Received: 23 November 2022 | Revised: 6 April 2023 | Accepted: 9 May 2023

DOI: 10.1002/wics.1626

ADVANCED REVIEW

WIREs COMPUTATIONAL STATISTICS WILEY

Prediction approaches for partly missing multi-omics covariate data: A literature review and an empirical comparison study

Roman Hornung[1,2] | **Frederik Ludwigs**[1] | **Jonas Hagenberg**[1,3,4,5] | **Anne-Laure Boulesteix**[1,2]

[1]Institute for Medical Information Processing, Biometry and Epidemiology, University of Munich, Munich, Germany

[2]Munich Center for Machine Learning (MCML), Munich, Germany

[3]Department of Translational Research in Psychiatry, Max Planck Institute of Psychiatry, Munich, Germany

[4]Institute of Computational Biology, Helmholtz Zentrum Müünchen, Neuherberg, Germany

[5]International Max Planck Research School for Translational Psychiatry, Munich, Germany

Correspondence
Roman Hornung, Institute for Medical Information Processing, Biometry and Epidemiology, University of Munich, Munich, Germany.
Email: hornung@ibe.med.uni-muenchen.de

Funding information
Deutsche Forschungsgemeinschaft, Grant/Award Numbers: BO3139/4-3, BO3139/6-2, HO6422/1-2

Edited by: Henry Lu, Commissioning Editor and David Scott, Review Editor and Co-Editor-in-Chief

[Correction added on 8 June 2023, after first online publication: A missing affiliation for Roman Hornung and Anne-Laure Boulesteix was added.]

Abstract

As the availability of omics data has increased in the last few years, more multi-omics data have been generated, that is, high-dimensional molecular data consisting of several types such as genomic, transcriptomic, or proteomic data, all obtained from the same patients. Such data lend themselves to being used as covariates in automatic outcome prediction because each omics type may contribute unique information, possibly improving predictions compared to using only one omics data type. Frequently, however, in the training data and the data to which automatic prediction rules should be applied, the test data, the different omics data types are not available for all patients. We refer to this type of data as block-wise missing multi-omics data. First, we provide a literature review on existing prediction methods applicable to such data. Subsequently, using a collection of 13 publicly available multi-omics data sets, we compare the predictive performances of several of these approaches for different block-wise missingness patterns. Finally, we discuss the results of this empirical comparison study and draw some tentative conclusions.

This article is categorized under:

Applications of Computational Statistics > Genomics/Proteomics/Genetics

Applications of Computational Statistics > Health and Medical Data/ Informatics

Statistical and Graphical Methods of Data Analysis > Analysis of High Dimensional Data

KEYWORDS

missing values, molecular data, multi-omics, prediction

1 | INTRODUCTION

The generation of various types of omics data is becoming increasingly rapid and cost-effective. As a consequence, there are more so-called multi-omics data becoming available, that is, high-dimensional molecular data of several types such as genomic, transcriptomic, or proteomic data measured for the same patients. In the last few years, several approaches to use these data for patient outcome prediction have been developed (see Hornung and Wright (2019) for an extensive literature review). Nevertheless, doubts have recently emerged as to whether there is benefit to using multi-omics data over simple clinical models (Herrmann et al., 2020).

Regardless of their usefulness for prediction, multi-omics data from different sources that are used for the same prediction problem, for various reasons, often do not feature the exact same types of data. Most importantly, the data for which predictions should be obtained, that is, the test data, often do not contain the same data types as the data available for obtaining the prediction rule, that is, the training data (Krautenbacher et al., 2019). The training data are also frequently composed of subsets originating from different sources (e.g., different hospitals) that frequently consist of different combinations of omics data types, as mentioned previously. When focusing on the collection of all omics data types available in at least one of the observations and considering data types not available for the different observations as missing, we can concatenate the data associated with all observations to obtain a large data set with partly missing data (cf. Figure 1 for illustrative examples). In the following, data concatenated in this form will be referred to as block-wise missing multi-omics data, where the different omics data types will be denoted as "blocks." The groups of observations in the data set that share the same combinations of observed data types will be denoted as "subsets" for simplicity. Note that in addition to the omics data, there are often clinical covariates (e.g., age or disease stage) available in practice, which usually contain plenty of predictive information. In this article, we assume that the clinical covariates are always available for all patients. This data will be referred to as the "clinical block" in the following.

We begin with a presentation of the current state-of-the-art prediction approaches applicable to block-wise missing multi-omics data, followed by an empirical benchmark comparison study of the performance of some approaches. For this study we used a large collection of publicly available multi-omics data sets, for which missing data were generated artificially. As a response variable, we used the presence versus absence of a TP53 mutation. While it is not clinically meaningful to derive a model that predicts the presence of TP53 mutations, these mutations have been associated with poor clinical outcomes (Wang & Sun, 2017). Against this background, we use TP53 as a surrogate for a phenotypic outcome in our benchmark study.

2 | EXISTING PREDICTION APPROACHES FOR PARTLY MISSING MULTI-OMICS COVARIATE DATA

Existing methods that can handle block-wise missing data can be broadly characterized into three categories: naive methods, imputation methods to be used before prediction modeling, and methods that deal with the missingness structure within the prediction modeling process. In the following, we will describe these methods, where we will go into particular detail about those methods that have been included in our empirical comparison study. Tables 1 and 2 facilitate the identification of commonalities and differences between the methods discussed below by providing key characteristics for each method. Unfortunately, for most of the methods there do not seem to exist publicly available implementations. In the following descriptions, we will mention all publicly available implementations that we identified. In addition, we implemented some of the methods in R in order to include them in our benchmark study. These implementations, which are exclusively available in the GitHub repository associated with this article (https://github.com/RomanHornung/bwm_article), will also be mentioned in the following. Hereafter, for simplicity, we will use the terms "training data" and "test data" instead of "block-wise missing multi-omics training data" and "block-wise missing multi-omics test data."

2.1 | Naive methods

The simplest method is the complete case approach. In conventional complete case analysis, all observations with missing values are removed. However, in the presence of block-wise missingness this would often be impossible or ineffective because there may be no or very few observations with all blocks observed. Therefore, to increase the number of

training observations with no missing data, it makes sense to first remove all those blocks from the training data that do not occur in the test data. When subsequently removing all observations with missing values from the training data, more complete observations can be retained. However, still a potentially important part of the training observations may be discarded with this method. In extreme cases, there may not even be any patients with complete data. For example, one study center may have collected transcriptomic and proteomic data only, while the other center has collected transcriptomic and genomic data only; none of the patients are a complete case with respect to the combination of transcriptomic, proteomic, and genomic data. Note that with the complete case approach, as with most of the approaches described below, the prediction rule can only be trained if the missing data structure of the test data is known.

TABLE 1 Overview of existing prediction approaches for partly missing multi-omics covariate data—I.

Approach	Type	Developed for multi-omics data	Parametric versus non-parametric	Prediction method type	Computational effort required
Complete case approach	Naive	No	–	Any	Low
Single block approach	Naive	No	–	Any	Low
Conventional imputation	Imputation-based	No	Depends on imputation procedure	Any	High
Semi-supervised imputation (W. Lan et al., 2022)	Imputation-based	No	Non-parametric	Any	Unclear
TOBMI (Dong et al., 2019)	Imputation-based	Yes	Non-parametric	Any	Moderate
Thung et al. (2014)	Imputation-based	No (medical data including omics data)	Semi-parametric	Using imputation, binary outcomes only	Unclear
Cai et al. (2016)	Imputation-based	Yes	Non-parametric	Any	Unclear
Linder and Zhang (2019)	Imputation-based	Yes	Non-parametric	Any	Unclear
Generalized integrative principal component analysis (Zhu et al., 2020)	Imputation-based	No	Semi-parametric	Any	Unclear
Multiple block-wise imputation (Xue & Qu, 2021)	Imputation-based	No (medical data including omics data)	Semi-parametric	Linear regression model	Moderate
mdd-sPLS (Lorenzo, Razzaq, et al., 2019; Lorenzo, Saracco, & Thiébaut, 2019)	Imputation-based	No (medical data including omics data)	Semi-parametric	Linear regression model, linear discriminant analysis	Unclear
Zhang et al. (2020)	Imputation-based	No (medical data including omics data)	Semi-parametric	Latent factor regression	Unclear
PRIME (Yang et al., 2020)	Imputation-based	No	Semi-parametric	SVM with variable selection using LASSO	Unclear
Hieke et al. (2016)	Imputation-based	Yes	Parametric	Componentwise likelihood-based boosting	Unclear (probably low to moderate)
priority-LASSO-impute (Klau et al., 2018)	Imputation-based	Yes	Parametric	Priority-LASSO	Moderate

TABLE 2 Overview of existing prediction approaches for partly missing multi-omics covariate data—II.

Approach	Type	Developed for multi-omics data	Parametric versus non-parametric	Prediction method type	Computational effort required
Ingalhalikar et al. (2012)	Missingness pattern-based	No	Non-parametric	Any	Unclear
Multi-source random forests (Ludwigs, 2020)	Missingness pattern-based	Yes	Non-parametric	Variant of random forests	High
Krautenbacher et al. (2019)	Missingness pattern-based	No (medical data including omics data)	Non-parametric	Any	Moderate
Incomplete Multi-Source Feature (iMSF) learning (Yuan et al., 2012)	Missingness pattern-based	No (medical data including omics data)	Parametric	Ensemble of regularized logistic regression models	Unclear
Incomplete Source-Feature Selection (ISFS) model (Xiang et al., 2014)	Missingness pattern-based	No (medical data including omics data)	Parametric	Ensemble of regularized linear regression models	Unclear
MMPFS (Q. Lan & Jiang, 2021)	Missingness pattern-based	No	Parametric	Ensemble of regularized logistic regression models	Unclear
Multi-hypergraph learning (MHL) method (Liu et al., 2017)	Missingness pattern-based	No	Semi-parametric	Hypergraph-based transductive classification	Low
Dong et al. (2021)	Missingness pattern-based	No	Semi-parametric	Ensemble of SVMs	Moderate
Heterogeneous Graph-based Multimodal Fusion (HGMF) (Chen & Zhang, 2020)	Missingness pattern-based	No	Semi-parametric	Graph neural network-based transductive classification	Unclear (probably high)

Another naive method is the single block approach, where a model is trained only on one block that is available in both the training and test data. This may be advantageous in situations in which a single block carries most of the available predictive information or in which the predictive information contained in the different blocks is redundant. In the first step of the single block approach, all blocks not featured in the test data are removed from the training data. Subsequently, the predictive performance of classifiers of the same but arbitrary type trained on each of the remaining blocks is measured using, for example, cross-validation. Finally, the considered classifier is trained on the block associated with the best measured predictive performance. This method has the major disadvantage that it leads to discarding a potentially important part of the predictor variables. The complete case approach and the naive method share the advantage that, for training, any prediction method suitable for high-dimensional covariate data can be used. R implementations of both approaches are available in the GitHub repository associated with this article (https://github.com/RomanHornung/bwm_article).

2.2 | Imputation-based methods

A more sophisticated approach to dealing with missing data than the naive approaches is to impute the missing values using a data-driven procedure. An important advantage of the imputation approach over the complete case approach described above is that no observations need to be excluded when training the prediction rule. Note that some of the

methods discussed in this section assume that all covariates are continuous. However, these methods can still handle categorical covariates by dummy coding them. This means creating a separate binary variable for each category and assigning a value of 1 or 0 to each observation to indicate whether it belongs to that category or not (except for a reference category). The imputed values will generally be continuous, which is why they cannot be interpreted as those of categorical covariates. However, this is not a problem if the sole purpose is prediction.

In the simplest form, imputation methods can be used to impute the missing values in the same ways as performed in conventional missing data imputations. Here, missForest (Stekhoven & Bühlmann, 2011) or a semi-supervised learning approach (W. Lan et al., 2022) may be used. With these methods, as with those presented in the last subsection, any prediction method suitable for high-dimensional covariate can be used. It is in principle possible to use also multiple imputation methods with this most simple approach. In contrast, all other approaches discussed in this subsection perform single imputation. Multiple imputation generates multiple plausible values for each missing value based on the observed data. The resulting multiple imputed data sets are then analyzed separately and the results are combined.

While, as discussed in the last paragraph, it is possible to use conventional imputation methods to estimate the missing values, such methods may not be optimal for block-wise missing multi-omics data. This is because block-wise missingness patterns are quite different from conventional missingness patterns. With block-wise missing data, the observations can be divided into a limited number of groups, with a different combination of blocks observed in each group. In contrast, conventionally the missingness is more diffuse in the sense that each observation may have a different combination of covariates missing. In addition, because entire blocks representing different data types are missing in block-wise missing data, the missing values may differ more from the observed values of the same observations than in standard missingness patterns.

The TOBMI (Dong et al., 2019) approach is based on the popular k-nearest neighbor imputation principle (Beretta & Santaniello, 2016). TOBMI was designed for situations with two (omics) blocks A and B, where block A is available for all observations and block B is missing for part of the observations. First, the Mahalanobis distance matrix M between all observations is calculated, where importantly only the measurements from block A are used. Subsequently, the data of each observation i with missing data from block B are imputed in the following way: (1) Determine the k observations in the subset of the observations with measurements in block B that are closest to i according to M (note again that the latter distance matrix was obtained only using block A). These k observations likely behave similar to i and thus are used for imputing the missing values of i in the second step; (2) Impute each missing block B value in i by its weighted mean across the k nearest neighbors determined in step 1. As weights, the reciprocals of the Mahalanobis distances from i are used. For k, the (rounded down) square root of the number of observations with measurements in block A and B is used. Although TOBMI was only intended for situations with no more than two blocks, it can easily be applied to situations with more than two blocks; see Section 3.2.1 where we describe the configuration used for TOBMI in our empirical comparison study. Note again that with this procedure, there needs to be at least one block that is available for all observations because we need at least one block that can be used to calculate the Mahalanobis distances between all observations (i.e., matrix M). This should be fulfilled in practice in most cases because usually clinical information is available for all patients. The original version for two blocks is implemented in R and is available on a GitHub repository by the authors of Dong et al. (2019) (https://github.com/XuesiDong/TOBMI). We provide an accelerated version of this code, also applicable to situations with more than two blocks, in the GitHub repository accompanying the current article (https://github.com/RomanHornung/bwm_article).

Several methods use matrix completion algorithms to impute missing values. In Thung et al. (2014), first, the training data is reduced in size by removing noisy and redundant covariates and selecting observations that well represent the observations in the test data. Second, after removing the same covariates from the test data, the missing data in the combined training and test data are imputed, including the missing outcome values in the test data. Cai et al. (2016) assume an approximately low rank matrix for block-wise missing data and propose a structured matrix completion algorithm. This is the basis for Linder and Zhang (2019) who also allow for missing values in individual covariates in addition to the block-wise missingness structure.

Other approaches for imputation include Zhu et al. (2020). They assume that the data are realizations from exponential families and estimate the parameters of the distributions of the missing values using principal component analysis-based techniques, where they take relations within and between omic types into account. The empirical studies presented in Zhu et al. (2020) suggest that the performance of their approach tends to be lower in situations with more than two blocks. Moreover, it is only applicable if there are complete observations that feature values for all blocks. The approach is implemented in R and available on a GitHub repository by the authors of Zhu et al. (2020) (https://github.com/zhuhuichenecho/GeneralizedIntegrativePCA). In the multiple block-wise imputation approach (Xue & Qu, 2021),

the data are divided into disjoint groups based on the missingness pattern, the missing values are imputed multiple times, and the results aggregated. The imputed data sets are used to generate estimation equations and the different estimators are combined to yield one prediction. MI-GAN (Dai et al., 2021) learns generative adversarial networks for the different missingness patterns to multiply impute the missing values.

Multi-Block Data-Driven sparse partial least squares (mdd-sPLS) (Lorenzo, Razzaq, et al., 2019; Lorenzo, Saracco, & Thiébaut, 2019) is a complex method based on the partial least squares (PLS) technique. Both the training data and the test data are imputed, where these imputations are performed separately. After imputation, there are no missing values in the training and test data, which is why all blocks can be used for prediction. Put simply, mdd-sPLS performs PLS-type procedures on the blocks separately and predicts the outcome using a specific function of the (imputed) covariate values and the estimated parameters. Note, however, that although the PLS-type procedures are performed separately on the blocks, information is shared between the blocks by estimating also block-unspecific parameters in these procedures. The missing values in the training and test data of those covariates used by mdd-sPLS in prediction are imputed using additional PLS-based models. The missing values are imputed using the outcome and the other covariates respectively. These two steps, building the prediction model and imputing missing values, are repeated until convergence of the involved latent variables. The approach is implemented in the R package "ddsPLS," which was previously available on the CRAN network, but is currently only available as an archived version (https://cran.r-project.org/src/contrib/Archive/ddsPLS/). We have also included it in the GitHub repository that accompanies this article (https://github.com/RomanHornung/bwm_article).

Some methods make use of latent structure for the imputation. Zhang et al. (2020) estimate a factor model that is then used to impute missing values. Yang et al. (2020) first learn the pairwise mappings between each data type A and the remaining data types using autoencoders. Second, these pairwise mappings are regularized to be consistent with each other, each pairwise mapping is applied to A imputing the missing values, and finally the imputations obtained using the different pairwise mappings are averaged.

Hieke et al. (2016) build different penalized regression models and use the predictions from one model as the offset for the next model. For observations that do not feature predictions from one model due to missing values, only the offset is imputed and not the missing values themselves. This method is only applicable in situations with two blocks, but a potential extension to more blocks is mentioned in Hieke et al. (2016). The authors of Hieke et al. (2016) provide R code that reproduces part of their analysis as a supplement to their article (https://doi.org/10.1186/s12859-016-1183-6), demonstrating how their approach can be applied in R.

The same idea is used and generalized in priority-LASSO-impute (pL-imp (available)). This variant of priority-LASSO (Klau et al., 2018) is implemented in the R package "prioritylasso" available on the CRAN network (Klau et al., 2023). Priority-LASSO is a multi-omics prediction method based on the Least Absolute Shrinkage and Selection Operator (LASSO) (Tibshirani, 1996) that allows researchers to specify a priority ranking of the blocks. A priority ranking is often given because some blocks are more established or easier to obtain than others. The first step in obtaining the priority LASSO prediction rule is to fit a LASSO model using only the covariates in the block with highest priority. In the second step, a LASSO model is fitted again, but this time using only the covariates in the block with the second highest priority, and using the linear predictor from the LASSO model fitted in the first step as an offset in the model equation. By including this latter offset, only that part of the predictive information contained in the block with second-highest priority that is not contained in the block with highest priority is used. This process is continued iteratively for all blocks in order of priority, thus obtaining estimated model coefficients for all covariates.

In the case of block-wise missing multi-omics data, this estimation scheme would not be applicable, as in each step the offsets would only be available for those observations for which the respective block is available. With priority-LASSO-impute, the missing offsets are imputed using a LASSO model. Roughly speaking, in each step, this approach first learns a LASSO model for each subset using observations for which the offsets are available. In this LASSO model, the offsets are the response variables and the other blocks contain the covariates. The fitted model is then used to predict the missing offsets.

2.3 | Methods that deal with the missingness pattern

A different approach is to divide the data into groups based on the missingness pattern and then deal with this structure without imputing missing values. One of the earliest methods of this kind was devised by Ingalhalikar et al. (2012) who

divided the data into subsets so that every subset has complete observations for a specific combination of blocks. On each of these subsets a model is trained and the predictions are weighted inversely by their expected error.

A similar approach are multi-source random forests, described and evaluated in a smaller-scale empirical study (Ludwigs, 2020). With this method, separate random forests are learned on subsets of the data where each contains observations that feature a particular combination of blocks. In the prediction phase, first, each tree in the forests is pruned in the following way: Starting with the first split that divides the full (bootstrapped or subsampled) data set, each branch in the tree is followed, and for each split encountered, it is checked whether that split uses a covariate available in the test data, and if not, the branch is cut. This ensures that the trees in the forests only use covariates that are available in the test data. Then, similar to Ingalhalikar et al. (2012), each forest is applied to the test data and each prediction is weighted proportionally to the out-of-bag AUC value of that forest. This approach is implemented in the R package "multisForest" available on GitHub (https://github.com/RomanHornung/multisForest).

A similar idea is used in Krautenbacher et al. (2019) where a model is trained on each block. A separate classifier is trained for each block, using all observations that contain measurements for the corresponding block in the training data. To obtain predictions for the test data, first the corresponding classifiers are applied to each block available in the test data, and in each case a predicted probability for $Y = 2$ is calculated, where $Y \in \{1,2\}$ describes the binary response. Second, a weighted average of the predicted probabilities obtained for each test data block is computed, with weights proportional to the cross-validated AUC values of the corresponding random forests. An R implementation of this approach is available in the GitHub repository associated with this article (https://github.com/RomanHornung/bwm_article).

The methods described above in this subsection learn models independently on subsets of the training data. There are, however, also several methods which share information in the learning of models obtained on different subsets of the data. This has the advantage of increasing the robustness of the individual models. Yuan et al. (2012) proposed the incomplete Multi-Source Feature (iMSF) learning method. First, the data are divided into disjoint subsets, where each subset contains observations that feature a particular combination of blocks (and no further blocks). Subsequently, a regularized regression model is fitted on every subset. These models are constrained in such a way that in all models that share a block, the same covariates within this block have to be selected. However, the coefficients for one block do not need to have the same values. In contrast to the methods discussed in the previous paragraph, for iMSF, it is not necessary to apply different models to the test data and obtain a weighted prediction. Instead, to obtain predictions the model with the correct block combination is applied. For example, suppose the training data featured blocks A, B, and C. In this case, using the training data, a separate model is learned using each possible combination, that is, using the following combinations: $\{A,B\}$, $\{A,C\}$, $\{B,C\}$, and $\{A,B,C\}$. Suppose now that a test observation is missing the block B, thus it only features the blocks A and C. In that case, to obtain a prediction, the model learned on the combination $\{A,C\}$ is applied to the test observation.

The iMFS method was developed further by Xiang et al. (2014) to the incomplete Source-Feature Selection (iSFS) model. Again, different subsets are formed based on the missingness patterns, but in contrast to for iMFS these subsets are overlapping. Each subset is defined as all observations that feature all blocks from a particular combination of blocks, but potentially also other blocks. For example, if there are three blocks A, B, and C, the subset corresponding to the combination A with C contains all observations that feature only A and C, but also all observations that feature B in addition to A and C. This procedure has the advantage that more observations are considered per combination compared to in the case of iMFS. Moreover, in contrast to iMFS, not a separate coefficient vector for each combination is learned. Instead, a common coefficient vector is learned for the variables from all blocks and to obtain the coefficient vectors for the individual combinations, the entries in the common coefficient vector are multiplied by block- and combination-specific weights. This has the advantage that less parameters need to be estimated for the iSFS, which should lead to more stable models. Q. Lan and Jiang (2021) reformulate the iMSF method as a multi-task learning problem in their method MMPFS.

Liu et al. (2017) again divide the observations into subsets based on the missingness pattern in a similar way to iMSF, where, however, the available data are exploited better than for iMSF. Then a hypergraph is learned on each subset. Hypergraphs are graphs for which each edge can connect more than two vertices. This allows to model high-order relationships. In Liu et al. (2017), the subjects in the data represent the vertices in the hypergraphs. Thus, the purpose of the hypergraphs here is to capture high-order relationships between the subjects. The different hypergraphs are combined and used to train a transductive classifier. In Dong et al. (2021), a similar approach is used. In contrast to Liu et al. (2017), the hypergraphs are learned on a low-rank representation of the data. Afterwards, for every block, a support vector machine classifier is trained and the predictions are combined. Another method working with hypergraphs

is Heterogeneous Graph-based Multimodal Fusion (Chen & Zhang, 2020). The data are first divided into subsets based on the missingness pattern and heterogeneous hypernode graphs are constructed. On every graph, the relationships between different data types are learned and this information is used to construct a new hypergraph. Then, interactions between the different missingness patterns are learned and the information of the different data types are fused into one embedding.

Note that almost all of the methods described in this and the previous subsections need to be reapplied and the prediction rule retrained for varying test data sets that do not have data for all the blocks observed in the training data. This is because most methods do not allow for missing values in the test data. For these methods, blocks that are not present in the test data must be removed from the training data before the procedures are reapplied. Thung et al. (2014) do allow missing values in the test data, but the missing data in the training and test data are imputed jointly, which is why retraining is also required here. For mdd-sPLS (Lorenzo, Razzaq, et al., 2019; Lorenzo, Saracco, & Thiébaut, 2019), separate imputation is performed on both training and test data. However, since the mdd-PLS procedure involves predicting the outcome, it needs to be reapplied as well when new test data are received.

Multi-source random forests (Ludwigs, 2020), iMSF (Yuan et al., 2012), and MMPFS (Q. Lan & Jiang, 2021) are the only methods that do not need the prediction rule to be retrained when new test data sets with missing values are obtained. As described above, with multi-source random forests the trees are pruned to use only covariates that are available in the training data. With iMSF and MMPFS, the models learned on the training data that are (most) consistent with the specific combination of blocks in the test data are applied (see above for details).

3 | EMPIRICAL COMPARISON STUDY OF APPROACHES FOR PARTLY MISSING MULTI-OMICS COVARIATE DATA

3.1 | Neutrality disclosure

Our study is intended as "neutral" in the sense that we are focusing on the comparison rather than promoting a particular new method (Boulesteix, Binder, et al., 2017; Boulesteix, Wilson, & Hapfelmeier, 2017). However, we are not equally familiar with all methods. In particular, two of the included methods, multi-source random forests and priority-LASSO-impute, were developed by some of the authors of this article. While our familiarity with these methods helped us to set them up appropriately, we were committed to providing a fair comparison, that is, we neither spent more efforts to optimize these two methods than the other methods nor did we design the study to favor one or the other method.

3.2 | Design of the comparison study

3.2.1 | Configurations of the compared approaches

In this subsection, we describe the configurations and implementations used for the approaches compared in the empirical study. When selecting these approaches out of the methods described in Section 2 we ensured that at least two methods from each of the three categories described in Sections 2.1–2.3 were included. Moreover, we selected only methods that are either implemented in publicly available R packages or could be implemented with reasonable effort.

The following approaches were considered in the study: complete case approach, single block approach, imputation based on TOBMI, mdd-sPLS, block-wise random forest, multi-source random forest, and priority-LASSO-impute. As described in Section 2.1, the first two of these are naive approaches. These served as a baseline against which the other more sophisticated methods were compared. Three of the other methods were imputation-based (imputation based on TOBMI and mdd-sPLS, as well as priority-LASSO-impute, see Section 2.2) and two methods deal with the missingness pattern without imputation (block-wise random forest and multi-source random forest, see Section 2.3).

Preliminary remarks on the use of random forests in the compared approaches

With the exception of mdd-sPLS, multi-source random forests and priority-LASSO-impute, all the approaches compared allow the use of any classifier for prediction. To make the results of the different approaches more comparable, we used

random forests as classifiers for all these approaches. Random forests are known to perform well in a wide range of tasks, providing robust and accurate predictions in many applications.

When using random forests for classification tasks, the response classes most frequently predicted by the individual trees in the forest are typically used as predictions. However, we needed class probability predictions as we evaluated performance not only in terms of accuracy, but also in terms of Brier score and area under the receiver operating characteristic curve (AUC). In random forests, probability predictions for each class are obtained by averaging the proportions of observations belonging to that class in the predicted leaf nodes across all trees in the forest. For (binary) classification, we used the class with the higher predicted probability.

The random forests were constructed using the R package "randomForestSRC" (version 2.9.2). The values of the tuning parameters were set to the default values provided by "randomForestSRC." For instance, the number of covariates sampled for each split, denoted *mtry*, was set to the square root of the total number of covariates, and each forest consisted of 500 trees.

Complete case approach (`ComplcRF`)

The complete case approach as described in Section 2.1 was applied.

Single block approach (`SingleBlRF`)

Because we used random forests as classifiers, it was not necessary to perform cross-validation to measure the performance of the classifiers trained on each block. Instead, it was possible to use the out-of-bag predictions (Breiman, 2001) of the random forests as out-of-sample class probability predictions. The AUC was used as a performance measure for selecting the block used for training the final random forest classifier.

Imputation approach based on TOBMI (`ImpRF`)

Since, as described in Section 2.2, TOBMI is not applicable to general block-wise missingness patterns, we proceeded as follows to impute the training data in our comparison study. We first concatenated all those blocks that featured no missing values for any of the observations. Subsequently, we used TOBMI repeatedly, each time for imputing the values of a different block with missing values. Here, the concatenation of the blocks without missing values took the role of "block A" from the previous paragraph and the block to impute at the current repetition took the role of "block B." The imputation was performed across subsets. Subsequently, we removed all blocks in the training data not available in the test data set and constructed a random forest using the training data processed in this way.

Multi-block data-driven sparse PLS (mdd-sPLS) (`MddsPLS`)

We used 10-fold cross-validation to determine the optimal regularization parameter for the correlation matrices, performing a grid search on 10 values. Moreover, we used one component for the involved matrix decomposition and inversely weighted the components per block by the number of selected covariates per block. See Lorenzo, Saracco, and Thiébaut (2019) for details.

Block-wise random forest (`BlwRF`)

Similar to the `SingleBlRF` case, because we used random forests as classifiers, it was not necessary to perform cross-validation to calculate the AUC values that serve as weights for the predicted probabilities obtained using the block specific classifiers. Instead, we used the out-of-bag estimated AUC values.

Multi-source random forest (`MultisRF`)

Due to the comparably high computational burden associated with this approach we used only 250 trees per forest instead of 500 trees as in the cases of the other random forest-based methods. For the remaining tuning parameter values, we used the default values from the R package "randomForestSRC" (version 2.9.2).

priority-LASSO-impute (pL-imp (available)) (`PrLasso`)

As indicated in Section 2.3, the pL-imp (available) algorithm was not designed to handle missing blocks in the test data. In order to deal with this issue, we excluded all blocks that were not available in the test data before fitting the model to the training data. Moreover, the pL-imp (available) algorithm, similar to the original priority-LASSO algorithm, requires the user to provide a priority ranking of the available blocks. As no useful biological information was available for the data sets considered in the comparison study performed for this article, the priority rankings were determined

in the following way: (1) Fitting a LASSO model to each block and estimating the deviance associated with each model using 5-fold cross-validation (CV); (2) Assigning the highest priority to the block associated with the lowest cross-validated deviance value, the second highest priority to the block associated with the second-lowest cross-validated deviance value, and so on; in cases in which two or more blocks were associated with the same cross-validated deviance value, the priority order between these blocks was assigned randomly.

The shrinkage parameters in the LASSO models involved in the priority-LASSO and priority-LASSO-impute estimation procedures were determined using grid search and 10-fold CV.

3.2.2 | Data

The data material consists of 13 publicly available multi-omics data sets from The Cancer Genome Atlas (TCGA) project (Weinstein et al., 2013). These data are a subset of 21 data sets previously used in Hornung and Wright (2019). From these 21 data sets, 18 contained all four omics blocks that were considered as covariates (see below), the clinical block and the response variable, that is, the presence versus absence of the TP53 mutation. From the remaining 18 data sets, we removed imbalanced data sets for which the smaller response variable class was represented by less than 15% of the observations. This resulted in 13 data sets for use in the comparison study. The covariates consisted of the following four blocks: clinical block, copy-number variation block, miRNA block, and RNA block. There were no missing values in these data. Table 3 gives an overview of the used data sets. The clinical covariates available differed slightly across the data sets. Most data sets provided information on patient age, gender, and race, while many also included details on tumor stage. Moreover, the BRCA and LUSC data sets included cancer-specific variables. In Section A of the Supplementary Materials, we provide a detailed overview of which covariates were available for each data set.

3.2.3 | Generation of block-wise missingness and performance evaluation

Block-wise missingness patterns are generated by randomly deleting parts of the data sets. First, the data sets are split into training and test data in the ratio 3:1. Second, as described in more detail in the next paragraph, the block-wise missingness patterns are induced separately in training and test data, where there are five different patterns for the training data sets and four for the test data sets, see Figure 1. In the following, training data block-wise missingness patterns will be abbreviated as "trbmp" and test data block-wise missingness patterns as "tebmp," respectively. As is

TABLE 3 Overview of the data sets.

Label	*n*	Prop. TP53	clin	cnv	mirna	rna
BLCA	310	0.49	4	57,964	825	23,081
BRCA	863	0.34	8	57,964	835	22,694
COAD	350	0.56	5	57,964	802	22,210
ESCA	121	0.83	4	57,964	763	25,494
HNSC	411	0.69	5	57,964	793	21,520
LGG	454	0.46	3	57,964	645	22,297
LIHC	298	0.29	4	57,964	776	20,994
LUAD	424	0.49	6	57,964	799	23,681
LUSC	365	0.85	7	57,964	895	23,524
PAAD	142	0.63	4	57,964	612	22,348
SARC	183	0.36	2	57,964	778	22,842
STAD	284	0.47	6	57,967	787	26,027
UCEC	503	0.36	3	57,447	866	23,978

Note: The second column shows the number of observations. The third column shows the proportion of observations with TP53 mutation. The fourth to the seventh column show the numbers of covariates in the respective blocks.

Abbreviations: clin, clinical covariates; cnv, copy-number variation; mirna, miRNA; rna, RNA.

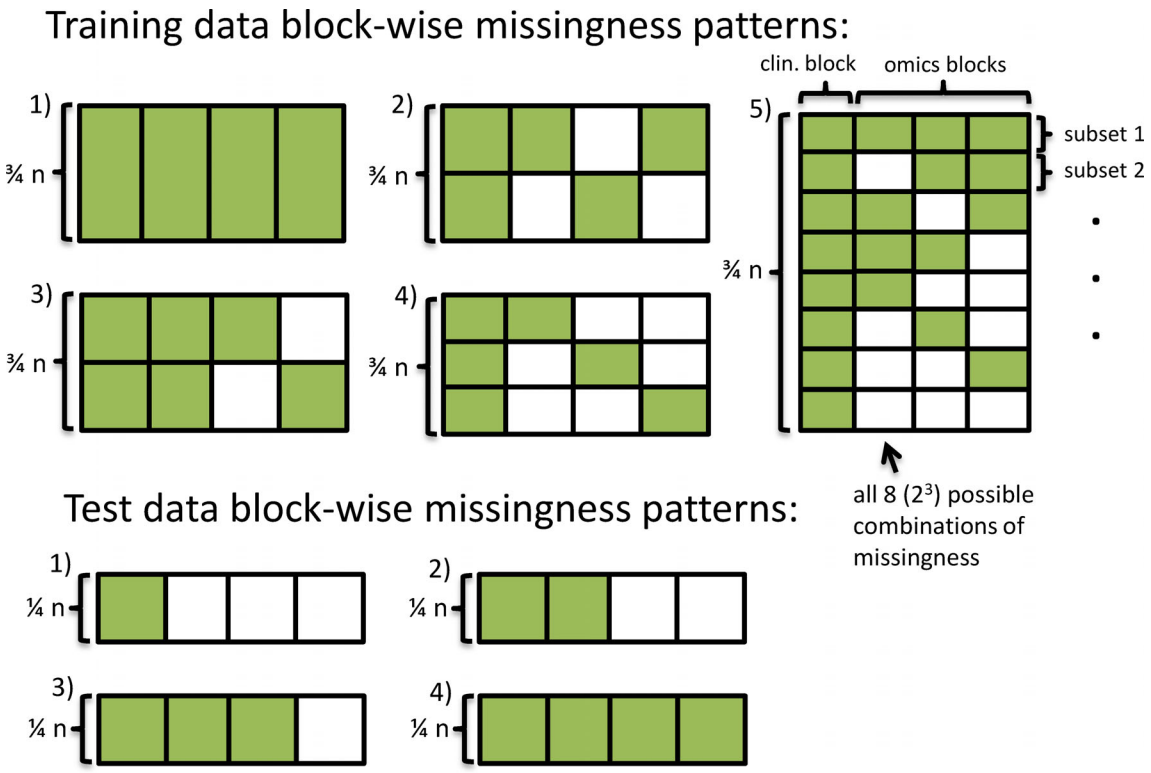

FIGURE 1 Block-wise missingness patterns in training and test data. Colored/empty boxes indicate that the respective blocks are present/absent. The first column always represents the clinical block.

evident from Figure 1, each of the trbmps consists of either one, two, three, or eight subsets of observations. For each combination of data set, trbmp, and tebmp, the above procedure is repeated five times.

Figure 2 illustrates the procedure for inducing block-wise missingness in the training and test data. Following each division into training and test data, the subset memberships of the training observations are assigned randomly and the subsets are of equal size for each training data set (note again that the data are split into training and test data in the ratio 3:1). We induce the missingness patterns after random permutation of the omics blocks, where a different permutation is used for each repetition. This is performed separately for the training data and the test data. First, the omics blocks are permuted randomly (the clinical block always stays at the first position). Second, values in the data matrix are deleted according to the respective considered block-wise missingness pattern (Figure 1). After having performed these steps separately for the training and test data, the blocks are re-ordered again to have the original ordering to ensure that the block ordering is the same in training and test data. Without the random permutation of the omics blocks described above, each block would have been observed with unequal frequency for the different missingness patterns. This would have made it impossible to tell whether differences in the results observed for different missingness patterns (trbmps and tebmps) are actually due to the missingness patterns or the fact that specific influential blocks are featured to different degrees in the missingness patterns. The permutation procedure ensures that different blocks are missing in the subsets for different repetitions even when considering the same trmps and tebmps. For example, consider trbmp 2 and tebmp 3; for the first division into training and test data, the first set in the training data might include RNA and miRNA data and the second set only mutation data, whereas the test data might include RNA and mutation, but no miRNA data. For the second division, the first set in the training data might include only mutation and miRNA data and the second set only RNA data, while the test data may include mutation and miRNA data, but no RNA data.

For each division, the methods described in Section 3.2.1 are learned on the training data, subsequently applied to the test data, and the performance on the test data measured according to the three considered performance metrics: the Brier score, the AUC, and the accuracy. The Brier score measures discrimination and calibration and is a proper scoring rule in the sense that it measures the accuracy of class probability predictions (e.g., probability for class B if there are two classes A and B). The accuracy is not a proper scoring rule because it does only evaluate the precision of

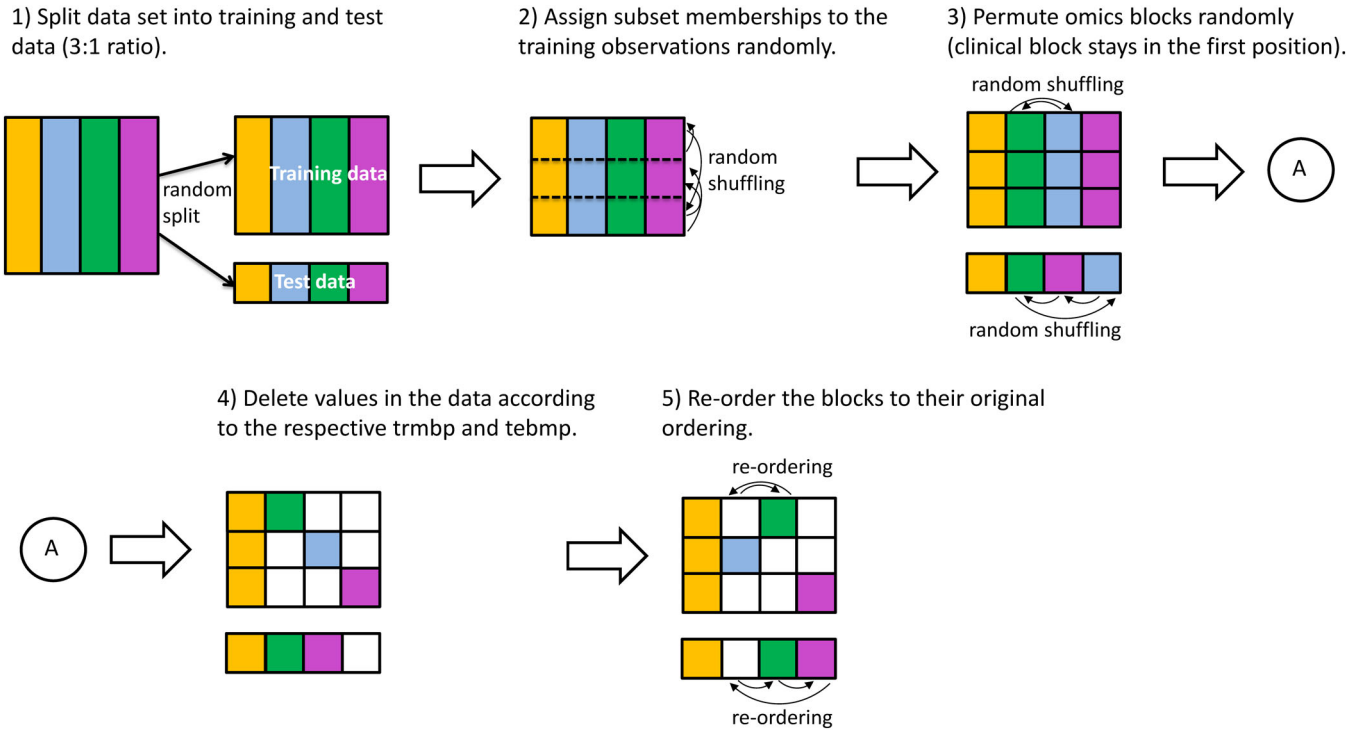

FIGURE 2 Overview of the procedure used for inducing block-wise missingness in the training and test data. Here trbmp 4 and tebmp 3 are used, but the figure would appear similar for other trbmps and tebmps.

class predictions, not class probability predictions. Thus, it uses less information from the predictions than the Brier score. The AUC only measures discrimination. More precisely, it measures the models' abilities to order different subjects correctly in terms of their predicted probabilities. Note that a model can feature a high AUC value even though it is not calibrated well. This would, for example, be the case in a situation in which the predicted probabilities output by a model are systematically too small, but still observations with larger predicted probabilities for class B tend to feature class B much more often than those with smaller predicted probabilities.

3.2.4 | Code availability

All R code written to perform and evaluate the analyses is available on GitHub (https://github.com/RomanHornung/bwm_article). The pre-processed data sets are available as Rda files on the online open access repository figshare (https://doi.org/10.6084/m9.figshare.22304050.v2). For details on the process of the pre-processing and the associated code, see Hornung and Wright (2019).

3.3 | Results

As a preliminary remark, please note that some methods delivered few or no successful predictions for specific trbmps and tebmps. For example, `ComplcRF` is not applicable for tebmp 4 when considering trbmp 2, 3, or 4: For tebmp 4, the test data do not contain missing values, which is why none of the variables in the training data are removed (cf. Section 3.2.1); because of this there are no complete cases in the training data for any of the trbmps 2, 3, or 4 (cf. Figure 1) and the complete case approach `ComplcRF` is thus not applicable. `MultisRF` is not applicable for trbmp 1 because the R package "multisForest" (version 0.1.0) implementing `MultisRF` does not allow training data sets without missing values. In addition, `PrLasso` lead to errors in rare cases. More precisely, 10 of the 1300 repetitions performed in total for `PrLasso` resulted in an error, which all but one occurred for the data set ESCA. The remaining methods did deliver predictions in all cases. The frequencies of repetitions with missing results were 25.0%, 34.1%, and

0.8% for `ComplcRF`, `MultisRF`, and `PrLasso` respectively. We describe the reasons for unsuccessful predictions in full detail in Section B of the Supplementary Materials.

3.3.1 | Global performance comparison

Figure 3 shows the ranks the methods achieved among each other, pooled across all trbmps, tebmps, and data sets. Note that here we only included those repetitions for which each of the seven considered methods delivered a result. If

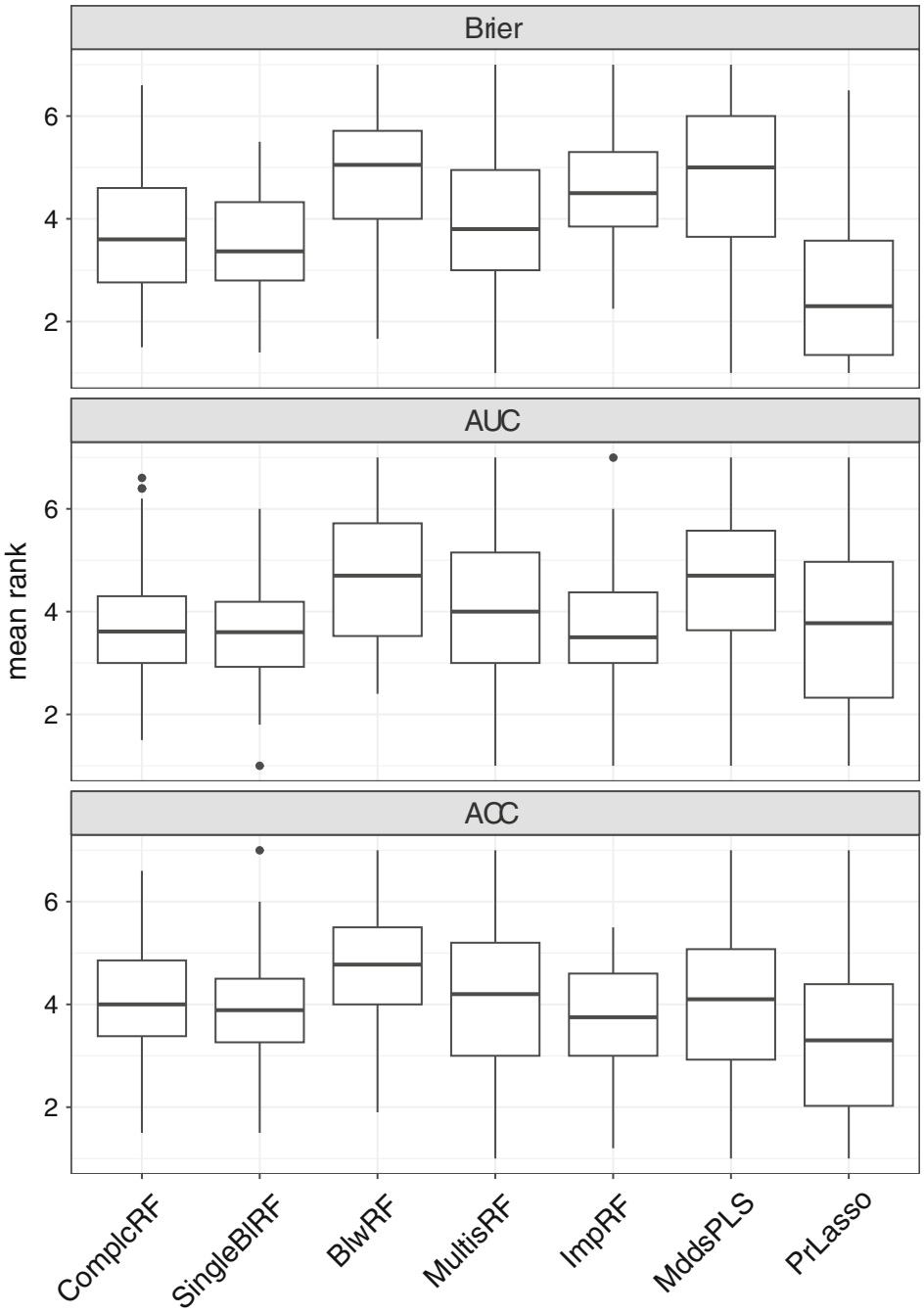

FIGURE 3 Ranks each method achieved among the other methods in terms of the three considered performance metrics—global performance. The ranks were computed for each combination of trbmp, tebmp, data set, and repetition, where only those repetitions were considered for which the results were available for all seven considered methods. The figure shows these ranks averaged across repetitions, with lower ranks indicating better performance.

we would have included all available repetitions for each method, the comparison between the methods would not have been fair. This is because, as stated above, some methods did not deliver predictions for specific trmps and tebmps. Given that the predictive performance generally differed across different trbmps and tebmps, the comparison of the methods would have been confounded by the trmps and tebmps if all available results would have been used.

In general, the differences between the performances observed for the different methods are not very strong. For the Brier score and the accuracy `PrLasso` performed best, while for the AUC there is no clear winner among the methods. `BlwRF` was among the worst-performing methods for all three performance metrics, while for `MddsPLS` this was the case only for the Brier score and the AUC.

As stated above, in Figure 3, we only included those repetitions for which results were available for all seven methods. Because of this, many repetitions were excluded. This was mainly due to `ComplcRF` and `MultisRF`. For example, as stated above, all repetitions with trbmp 1 were excluded because `MultisRF` did not deliver results for this method. Therefore, Supplementary Figures S1 and S2 show the results presented above, however, under the exclusion of `ComplcRF` and `MultisRF`, respectively. Excluding `ComplcRF` and `MultisRF` allowed us to consider more repetitions because we did not need to exclude all repetitions for which results were not available for `ComplcRF` and `MultisRF`, respectively. We do not see any notable differences in the results after the exclusion of `ComplcRF` and `MultisRF`, respectively.

Supplementary Figures S3–S5 show the raw values of the metrics, which confirm that the differences between the results obtained for the different methods are not strong.

3.3.2 | Performance separately by trbmp

Figure 4 shows the ranks each method achieved with respect to the Brier score among the other methods separately by trbmp. For reasons of clarity, in the main paper, we do not present these results for all three performance metrics. The corresponding results obtained for the AUC and the accuracy are shown in Supplementary Figures S6 and S7. The reason why we decided on the Brier score was that it can be seen as the most important metric because it measures both discrimination and calibration, whereas the AUC and the accuracy each measure only one of these. In the following descriptions we will focus on the Brier score, but also describe differences observed for the other two performance metrics.

`PrLasso` performed best for all four trbmps shown in Figure 4. Note that trbmp 1 is not included here because `MultisRF` is not applicable for trbmp 1 and we again only considered repetitions for which results were available for all seven methods. For trbmps 2 to 5, `BlwRF` and `MddsPLS` were again among the worst-performing methods. For trbmps 2, 4, and 5, `ImpRF` was also among the worst-performing methods, but not for trbmp 3. The observation that `ImpRF` was not among the worst methods for trbmp 3 can likely be explained by a feature of `ImpRF`. As described in Section 3.2.1, `ImpRF` uses the concatenation of those blocks that are observed for all observations to calculate the distance matrix that is used in the imputation. For trbmp 3, two blocks are available for all observations, whereas for the other trbmps this is the case for only one block (excluding trbmp 1 with no missing observations). Thus, for trbmp 3 the calculated distance matrix can be expected to better reflect the true distances between the observations, which would explain, why `ImpRF` performed better for trbmp 3 than for the other trbmps. However, we did not see this for the AUC and the accuracy (Supplementary Figures S6 and S7), where `ImpRF` generally performed better compared to the results obtained for the Brier score.

For trbmp 5, excluding `PrLasso` and `SingleBlRF`, all methods performed similarly poorly. `ComplcRF` likely performed worse for trbmp 5 than for the other trmps because the number of complete observations is much smaller for this trbmp. The likely reason `MultisRF` performed worse for trbmp 5 was that the subsets were much smaller in size in comparison to the other trbmps. The predictive performance of the random forests learned on these small subsets probably suffered. Again, `ImpRF` performed better with respect to the AUC and the accuracy for trbmp 5.

The results obtained under the exclusion of `ComplcRF` and `MultisRF` were again very similar (Supplementary Figures S8–S13). The results obtained for trbmp 1 can only be studied by excluding `MultisRF` (Supplementary Figure S11) because this method was not applicable for trbmp 1. These results were similar to those obtained for trbmps 2, 3, and 4 except that `BlwRF` clearly performed worst here. However, for the AUC `MddsPLS` performed similarly bad (Supplementary Figure S12). Note that, if there are no missing blocks in the training data (i.e., for trbmp 1), `ComplcRF` and `ImpRF` are identical.

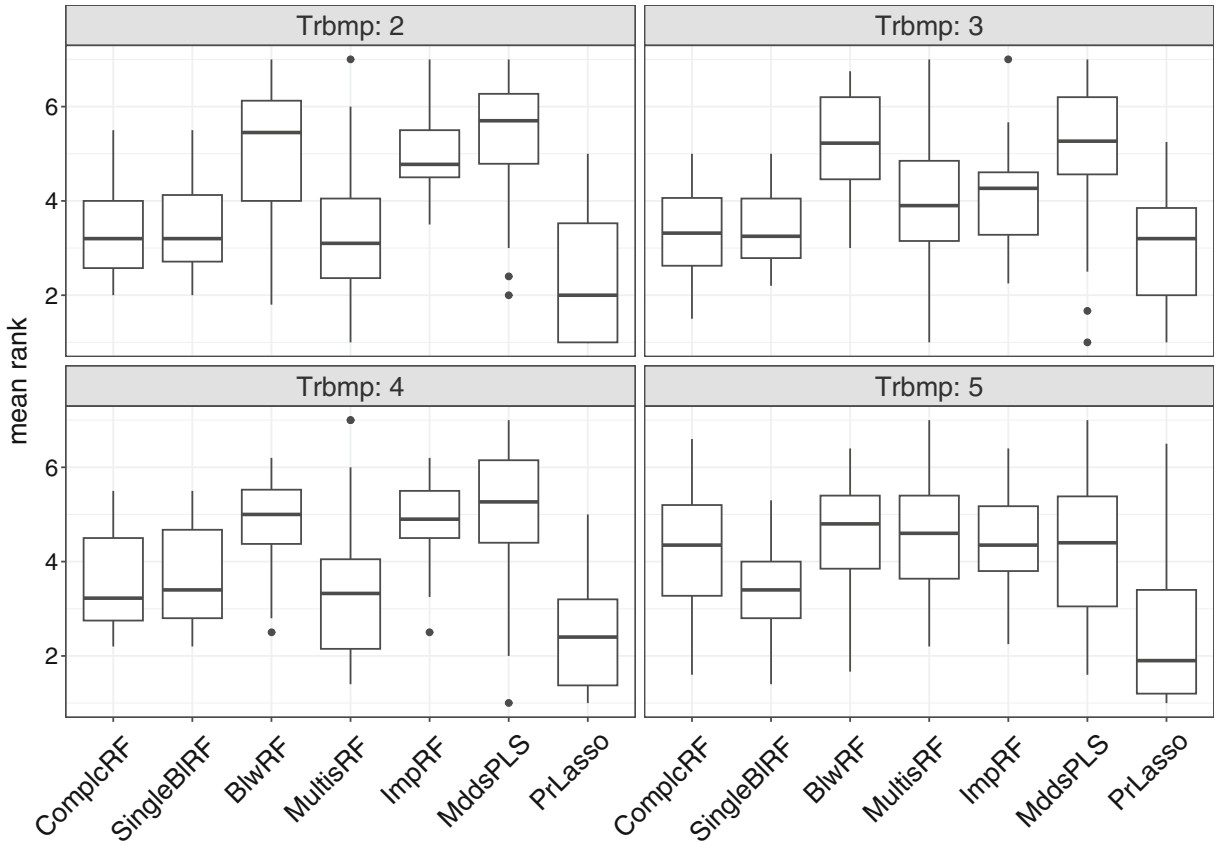

FIGURE 4 Ranks each method achieved among the other methods in terms of the Brier score—separately by trbmp. The ranks were computed for each combination of trbmp, tebmp, data set, and repetition, where only those repetitions were considered for which the results were available for all seven methods. The figure shows these ranks averaged across repetitions, with lower ranks indicating better performance.

In general, for the AUC and the accuracy, the differences in performance between the methods were smaller than for the Brier score (Supplementary Figures S6, S7, S9, S10, S12, and S13). The raw values of the performance metrics obtained for the analysis separated by trbmp are shown in Supplementary Figures S14–S22.

3.3.3 | Performance separately by tebmp

For comparing the results observed for the different methods separately by tebmp we again focus on the Brier score (Figure 5). `PrLasso` again performed the best for all tebmps excluding tebmp 2, where `ComplcRF` and `SingleBlRF` performed similarly well. For `ComplcRF`, this good performance may be explainable by the fact that for tebmp 2 there are few blocks observed in the test data, which is why many blocks are removed from the training data which in turn increases the number of complete observations in the training data. Note, however, that for the AUC and the accuracy (Supplementary Figures S23 and S24), `PrLasso` was not clearly the best method.

For tebmp 1 `ComplcRF`, `SingleBlRF`, `BlwRF`, and `ImpRF` all performed equally well. This can be explained by the fact that for this tebmp only the clinical block is available in the test data and for these methods all blocks not available in the test data are removed from the training data. Therefore, these four methods all function identically for tebmp 1 because they all construct standard random forests using only the clinical block. Note that for tebmp 1, `PrLasso` corresponds to a simple Lasso model fitted to the clinical block. Against this background it is interesting to see that `PrLasso` still performed better than these four random forest-based methods named above (not in the case of the AUC, see Supplementary Figure S23). This means that, when using the clinical block as covariate data, standard Lasso performs better than standard random forests. Tebmp 1 is also the only setting for which `MddsPLS` worked better than these four random forest-based methods named above. Given that tebmp 1 is the only tebmp for which all blocks

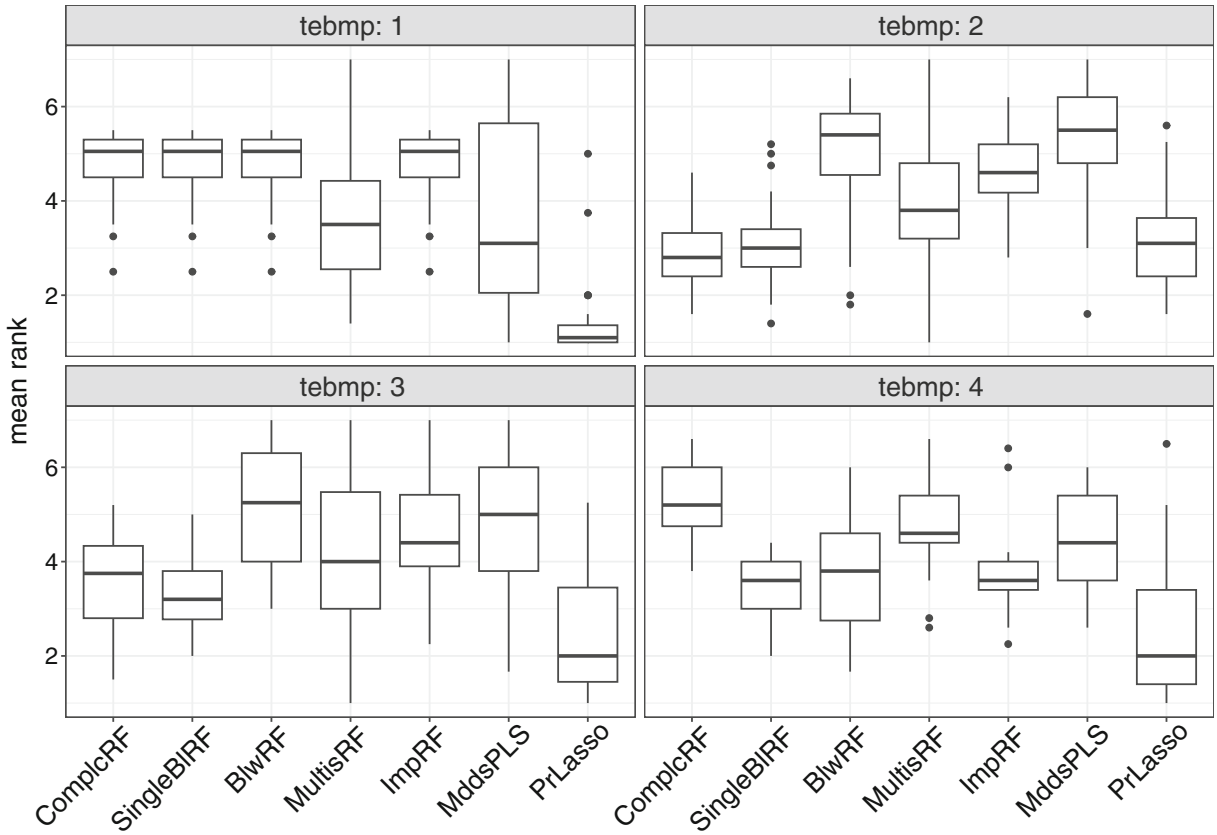

FIGURE 5 Ranks each method achieved among the other methods in terms of the Brier score—separately by tebmp. The ranks were computed for each combination of trbmp, tebmp, data set, and repetition, where only those repetitions were considered for which the results were available for all seven methods. The figure shows these ranks averaged across repetitions, with lower ranks indicating better performance.

except the clinical block are missing in the test data, the better performance of `MddsPLS` for tebmp 1 might be due to the fact that this method is the only one that imputes the missing blocks in the test data. For the AUC, however, `MddsPLS` performed worse than the four random forest-based methods named above also for tebmp 1; for this metric, `MultisRF` also performed worse.

When interpreting the results for the remaining tebmps, it must be considered that, in Figure 5, the results displayed for tebmp 4 are only those obtained for trbmp 5. In this figure, as before, we show only the results of those repetitions for which each of the seven considered methods delivered a result. As explained above, for tebmp 4, the results of either `ComplcRF` (for trbmps 2, 3, and 4) or `MultisRF` (for trbmp 1) were missing for all trbmps excluding trbmp 5. Therefore, when interpreting the results obtained for tebmp 4 we must resort to Supplementary Figure S25, which shows the corresponding results obtained under the exclusion of `ComplcRF`. `BlwRF` and `MddsPLS` are again among the worst-performing methods for tebmps 2, 3, and 4. For tebmps 3 and 4 (cf. Supplementary Figure S25 for tebmp 4) `ImpRF` also is among the worst-performing methods, but this is seen less clearly when studying the results for each combination of trbmp and tebmp (Section 3.3.4) and not at all for the AUC and the accuracy (Supplementary Figures S23, S24, S26, S27, S29, and S30). An explanation for why we observed this slightly worse performance of `ImpRF` for tebmps 3 and 4 in the case of the Brier score could be the following: for larger numbers of blocks in the test data more blocks and thus more missing values are retained in the training data meaning that larger proportions of missing values need to be imputed.

The differences between the methods in performance were, as in the previous sections, smaller for the AUC and the accuracy (Supplementary Figures S23, S24, S26, S27, S29, and S30). Excluding `ComplcRF` and `MultisRF` (Supplementary Figures S25–S30) did again not change the results strongly, except for tebmp 4, as discussed above. For the raw values of the performance metrics, see Supplementary Figures S31–S39.

3.3.4 | Performance separately for each combination of trbmp and tebmp

In the previous two subsections we studied the performances of the methods separately by trbmp and by tebmp, but not separately by the various combinations of trbmps and tebmps. Figure 6 shows the ranks each method achieved among the other methods with respect to the Brier score separately for the different combinations of trbmps and tebmps. Because some methods were not applicable for certain (combinations of) trbmps and tebmps, to interpret the results obtained for all combinations of trbmps and tebmps, we also have to consult Supplementary Figures S42 and S45, which show these results under the exclusion of `ComplcRF` and `MultisRF`, respectively.

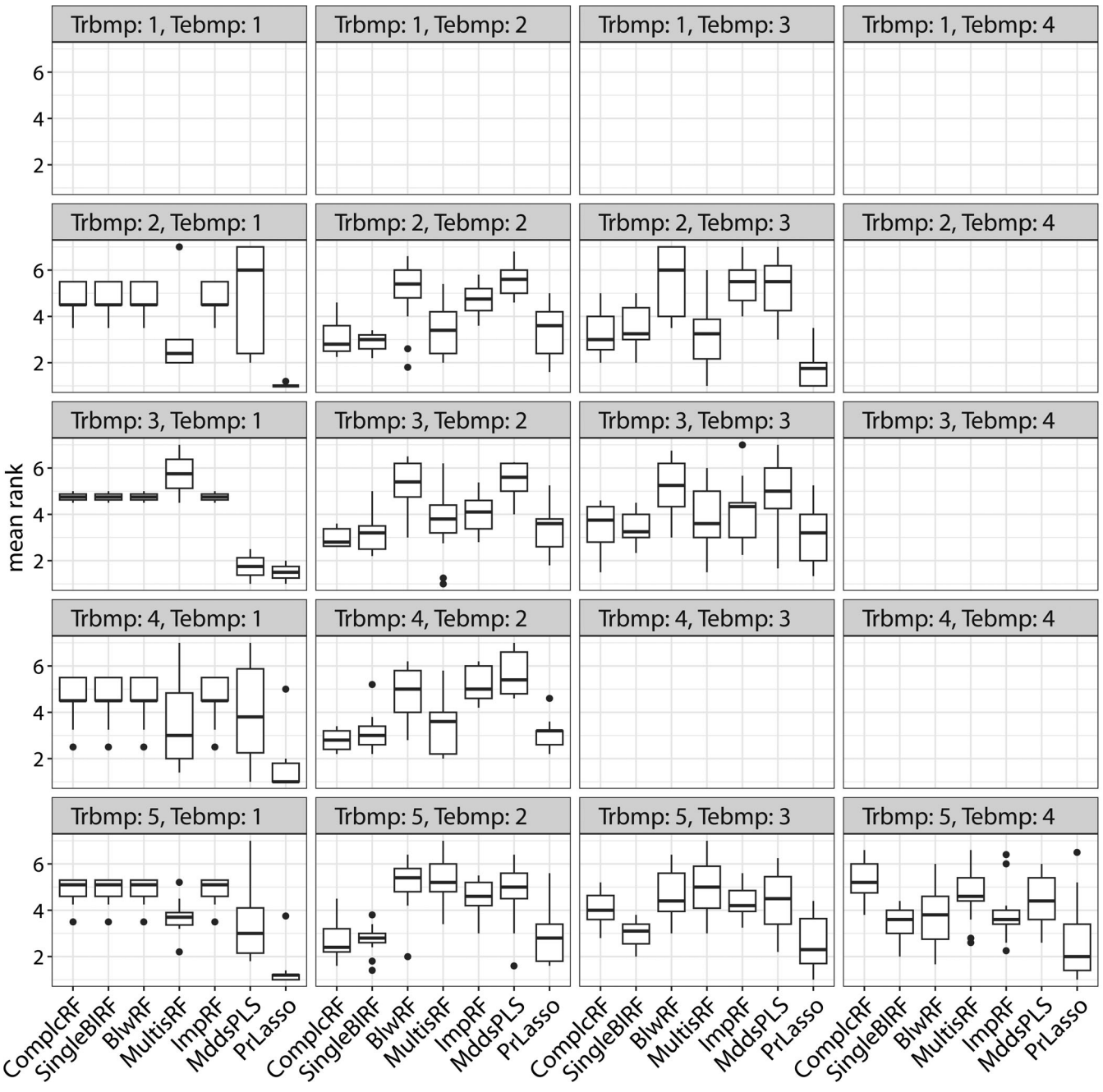

FIGURE 6 Ranks each method achieved among the other methods in terms of the Brier score—separately by each combination of trbmp and tebmp. The ranks were computed for each combination of tebmp, tebmp, data set, and repetition, where only those repetitions were considered for which the results were available for all seven methods. The figure shows these ranks averaged across repetitions, with lower ranks indicating better performance.

One observation that can be made across all three performance metrics considered is that for tebmp 4, that is, the setting with no missing observations in the test data, `ComplcRF` performed much worse for trbmp 5 than for trbmp 1 (Supplementary Figures S45–S47). This can most probably be explained by the fact that for trbmp 1 all observations are complete, whereas for trbmp 5 only the observations in the first subset are complete. This is why for trbmp 1 there are many more observations available for training in the case of `ComplcRF`. In Section 3.3.2, we already made the observation that `ComplcRF` performed worse for trbmp 5, but here, when focusing on tebmp 4 this worsening is considerably stronger because if all blocks are available in test data (i.e., for tebmp 4), the complete observations are always restricted to the first subset in trbmp 5.

Another observation which can be made for the Brier score (Figure 6) and the accuracy (Supplementary Figure S41), however, not for the AUC (Supplementary Figure S40), is that in the case of tebmp 1, `MultisRF` performed worse for trbmp 3 than for trbmps 2, 4, 5. This may be explained by the fact that, starting with the first splits, `MultisRF` prunes all trees, cutting tree branches at splits for which variables are used that do not occur in the test set. Because the clinical covariates are so few in comparison to the omics covariates it is very unlikely that the first splits use clinical covariates, which is why, if the test data only feature the clinical covariates (i.e., for tebmp 1), many trees are removed entirely from the forests, which naturally leads to a worsening of the predictive performance. Finally, the reason why `MultisRF` performed worse for trbmp 3 than for the other trbmps (2, 4, and 5) could be that for trbmp 3 more blocks are observed, which could make it particularly unlikely that the first splits use clinical covariates.

For the combination trmbp 1 and tebmp 4 `PrLasso` did not clearly outperform `ComplcRF` (Supplementary Figures S45–S47). This is interesting against the background that in Section 3.3.3, it was seen that `PrLasso` outperformed `ComplcRF` (as well as `SingleBlRF`, `BlwRF`, and `ImpRF`) in the case of tebmp 1, for which only the clinical block is used as covariate data. This indicates that `PrLasso` may only outperform `ComplcRF` (i.e., standard random forests) if only clinical covariate data are used, but not necessarily for multi-omics data.

The results obtained for the AUC and the accuracy were mostly very similar to those obtained for the Brier score; however, the differences between the methods' performance tended to again be smaller for the former two performance metrics (Supplementary Figures S40, S41, S43, S44, S46, and S47). With a few exceptions, excluding `ComplcRF` and `MultisRF` hardly changed the results obtained for the remaining methods (Supplementary Figures S42–S47).

The raw values of the performance metrics are shown in Supplementary Figures S48–S56. Note that here, in each sub-figure, we have added the performance metric values obtained using random forests constructed and evaluated on the data without missing values (equivalent to the combination trbmp 1 and tebmp 4). For these random forests, hereafter referred to as `FulldataRF`, we used the same configuration as for the other approaches using random forests. We have included the results of `FulldataRF` to assess the degree to which the missing values lead to a deterioration in the predictive performance. If `FulldataRF` would perform much better than the best methods evaluated using the data with block-wise missingness, this would suggest that none of the compared methods are able to deal with block-wise missingness efficiently. In this case, addressing the issue of missing data in the field of multi-omics data would be more crucial than attempting to deal with it post hoc using sophisticated (prediction) methods.

Supplementary Figures S48–S56 reveal that only for tebmp 1 does `FulldataRF` demonstrate a markedly better performance than methods using the data with block-wise missingness. Note that tebmp 1 is the setting where only the clinical covariates are available in the test data. Therefore, this finding only means that the performance is improved when the omics blocks are also considered. For tebmp 2, which includes the clinical covariates and an omics block in the test data, FulldataRF also performed best; however, the performance difference compared to the methods using the data with block-wise missingness is much smaller. For tebmps 3 and 4, FulldataRF performed similarly to the other methods. There are no distinct differences between the trbmps regarding the performance gap between FulldataRF and the methods that use the data with block-wise missingness.

In summary, the results suggest that the methods seem to effectively exploit the predictive information present in data with block-wise missingness. However, the test data should contain more than just clinical covariates to exploit the predictive information in the omics blocks.

4 | DISCUSSION

For the great majority of the different trbmps and tebmps we considered, `PrLasso` performed best with respect to the Brier score. In Section 3.3.3, we obtained strong evidence that `PrLasso` better exploited the predictive information contained in the clinical covariates than the random forest-based methods. This could be an important reason for the

superior performance of PrLasso in our comparison study, given that the clinical covariates have been found to be very important to prediction with multi-omics data (Herrmann et al., 2020).

BlwRF and MddsPLS performed worst for most settings. These two methods have in common that they treat the blocks independently from each other, where for prediction all blocks are used. Treating the different blocks separately might not be effective because the predictive information contained in them is overlapping. Surprisingly, the naive methods were not among the worst methods in general and even among the best for some settings. In contrast, PrLasso demonstrated a more robust behavior in the sense that it was consistently among the best methods. As expected, the complete case approach ComplcRF did not perform well if there were only few complete observations in the training data.

In its current form MultisRF is not very well suited for the multi-omics case. An important reason for this is the pruning procedure performed prior to prediction (cf. Section 3.2.1). If the test data contain blocks of small size, this procedure is expected to lead to the removal of large proportions of the trees in the random forests constructed with MultisRF. We saw this in our comparison study, where, as discussed in Section 3.3.4, the predictive performance was diminished if the test data only contained clinical covariates, but at the same time in the test data many omics blocks were available. MultisRF could also be better adjusted to the multi-omics case by replacing the standard random forests constructed using the different subsets by a prediction method that takes the multi-omics structure into account, for example, by the block forests method (Hornung & Wright, 2019).

The results differed across the different performance metrics. For example, while PrLasso performed substantially better than the other methods with respect to the Brier score, there was no clear winner among the methods for the AUC. In general, great care has to be taken in the interpretation of our results. As all studies based on real data sets, ours may have yielded different results to some extent if we had considered other real data sets as a basis (other inclusion criteria, or data from other databases) and more random repetitions for each of these real data sets. Moreover, while the number of data sets is larger than in many benchmark studies that are based on few, say four or five, data sets the number of included data sets is still limited. Nießl et al. (2022) have shown that the results of benchmark studies are in general variable and strongly affected by analytic choices even if large numbers of data sets are used. Against this background, to avoid drawing non-replicable conclusions, we took great care to offer reasonable explanations for our observations. In the cases of ComplcRF and MultisRF there were no results for certain trmpbs and tempbs. In general, systematically missing values of this kind are an issue in empirical benchmark studies, which may lead to biased results. However, by only including repetitions obtained for all seven methods in our visualizations we took care to present our results appropriately in view of this issue. Moreover, the additional figures presented in each case that excluded ComplcRF and MultisRF, respectively, suggested that our results are quite robust against biases caused by the missing values.

Our study has a number of further limitations that may be addressed in future research. For reasons of clarity, we investigated only a limited number of simplified missing patterns, which may not cover all potential scenarios encountered in reality. Practical scenarios may for example feature more "irregular patterns" with subsets of patients of very different sizes. Moreover, there could be missing values in individual covariates, that is, missing values beyond whole types of omics data missing for subsets of patients.

Importantly, in practice the data in the subsets that feature different block combinations often stem from different sources (e.g., generated at different time points or using different machines). This leads to systematic differences between the data subsets that are generally known as batch effects (Li et al., 2009). The training data and test data subsets in our empirical study were random partitions of the same data sets and thus did not feature batch effects. In the training data such batch effects can be corrected using batch effect removal methods such as ComBat (Johnson et al., 2006). For correcting the test data correspondingly, add-on batch effect removal can be used (Hornung et al., 2016), where the test data is transformed to be similar to the training data in distribution. This add-on batch effect removal helps to improve the predictive performance which tends to deteriorate in the presence of batch effects (Hornung et al., 2017).

We assumed that the model can be retrained depending on the respective test data missingness pattern before prediction. However, this may not always be feasible in practice. Applying models to test data with strongly varying missingness patterns is fundamentally different from the setting considered in the present study and requires different, possibly more complex techniques.

While we claim that the choice of the TP53 mutation as a binary response variable considered as surrogate for disease outcome is acceptable from the purely methodological perspective adopted here, it may make sense to consider other response variables or to extend the study to other types of response variables such as censored survival times. However, not all methods considered in our study are capable to handle this case. Our study could also be extended to include further methods, in particular methods not implemented in R, which we excluded from our comparison.

Finally, in our empirical study, we assumed that the data types are missing completely at random (MCAR) (Little & Rubin, 2019), meaning that their missingness does not depend on their own values or the values of other data types. It is reasonable to assume an MCAR mechanism for multi-omics data, as the reasons for missingness are typically technical rather than biological. Hieke et al. (2016) cite assay failure, sample quality issues, and changes in measurement platforms as reasons why not all data types are available for every patient in multi-omics data. However, since the data come from different sources, there may still be biological differences between subsets. In such cases, the observed and missing data could differ systematically between different subsets. In these situations, the differences between missing data in different subsets would be explainable by other data types, particularly the clinical block, which would make the data missing at random (MAR) (Little & Rubin, 2019). With the exception of the complete case approach, the compared methods include all observations and, therefore, are not biased in the sense that they would exclude certain observations based on their data distributions. Thus, we expect that we would have obtained similar results under the MAR assumption.

As seen in Section 2, most methods unfortunately lack public implementations, making it challenging for interested readers to apply them. From our experience, this is a prevalent issue in the methodological literature. Given that authors need to implement their methods for evaluation in the papers introducing them, making these implementations publicly accessible, for example, through a GitHub repository, should not require significant effort. We hope that this approach becomes more widespread in the future, as it would significantly enhance the applicability of new methods and the execution of neutral comparison studies like the one conducted in this article. Neutral comparison studies are essential, considering the tendency for new methods to perform better in the analyses presented in their introductory papers than in subsequent comparison studies (Buchka et al., 2021).

5 | CONCLUSION

We first provided a state-of-the-art literature overview on prediction methods for block-wise missing multi-omics covariate data. Subsequently, we presented a large-scale benchmark comparison study of some of these methods. The results of this study may aid applied researchers confronted with block-wise missing multi-omics data to select suitable methods. Nevertheless, given the generally high variability of the findings of benchmark studies, it is important to not over-interpret details of the results of our study. In addition to applied researchers, the literature overview and the benchmark study may also aid methodological researchers in developing new, stronger methods that share the strengths of the most promising methods, while addressing their weaknesses.

AUTHOR CONTRIBUTIONS

Roman Hornung: Conceptualization (lead); formal analysis (supporting); funding acquisition (equal); methodology (supporting); project administration (equal); writing – original draft (lead). **Frederik Ludwigs:** Conceptualization (supporting); formal analysis (lead); methodology (supporting); writing – review and editing (supporting). **Jonas Hagenberg:** Formal analysis (supporting); methodology (lead); writing – review and editing (equal). **Anne-Laure Boulesteix:** Funding acquisition (equal); methodology (supporting); project administration (equal); writing – review and editing (equal).

ACKNOWLEDGMENTS

The authors thank Anna Jacob for valuable language corrections and Fei Xue for advice regarding the multiple block-wise imputation approach. Open Access funding enabled and organized by Projekt DEAL.

FUNDING INFORMATION

This work was supported by the German Science Foundation (grant number HO6422/1-2 to Roman Hornung; grant numbers BO3139/6-2 and BO3139/4-3 to Anne-Laure Boulesteix).

CONFLICT OF INTEREST STATEMENT

We have developed some of the methods compared in the empirical comparison study. The first and second authors have developed the multi-source random forest approach and the third author has developed the priority-LASSO-impute approach. However, we were committed to providing a fair comparison, that is, we neither spent more efforts to optimize these two methods than the other methods nor did we design the study to favor one or the other method.

OPEN RESEARCH BADGES

This article has earned an Open Data badge for making publicly available the digitally-shareable data necessary to reproduce the reported results. The data is available at https://doi.org/10.6084/m9.figshare.22304050.v2 and https://github.com/RomanHornung/bwm_article.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in figshare at https://doi.org/10.6084/m9.figshare.22304050.v2. All R code written to perform and evaluate the analyses, including data pre-processing, is available on GitHub (https://github.com/RomanHornung/bwm_article).

ORCID

Roman Hornung https://orcid.org/0000-0002-6036-1495
Jonas Hagenberg https://orcid.org/0000-0002-1849-1106
Anne-Laure Boulesteix https://orcid.org/0000-0002-2729-0947

RELATED WIREs ARTICLES

Integrative clustering methods for multi-omics data

REFERENCES

Beretta, L., & Santaniello, A. (2016). Nearest neighbor imputation algorithms: A critical evaluation. *BMC Medical Informatics and Decision Making*, *16*(3), 74.

Boulesteix, A.-L., Binder, H., Abrahamowicz, B., Sauerbrei, W., & Simulation Panel of the STRATOS Initiative. (2017). On the necessity and design of studies comparing statistical methods. *Biometrical Journal. Biometrische Zeitschrift*, *60*(1), 216–218.

Boulesteix, A.-L., Wilson, R., & Hapfelmeier, A. (2017). Towards evidence-based computational statistics: Lessons from clinical research on the role and design of real-data benchmark studies. *BMC Medical Research Methodology*, *17*(1), 1–12.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Buchka, S., Hapfelmeier, A., Gardner, P. P., Wilson, R., & Boulesteix, A.-L. (2021). On the optimistic performance evaluation of newly introduced bioinformatic methods. *Genome Biology*, *22*, 152.

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

Chen, J., & Zhang, A. (2020). Hgmf: Heterogeneous graph-based fusion for multimodal data with incompleteness. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 1295–1305). Association for Computing Machinery.

Dai, Z., Bu, Z., & Long, Q. (2021). Multiple imputation via generative adversarial network for high-dimensional blockwise missing value problems. In *20th IEEE international conference on machine learning and applications (ICMLA)* (pp. 791–798). Institute of Electrical and Electronics Engineers (IEEE).

Dong, A., Li, Z., Wang, M., Shen, D., & Liu, M. (2021). High-order Laplacian regularized low-rank representation for multimodal dementia diagnosis. *Frontiers in Neuroscience*, *15*, 634124.

Dong, X., Lin, L., Zhang, R., Zhao, Y., Christiani, D. C., Wei, Y., & Chen, F. (2019). Tobmi: Trans-omics block missing data imputation using a k-nearest neighbor weighted approach. *Bioinformatics*, *35*(8), 1278–1283.

Herrmann, M., Probst, P., Hornung, R., Jurinovic, V., & Boulesteix, A.-L. (2020). Large-scale benchmark study of survival prediction methods using multi-omics data [bbaa167]. *Briefings in Bioinformatics*, *22*, bbaa167. https://doi.org/10.1093/bib/bbaa167

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

Hornung, R., Boulesteix, A.-L., & Causeur, D. (2016). Combining location-and-scale batch effect adjustment with data cleaning by latent factor adjustment. *BMC Bioinformatics*, *17*, 27.

Hornung, R., Causeur, D., Bernau, C., & Boulesteix, A.-L. (2017). Improving cross-study prediction through addon batch effect adjustment or addon normalization. *Bioinformatics*, *33*(3), 397–404.

Hornung, R., & Wright, M. N. (2019). Block forests: Random forests for blocks of clinical and omics covariate data. *BMC Bioinformatics*, *20*, 358.

Ingalhalikar, M., Parker, W. A., Bloy, L., Roberts, T. P., & Verma, R. (2012). Using multiparametric data with missing features for learning patterns of pathology. In *International conference on medical image computing and computer-assisted intervention* (pp. 468–475). Springer.

Johnson, W. E., Li, C., & Rabinovic, A. (2006). Adjusting batch effects in microarray expression data using empirical bayes methods. *Biostatistics*, *8*(1), 118–127.

Klau, S., Hornung, R., Bauer, A., & Hagenberg, J. (2023). *Prioritylasso: Analyzing multiple omics data with an offset approach* [R package version 0.3.1]. https://CRAN.R-project.org/package=prioritylasso

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

Krautenbacher, N., Flach, N., Böck, A., Laubhahn, K., Laimighofer, M., Theis, F. J., Ankerst, D. P., Fuchs, C., & Schaub, B. (2019). A strategy for high-dimensional multivariable analysis classifies childhood asthma phenotypes from genetic, immunological, and environmental factors. *Allergy*, *74*(7), 1364–1373.

Lan, Q., & Jiang, S. (2021). A method of credit evaluation modeling based on block-wise missing data. *Applied Intelligence*, *51*(10), 6859–6880.

Lan, W., Chen, X., Zou, T., & Tsai, C.-L. (2022). Imputations for high missing rate data in covariates via semisupervised learning approach. *Journal of Business & Economic Statistics*, *40*(3), 1282–1290.

Li, J., Bushel, P., Chu, T.-M., & Wolfinger, R. D. (2009). Principal variance components analysis: Estimating batch effects in microarray gene expression data. In A. Scherer (Ed.), *Batch effects and noise in microarray experiments: Sources and solutions* (pp. 141–154). John Wiley & Sons.

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

Little, R. J. A., & Rubin, D. B. (2019). *Statistical analysis with missing data* (3rd ed.). Wiley.

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

Lorenzo, H., Razzaq, M., Odeberg, J., Morange, P.-E., Saracco, J., Trégouët, D.-A., & Thiébaut, R. (2019). High-dimensional multi-block analysis of factors associated with thrombin generation potential. In *IEEE 32nd international symposium on computer-based medical systems (CBMS)* (pp. 453–458). Institute of Electrical and Electronics Engineers (IEEE).

Lorenzo, H., Saracco, J., & Thiébaut, R. (2019). *Supervised learning for multi-block incomplete data* [arXiv:1901.04380].

Ludwigs, F. (2020). *A comparison study of prediction approaches for multiple training data sets and test data with block-wise missing values* [Master's thesis]. University of Munich.

Nießl, C., Herrmann, M., Wiedemann, C., Casalicchio, G., & Boulesteix, A.-L. (2022). Over-optimism in benchmark studies and the multiplicity of design and analysis options when interpreting their results. *WIREs Data Mining and Knowledge Discovery*, *12*, e1441.

Stekhoven, D. J., & Bühlmann, P. (2011). Missforest—Non-parametric missing value imputation for mixed-type data. *Bioinformatics*, *28*(1), 112–118.

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

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

Wang, X., & Sun, Q. (2017). Tp53 mutations, expression and interaction networks in human cancers. *Oncotarget*, *8*(1), 624–643.

Weinstein, J. N., Collisson, E. A., Mills, G. B., Shaw, K. R., & Ozenberger, B. A. (2013). The cancer genome atlas pan-cancer analysis project. *Nature Genetics*, *45*(10), 1113–1120.

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

Xue, F., & Qu, A. (2021). Integrating multisource block-wise missing data in model selection. *Journal of the American Statistical Association*, *116*(536), 1914–1927.

Yang, X., Kim, Y.-J., Taub, M., Azevedo, R., & Chi, M. (2020). Prime: Block-wise missingness handling for multimodalities in intelligent tutoring systems. In *International conference on multimedia modeling* (pp. 63–75). Springer International Publishing.

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

Zhang, Y., Tang, N., & Qu, A. (2020). Imputed factor regression for high-dimensional block-wise missing data. *Statistica Sinica*, *30*(2), 631–651.

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

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

How to cite this article: Hornung, R., Ludwigs, F., Hagenberg, J., & Boulesteix, A.-L. (2024). Prediction approaches for partly missing multi-omics covariate data: A literature review and an empirical comparison study. *WIREs Computational Statistics*, *16*(1), e1626. https://doi.org/10.1002/wics.1626