, max. n)	0.84
blocks 1-3 (and 4-6)		blocks 4, 2, 1, 3 (and 6)	
pL-ign (zero)	0.86	pL-ign (zero)	0.85
pL-ign (intercept)	0.85	pL-ign (intercept)	0.85
pL-imp (available, max. blocks)	0.86	pL-imp (available, max. blocks)	0.84
pL-imp (available, max. n)	0.84	pL-imp (available, max. n)	0.85
mdd-sPLS	0.81		
RF single block 1	0.83		
RF block-wise	0.78		
RF fold-wise	0.84		

6 Discussion

6.1 Real data application

The application of priority-Lasso to the clinical data set shows that including block-wise missing data can help improving the asthma predictions, especially when the cytokine expression data has the highest priority. However, the gene expression data has such a high missingness fraction that it worsens the predictions and may be excluded from the analysis.

6.1.1 Influence of the missingness fraction

The results presented in sections 5.3.2 and 5.3.4 show that priority-Lasso can improve the quality of the asthma predictions by including data sources with block-wise missing values. Across all approaches the AUC improves slightly, however already the prediction only based on the questionnaire which is complete for all observations leads to a very high AUC (see table 3). This shows that the concept works but may lead only to small improves against a strong baseline. The improvement is more pronounced when the cytokine expression data and the clinical routine diagnostics have the highest priorities - both are blocks with block-wise missing values. In such a scenario, the use of several blocks with a low fraction of missing observations by priority-Lasso leads to a considerable improvement of the predictions (see tables 4 and 5).

An important observation is that the addition of blocks with a high fraction of missing values can lead to a worse prediction accuracy. This is in contrast to the original idea of priority-Lasso that the addition of blocks can only yield better predictions. The dealing with the high fraction of missing values in the gene expression data - approximately 90% of the observations have no values - leads to a decline in the prediction quality. Apparently, the available data is not sufficient to generate a good model for this block. The effect can be seen in figure 7.

6.1.2 Comparison of priority-Lasso-ignore and priority-Lasso-impute

The results from the priority-Lasso-ignore and priority-Lasso-impute approaches are basically the same. The most prominent difference is the way worse performance of priority-Lasso-ignore when the prediction is only based on the cytokine expression data as the first block (see tables 4 and 5). However, these predictions are not really comparable to priority-Lasso-impute. While the latter approach only uses the model for the cytokine expression data for the prediction, it makes use of all the other blocks for the imputation model for observations that miss this block. For observations that miss this block,

priority-Lasso-ignore does not have any information about this block so its prediction is 0.5 for these observations. Adding the clinical routine diagnostics as the second block to the prediction already cancels this effect out.

6.1.3 Influence of different block priorities

The block priorities can have an influence if a model can be estimated or not. The most prominent differences again depend on the missingness fraction of the blocks. High priority blocks with little or no missingness lead already to good predictions only based on these blocks (see table 3). When the cytokine expression data is the block with the highest priority with 75% missing observations, the prediction accuracy improves with additional blocks.

On the other hand, it is evident that the gene expression data, which has a high fraction of missing observations, worsens the prediction accuracy, no matter which priority these blocks have. One possible reason is that the Lasso model fitted by priority-Lasso suffers from the small sample size. Therefore, one may chose to not include the gene expression data in the prediction model for asthma.

6.1.4 Comparison to other methods

mdd-sPLS shows a prediction accuracy comparable to the priority-Lasso approaches, especially to priority-Lasso-ignore. The AUC of mdd-sPLS is most similar to the priority-Lasso predictions that are based on all blocks, independent of the block priorities. Predictions by priority-Lasso that are not based on the gene expression data outperform mdd-sPLS. This suggests that mdd-sPLS in the same manner as priority-Lasso struggles with the high fraction of missingness in parts of the data set.

For the random forest based methods, no AUC was recorded. Therefore the F1 score is used for comparison as presented in table 6. A random forest fitted only on the complete questionnaire block outperforms priority-Lasso fitted on all blocks where the priorities are decreasing with higher missingness fractions and the predictions based on all blocks. This again illustrates the strong baseline of the questionnaire data, as already seen by priority-Lasso predictions only based on this block.

The fold-wise random forest leads to a better F1 score than the priority-Lasso methods using all blocks for the prediction. This indicates that this random forest approach handles blocks with a very high fraction of missing observations better than priority-Lasso. In contrast, the random forest approach were one model is fitted to each block is inferior to all other methods. One possible reason is that the high fraction of missing observations of the gene expression data leads to poor models influencing the overall performance.

All in all, the random forest method outperforms priority-Lasso when the latter is used with all blocks on the asthma data set. However, exclusion of the gene expression data leads to the same or better F1 score for priority-Lasso compared to the random forest approach. Moreover, priority-Lasso is more flexible in that it can also handle continuous and survival data which the random forest method cannot. Also, the computation of the priority-Lasso methods for the asthma data set takes less than 10 minutes, whereas the random forest implementation needs two to three hours.

6.1.5 Improvement in the prediction for priority-Lasso-ignore

As mentioned in section 5.2, for this analysis the prediction for priority-Lasso-ignore models was changed. Results from predictions with the method as described in section 3.3 were inferior to priority-Lasso-impute especially when the predictions also used the gene expression data (AUC differences of up to 0.25). In the approach as described in section 3.3, missing blocks in the test data are imputed with the intercept of the model estimated for this block. As the missingness fraction for these blocks is very high, the models for the gene expression data are rather poor. It seems that the addition of the intercepts derived from these models leads to the inferior prediction accuracy for the priority-Lasso-ignore methods. Thus the new prediction approach of omitting any information from the models which are missing in the test data is promising. The results of the two different prediction approaches can be found in additional table A1.

References

H. Abdi and L. J. Williams. *Partial least squares methods: partial least squares correlation and partial least square regression*, pages 549–579. Humana Press, Totowa, NJ, 2013.

A.-L. Boulesteix, R. De Bin, X. Jiang, and M. Fuchs. Ipf-lasso: Integrative-penalized regression with penalty factors for prediction based on multi-omics data. *Computational and Mathematical Methods in Medicine*, 2017:7691937, 2017.

T. Cai, T. T. Cai, and A. Zhang. Structured matrix completion with applications to genomic data integration. *Journal of the American Statistical Association*, 111:621–633, 2016.

M. S. Carro, W. K. Lim, M. J. Alvarez, R. J. Bollo, X. Zhao, E. Y. Snyder, E. P. Sulman, S. L. Anne, F. Doetsch, and H. Colman. The transcriptional network for mesenchymal transformation of brain tumours. *Nature*, 463:318–325, 2010.

T. Caulfield, J. Evans, A. McGuire, C. McCabe, T. Bubela, R. Cook-Deegan, J. Fishman, S. Hogarth, F. A. Miller, and V. Ravitsky. Reflections on the cost of" low-cost" whole genome sequencing: framing the health policy debate. *PLoS Biology*, 11:e1001699, 2013.

F. Crick. Central dogma of molecular biology. *Nature*, 227:561–563, 1970.

R. De Bin, A.-L. Boulesteix, A. Benner, N. Becker, and W. Sauerbrei. Combining clinical and molecular data in regression prediction models: insights from a simulation study. *Briefings in Bioinformatics*, pages 1477–4050, 2019.

L. Fagerberg, B. M. Hallström, P. Oksvold, C. Kampf, D. Djureinovic, J. Odeberg, M. Habuka, S. Tahmasebpoor, A. Danielsson, and K. Edlund. Analysis of the human tissue-specific expression by genome-wide integration of transcriptomics and antibody-based proteomics. *Molecular & Cellular Proteomics*, 13:397–406, 2014.

L. Fahrmeir, T. Kneib, S. Lang, and B. Marx. *Regression*. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2013.

J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33:1–22, 2010.

R. Gaujoux. *doRNG: Generic Reproducible Parallel Backend for 'foreach' Loops*, 2020. URL `https://CRAN.R-project.org/package=doRNG`. version:.

T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, New York, 2009.

D. F. Heitjan and S. Basu. Distinguishing "missing at random" and "missing completely at random". *The American Statistician*, 50:207–213, 1996.

S. Hieke, A. Benner, R. F. Schlenl, M. Schumacher, L. Bullinger, and H. Binder. Integrating multiple molecular sources into a clinical risk prediction signature by extracting complementary information. *BMC Bioinformatics*, 17:327, 2016.

S. Huang, K. Chaudhary, and L. X. Garmire. More is better: recent progress in multi-omics data integration methods. *Frontiers in Genetics*, 8:84, 2017.

M. Ingalhalikar, W. A. Parker, L. Bloy, T. P. Roberts, and R. Verma. Using multiparametric data with missing features for learning patterns of pathology. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 468–475, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

K. J. Janssen, A. R. T. Donders, F. E. Harrell Jr, Y. Vergouwe, Q. Chen, D. E. Grobbee, and K. G. Moons. Missing covariate data in medical research: to impute is better than to ignore. *Journal of Clinical Epidemiology*, 63:721–727, 2010.

K. J. Karczewski and M. P. Snyder. Integrative omics for health and disease. *Nature Reviews Genetics*, 19:299–310, 2018.

S. Klau, V. Jurinovic, R. Hornung, T. Herold, and A.-L. Boulesteix. Priority-lasso: a simple hierarchical approach to the prediction of clinical outcome using multi-omics data. *BMC Bioinformatics*, 19:322, 2018.

S. Klau, R. Hornung, and A. Bauer. *prioritylasso: Analyzing Multiple Omics Data with an Offset Approach*, 2019. URL `https://CRAN.R-project.org/package=prioritylasso`. prioritylasso version: 0.2.2.

N. Krautenbacher. *Learning on complex, biased, and big data: disease risk prediction in epidemiological studies and genomic medicine on the example of childhood asthma*. PhD thesis, Technische Universität München, 2018.

M. Lang. checkmate: Fast argument checks for defensive r programming. *The R Journal*, 9:437–445, 2017. URL `https://CRAN.R-project.org/package=checkmate`. checkmate version:.

H. Linder and Y. Zhang. Iterative integrated imputation for missing data and pathway models with applications to breast cancer subtypes. *Communications for Statistical Applications and Methods*, 26:411–430, 2019.

M. Liu, Y. Gao, P.-T. Yap, and D. Shen. Multi-hypergraph learning for incomplete multimodality data. *IEEE Journal of Biomedical and Health Informatics*, 22:1197–1208, 2017.

H. Lorenzo, J. Saracco, and R. Thiebaut. Supervised learning for multi-block incomplete data. *arXiv preprint arXiv:1901.04380*, 2019a. URL `https://github.com/hlorenzo/ddsPLS`. ddsPLS version: 1.1.7.

H. Lorenzo, J. Saracco, and R. Thiébaut. Supervised learning for multi-block incomplete data. *arXiv preprint arXiv:1901.04380*, 2019b.

F. Ludwigs. A comparison study of prediction approaches for multiple training data sets and test data with block-wise missing values. Master's thesis, Ludwig-Maximilians-Universität München, 2020.

S. D. Markowitz and M. M. Bertagnolli. Molecular basis of colorectal cancer. *New England Journal of Medicine*, 361:2449–2460, 2009.

Microsoft and S. Weston. *doParallel: Foreach Parallel Adaptor for the 'parallel' Package*, 2019. URL `https://CRAN.R-project.org/package=doParallel`. doParallel version:.

Microsoft and S. Weston. *foreach: Provides Foreach Looping Construct*, 2020. URL `https://CRAN.R-project.org/package=foreach`. foreach version:.

M. Pertea and S. L. Salzberg. Between a chicken and a grape: estimating the number of human genes. *Genome Biology*, 11:206, 2010.

P. Probst, A.-L. Boulesteix, and B. Bischl. Tunability: Importance of hyperparameters of machine learning algorithms. *Journal of Machine Learning Research*, 20:1–32, 2019.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2019. URL `https://www.R-project.org/`. R version: 3.6.1.

D. B. Rubin. *Multiple imputation for nonresponse in surveys*, volume 81. John Wiley & Sons, New York, 2004.

J. A. Sterne, I. R. White, J. B. Carlin, M. Spratt, P. Royston, M. G. Kenward, A. M. Wood, and J. R. Carpenter. Multiple imputation for missing data in epidemiological and clinical research: potential and pitfalls. *BMJ*, 338:b2393, 2009.

V. Thorsson, D. L. Gibbs, S. D. Brown, D. Wolf, D. S. Bortone, T.-H. O. Yang, E. Porta-Pardo, G. F. Gao, C. L. Plaisier, and J. A. Eddy. The immune landscape of cancer. *Immunity*, 48:812–830, 2018.

K.-H. Thung, C.-Y. Wee, P.-T. Yap, D. Shen, and A. D. N. Initiative. Neurodegenerative disease diagnosis using incomplete multi-modality data via matrix shrinkage and completion. *NeuroImage*, 91:386–400, 2014.

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

H. Wickham, M. Averick, J. Bryan, W. Chang, L. D. McGowan, R. François, G. Grolemund, A. Hayes, L. Henry, J. Hester, M. Kuhn, T. L. Pedersen, E. Miller, S. M. Bache, K. Müller, J. Ooms, D. Robinson, D. P. Seidel, V. Spinu, K. Takahashi, D. Vaughan, C. Wilke, K. Woo, and H. Yutani. Welcome to the tidyverse. *Journal of Open Source Software*, 4:1686, 2019. URL `https://CRAN.R-project.org/package=tidyverse`. tidyverse version: 1.3.0.

S. Xiang, L. Yuan, W. Fan, Y. Wang, P. M. Thompson, J. Ye, and I. Alzheimer's Disease Neuroimaging. Bi-level multi-source learning for heterogeneous block-wise missing data. *NeuroImage*, 102 Pt 1:192–206, 2014.

F. Xue and A. Qu. Integrating multi-source block-wise missing data in model selection. *Journal of the American Statistical Association*, just accepted:1–36, 2020.

L. Yuan, Y. Wang, P. M. Thompson, V. A. Narayan, J. Ye, and A. D. N. Initiative. Multi-source feature learning for joint analysis of incomplete multiple heterogeneous neuroimaging data. *NeuroImage*, 61:622–632, 2012.

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

H. Zhu, G. Li, and E. F. Lock. Generalized integrative principal component analysis for multi-type data with block-wise missing structure. *Biostatistics*, 21:302–318, 2020.

Appendix

Electronic Appendix

XXX

Additional Figures and Tables

Table A1: The table shows the AUC of the two different priority-Lasso-ignore approaches in a five fold cross validation. For the predictions, the results from different blocks were used. Added block 1 only uses the first block, added block 2 additionally uses block 2, so the prediction comprises the results from blocks 1 and 2, and so on. The old prediction approach is the one described in the method section 3.3, the new one does not include the estimated intercept for blocks with missing values.

method	**prediction approach**	**added block**					
		1	2	3	4	5	6
pL-ignore (zero)	old	0.89	0.91	0.92	0.92	0.65	0.62
pL-ignore (zero)	new	0.89	0.91	0.92	0.92	0.89	0.87
pL-ignore (intercept)	old	0.89	0.91	0.92	0.92	0.65	0.65
pL-ignore (intercept)	new	0.90	0.91	0.92	0.93	0.89	0.87

Statutory Declaration

I declare, that I, Jonas Hagenberg, have authored this thesis independently, that I have not used other than the declared sources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Place, Date	Signature