LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

INSTITUT FÜR STATISTIK

Roman Hornung, David Causeur, Christoph Bernau, Anne-Laure Boulesteix

Improving cross-study prediction through addon batch effect adjustment and addon normalization

Technical Report Number 194, 2016
Department of Statistics
University of Munich

http://www.stat.uni-muenchen.de

Improving cross-study prediction through addon batch effect adjustment and addon normalization

Roman Hornung[1*] David Causeur[2] Christoph Bernau[3]
Anne-Laure Boulesteix[1]

June 8, 2016

[1] Department of Medical Informatics, Biometry and Epidemiology, University of Munich, Munich, 81377, Germany

[2] Applied Mathematics Department, Agrocampus Ouest, Rennes, 35042, France

[3] Leibniz Supercomputing Center, Garching, 85748, Germany

Abstract

To date most medical tests derived by applying classification methods to high-dimensional molecular data are hardly used in clinical practice. This is partly because the prediction error resulting when applying them to external data is usually much higher than internal error as evaluated through within-study validation procedures. We suggest the use of addon normalization and addon batch effect removal techniques in this context to reduce systematic differences between external data and the original dataset with the aim to improve prediction performance. We evaluate the impact of addon normalization and seven batch effect removal methods on cross-study prediction performance for several common classifiers using a large collection of microarray gene expression datasets, showing that some of these techniques reduce prediction error. All investigated addon methods are implemented in our R-package `bapred`.

1 Introduction

A large variety of modern classification methods can be used to construct tests on the presence of diseases or disease outcomes of interest on the basis of high-dimensional, molecular data. Such tests, denoted prediction rules in the following, could potentially be established as useful tools to assist medical doctors in their decision finding (van't Veer and Bernards, 2008). Nevertheless, to date they are hardly applied in daily medical practice. Apart

*Corresponding author. Email: hornung@ibe.med.uni-muenchen.de.

from policy reasons, a major stumbling block counteracting a broader application is lacking comparability of the data from patients to predict, from now on denoted as "test data", to that the prediction rules are obtained on, in the following denoted as "training data". This leads to a higher prediction error when applying prediction rules to independent external data in practice than dataset internal validation through cross-validation suggests. High-dimensional bio-molecular measurements are highly sensitive to external conditions of the data generation procedure (Scheerer, 2009). Moreover, different datasets studying the same biological phenomenon also vary depending on the study population. For these reasons, the performance of prediction rules can be expected to be worse or even considerably worse in practice than the results of dataset-internal error estimation suggest (Castaldi *et al.*, 2011; Bernau *et al.*, 2014). From now on, we use the term "cross-study" to refer to situations where a prediction rule is learned using data from a study and applied to independent external data from another study.

It is a desirable goal to reduce the error frequency resulting when applying a constructed prediction rule in cross-study settings. There are various batch adjustment methods which are frequently used to make the distributions of different datasets more similar not only within a study but also across studies. However, it is far less acknowledged that these methods can also be applied to make test data more similar to the training data in the context of prediction. Some of these methods have to be adjusted slightly before being applicable in the context of prediction. The reason for this is that the training data must not change when adjusting the test data. This in turn ensures that the prediction rule remains fixed when new test data arrives. We speak of "addon batch effect adjustment" when batch effect adjustment is performed in this way. See Hornung *et al.* (2016), who discuss addon batch effect adjustment in detail.

Independently from addon batch effect adjustment, by normalizing the training and test data simultaneously, the severity of batch effects would already be strongly reduced. However, as noted above, in the context of prediction the prediction rule is required not to depend on the test data. This condition would however not be fulfilled when normalizing training and test set together, because the training data would change each time new test data arrives. This pitfall can be addressed by the so-called "addon normalization": the normalization of the training data is done without considering the test data. When normalizing the observations in the test data, for those parameters of the normalization procedure which do not entirely depend on the individual samples, estimates obtained from the training data only are used. Such an addon normalization procedure exists for Robust Multi-array Average (RMA) normalization, see Kostka and Spang (2008).

In this paper we study the potential improvement of cross-study prediction yielded by the use of addon normalization, addon batch effect ad-

justment and the combination of these two through application to 34 raw microarray datasets of the same chiptype. Our study represents a large-scale neutral comparison study following the recommendations of Boulesteix *et al.* (2013) and Boulesteix (2013). Beyond small illustrative (and often biased) real data studies provided in the great majority of papers presenting new methods, such neutral comparison studies yield crucial evidence to guide data analysis practice (Boulesteix, 2013; Gatto *et al.*, 2016). The high number of datasets considered in these studies considerably increases the reliability of the conclusions (Boulesteix *et al.*, 2015). We consider seven batch effect adjustment methods and the addon normalization procedure for RMA by Kostka and Spang (2008). The target variable considered for all datasets is "sex". Cross-validation delivers error rates close to zero here, because the biological signal present in gene expression for explaining "sex" is very strong. However, the error rate estimated by cross-study validation will be seen to be much higher, although from a biological point it should also be possible to perfectly predict sex based on microarray gene expression data in cross-study settings. This illustrates that batch effects can considerably deteriorate prediction accuracy in this context and that cross-validation does not reflect the true error rate to be expected when applying a prediction rule to an external dataset in practice.

Note that it is of course not meaningful to predict "sex" from a clinical point of view. However, for the purpose of our systematic large-scale study it is important to analyze a high number of datasets with the same phenotype target variable and collected using the same chiptype, which was possible only for the target variable "sex". Omitting the fact that the biological signal present in gene data for explaining "sex" is very strong, "sex" can be seen as a substitute for a meaningful phenotype target variable. Moreover, "sex" has the advantage of being a clearly defined target variable. By contrast, for clinically relevant target variables it is often difficult to find several datasets which do feature the same two biological groups and definitions may be ambiguous. Keeping in mind that prediction performance is usually better for "sex" than for most other target variables, in our study we will not examine the absolute sizes of the performance measure values but deliberately focus on the effect of addon batch effect adjustment and addon normalization.

Modern next generation sequencing (NGS) data is commonly associated with a strongly reduced variability in comparison to microarray data (Bullard *et al.*, 2010), wherefore here batch effects should be weaker. Nevertheless also for NGS data, batch effects have been found to pose a problem (Hansen and Irizarry, 2012). The question investigated in our study is thus relevant beyond the special case of microarray data.

The study by Luo *et al.* (2010) on addon batch effect adjustment investigates a question related to the question considered in our paper. The crucial difference is that Luo et al. do not consider cross-study prediction but cross-

batch prediction within the same study. In their paper, batches are different parts of a common dataset which are incomparable for reasons unrelated to the biological signal of interest. Since their batches originate from the same study, these share certain common characteristics. For example, the laboratory used for the data generation or the personnel involved may be the same for all batches. Such similarities between training and test data are however not present in general in cross-study settings when a prediction rule is made publicly available and applied by other teams throughout the world. Therefore, our analysis design better reflects practically relevant situations. Moreover, by considering a large number of datasets we obtain more stable results.

The paper is structured as follows: In the Methods Section, after a description of the data material we detail the analyses performed in the cases of cross-study-prediction using batch effect adjustment and addon-normalization. The following Results Section describes important features of our results. In the Discussion we interpret several of our findings and propose further possibilities for application of the methodology. The Conclusions Section summarizes the main messages of the paper.

2 Methods

2.1 Data material

All datasets were obtained from ArrayExpress (Kolesnikov *et al.*, 2015). As a first step we searched for datasets which met the following criteria: availability of a variable denoted as "sex" in the phenotypic data, availability of the raw data (necessary for (addon) normalization), number of samples between 30 to 500, human origin of the samples, samples of microarray chip type HG-U133PLUS2. From the corresponding search results we initially considered the 39 most recently published datasets, which actually met these criteria. Subsequently we investigated for each dataset whether there were repeated measurements and if so, randomly chose one sample per patient. Following this, we excluded any datasets which contained duplicates from other datasets. Lastly, we excluded those datasets which featured less than 20 observations after removal of repeated measurements resulting in 34 datasets used in the analysis. Supplementary Table S1 provides basic informations on these datasets after removal of repeated measurements.

2.2 (Addon) Batch effect adjustment

The seven considered batch effect adjustment methods are: ComBat (Johnson *et al.*, 2007), frozen SVA (fSVA) (Parker *et al.*, 2014), mean-centering, standardization, ratio-A, ratio-G (Luo *et al.*, 2010) and FAbatch (Hornung *et al.*, 2016). For ComBat we use the addon method presented by Luo *et al.*

(2010). Frozen SVA is an addon method for the batch effect adjustment method SVA (Leek and Storey, 2007). We consider both variants of this method presented in Parker *et al.* (2014): the "exact fSVA algorithm" and the "fast fSVA algorithm". For FAbatch we use the addon method presented in Hornung *et al.* (2016). All remaining methods do not have to be altered for addon batch effect adjustment, because these are performed batch-by-batch, in our case dataset-by-dataset, respectively. For a detailed discussion of addon batch effect adjustment, see Hornung *et al.* (2016).

2.3 (Addon) quantile normalization

RMA normalization (Irizarry *et al.*, 2003) will be used in our analysis. The latter consists of three steps: 1) background correction; 2) quantile normalization (Bolstad *et al.*, 2003); 3) summarization. Background correction and summarization are performed on an array-by-array basis, wherefore no addon strategies are necessary for these procedures. The quantile normalization step is conceptionally performed as follows. Be x_{sort,i^*j} the j smallest variable value of array i^*. Then for each array $i \in \{1,\ldots,n\}$ the j smallest value is determined and the average $\bar{x}_{\text{sort},j}$ over these n values taken. Finally x_{sort,i^*j} is replaced by $\bar{x}_{\text{sort},j}$. By performing this procedure for all variable values, the empirical distributions of all arrays become equal. When normalizing the test observations using addon quantile normalization (Kostka and Spang, 2008) the averages over the j smallest values are obtained using the training data only, i.e. excluding the corresponding test observations. As a consequence the scale of the normalized test observations is consistent with that of the normalized training observations without the latter having being changed in the procedure.

2.4 Cross-study validation

Bernau *et al.* (2014) suggest "cross-study validation" for obtaining estimates of the error expected when applying prediction rules to external data. This procedure requires a collection of I datasets studying the same biological phenomenon. The prediction rule of interest is learned once using each of the I datasets and its error evaluated in turn on every other dataset. This results in $I(I-1)$ error estimates which are more realistic than cross-validation error estimates as far as the application of prediction to external data in practice is concerned.

We altered this procedure slightly to fit our purposes. Instead of an error estimator we consider a performance metric, namely the Matthews Correlation Coefficient (MCC). The absolute size of the latter is interpretable analogously to that of the well-known Bravais-Pearson correlation coefficient used in the case of metric data. For this reason, we favoured it over

the more common misclassification error rate. The MCC is calculated as:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}, \qquad (1)$$

where TP designates the number of true positive predictions, TN the number of true negatives, FP the number of false positives and FN the number false negatives. We consider female and male patients as "positives" and "negatives", respectively. The MCC-values according to formula (1) are not calculable in cases where the denominator in the calculation of the MCC-value is zero. Therefore, firstly, for each of the I training sets we separately sum up the TP-, the TN-, the FP- and the FN-values, respectively, over the $I-1$ test set evaluations. Secondly, we apply formula (1) to the summed up TP-, TN-, FP- and FN-values. Here, in some cases it occurred that formula (1) was still not applicable, because the denominator was zero also in case of the summed up TP-, TN-, FP- and FN-values. In each of these cases, the respective prediction rule either classified all observations as negative or positive, respectively so that $TP + FP$ or $TN + FN$, respectively, was zero. Such prediction rules, which simply assign all observations to one class are no more effective than random guessing. Therefore we simply assigned a MCC-value of zero in these rare cases where either $TP + FP$ or $TN + FN$ was zero. The MCC-values calculated using the summed up TP-, TN-, FP- and FN-values are denoted as MCC_{rule}. This measure reflects the mean cross-study prediction performance of a specific prediction rule evaluated on a arbitrary test dataset.

2.5 Study design

We vary five parameters in our analyses:

- normalization type: "addon normalization" (`addon`), "separate normalization" (`separate`)
- batch effect adjustment method: "No batch effect adjustment" (`none`), "ComBat" (`combat`), "mean-centering" (`meanc`), "standardization" (`stand`), "ratio-G" (`ratiog`), "ratio-A" (`ratioa`), "fast frozen SVA" (`fsva_f`), "exact frozen SVA" (`fsva_e`), "FAbatch" (`fabatch`)
- Training set size: "original size of dataset, but maximal 70 observations" (`trainlarge`), "20 observations" (`trainsmall`)
- Test set size: "original size of dataset, but maximal 70 observations" (`testlarge`), "20 observations" (`testsmall`), "5 observations" (`testverysmall`)
- Classification method: "Linear Discriminant Analysis using Partial Least Squares" (`PLS-LDA`), "`PLS-LDA` using the 2000 variables with the

smallest p-values out of two-sample t-tests" (`PLS-LDA_varsel`), "Componentwise boosting with logistic loss function (LogitBoost)" (`Boost`), "`Boost` using the 2000 variables with the smallest p-values out of two-sample t-tests" (`Boost_varsel`), "Nearest Shrunken Centroids" (`NSC`), "Random Forests" (`RF`), "k-Nearest-Neighbors classification using the 2000 variables with the smallest p-values out of two-sample t-tests" (`kNN_varsel`)

For k-Nearest-Neighbors (kNN) classification we perform initial variable selection for the following reason: unlike the other classification methods used in our analysis kNN classification does not weigh the variables by importance, wherefore its performance relatively highly depends on the quality of the variables included (Pohjalainen *et al.*, 2015). We consider all possible combinations of the values of these parameters, leading to a total of 756 settings ($2 \times 9 \times 2 \times 3 \times 7$). In cases where subsetting was necessary, we drew random samples from the datasets. Here, except in the case of `testverysmall`, we ensured that the smaller class was represented by at least five observations. Because we consider all possible pairs of training and test datasets, for each setting there exist 34 MCC_{rule}-values, each corresponding to a specific training dataset.

All R-code written to produce and evaluate our results is available online from the Supplementary Materials.

3 Results

Supplementary Figures S1 to S7 shows boxplots of the MCC_{rule}-values for each classification method, separated by batch effect adjustment method, normalization type, training and test dataset size. In the following, unless otherwise stated, the description of the results of our study is based on these plots.

3.1 Addon quantile normalization

In many of the studied settings without addon batch effect adjustment, addon normalization improved performance and in no setting did it lead to a decrease of classification performance, see Figure 1. While addon batch effect adjustment, if applicable, is usually more effective than addon normalization, in some situations it impairs performance, see further down. Given performance was not impaired by addon normalization in any of the settings we studied, we recommend the following: addon normalization should be performed whenever test observations are not available in groups and addon batch effect adjustment is thus not possible and in the case of settings where addon batch effect adjustment tends not to improve results (see further down). While both approaches improve performance, there is no

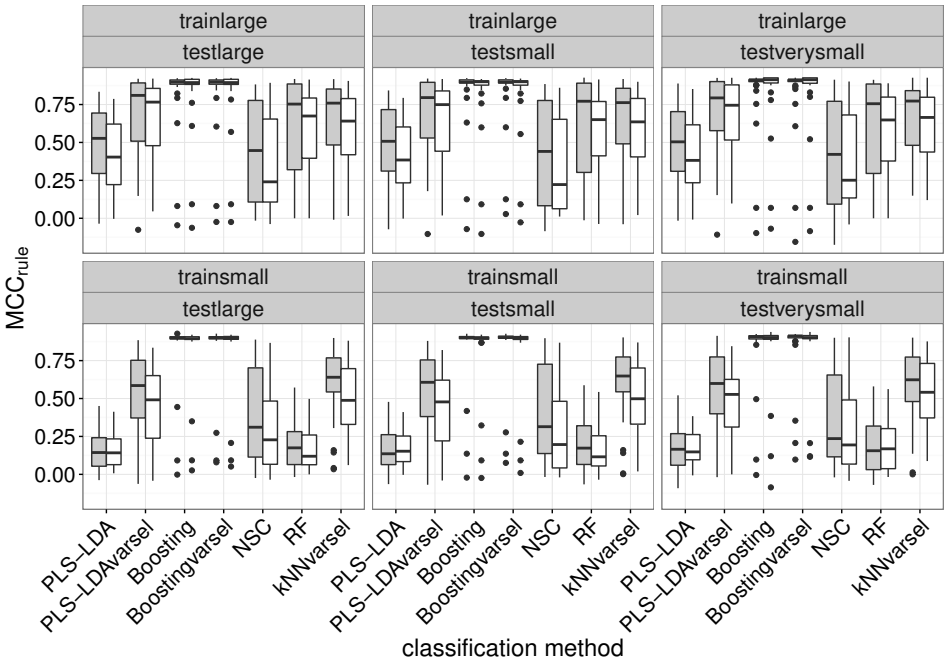

Figure 1: MCC_{rule}-values for the 34 datasets for each setting without addon batch effect adjustment. The grey and the white boxplots show the results when using addon and separate normalization, respectively.

benefit from using addon batch effect adjustment in combination with addon normalization over using addon batch effect adjustment alone. Instead in many cases the performance is slightly deterioriated by additional addon normalization, see also the discussion. Therefore, from now on, we will only examine the results obtained by either addon normalization or addon batch effect adjustment but not by the combination of these two.

3.2 Addon batch effect adjustment

3.2.1 Influence of training and test set size

As expected, the MCC_{rule}-values tended to be smaller for the setting with small training datasets. A striking observation is that `RF` did only deliver useful predictions in the setting with larger training datasets. Sonka *et al.* (2014) already noted that random forests do not generalize well when using small datasets as training data. While the size of the training dataset did influence the cross-study prediction performance, it had almost no influence on the benefit yielded by addon normalization and addon batch effect adjustment. For the sake of clarity, we will thus in the following focus only on the setting with large training datasets.

Figure 2 shows the median MCC_{rule}-values for all settings with large training datasets and separate normalization. Generally, we observe hardly any differences between the results for addon batch effect adjustment when using a large and a small test dataset. However, when using a very small test dataset (five observations), the MCC_{rule}-values tend to become considerably smaller. This frequently leads to a small deterioration by addon batch effect adjustment. Therefore we can further conclude that, as a general rule, for addon batch effect adjustment to be effective very small test datasets should be avoided.

3.2.2 Specific classification methods

Given a test dataset that comprises more than a few observations, whether or not batch effect adjustment significantly improved the result depended on the classification method used. For most classification methods we saw an improvement by certain addon batch effect adjustment methods, see the next subsection for details. The exceptions were `Boost` (Supplementary Figure S3), `Boost_varsel` (Supplementary Figure S4) and `RF` (Supplementary Figure S6).

For `Boost` and `Boost_varsel` the observed deterioration by addon batch effect adjustment can likely be explained by the very good performance these methods exhibit without addon batch effect adjustment. Here, without batch effect adjustment the MCC_{rule}-values are very high and have almost zero variance apart from a few outliers (Figure 1). See the disussion for an explanation why boosting may especially be suitable in cross-study prediction. Closer inspection of the results revealed that the small variance of the MCC_{rule}-values observed for boosting without batch effect adjustment can be explained as follows: there are two to three datasets which perform bad as training and test datasets, while the other datasets exhibit an almost perfect performance. This has the effect that the MCC_{rule}-values for the good training datasets are very similar, because in these cases the summed up values used for calculating the MCC_{rule}-values are almost the same: the corresponding prediction rules classify the observations from the good test datasets almost perfectly and that from the bad test datasets equally worse. What is more, the variability associated with the batch effect adjustment can change predictions. In the cases where the predictions are already almost perfect, the performance is necessarily diminished by changes in some predictions. This explains why in the cases of `Boost` and `Boost_varsel` the MCC_{rule}-values tend to be lower after batch effect adjustment. The outliers in the lower domain mentioned further above show the results obtained when using the bad datasets as training datasets.

As noted above, also for `RF` no batch effect adjustment method lead to an improvement of the prediction. The boxplots corresponding to `combat`, `meanc`, `stand`, `ratiog` and `ratioa` have a very similar form. These meth-

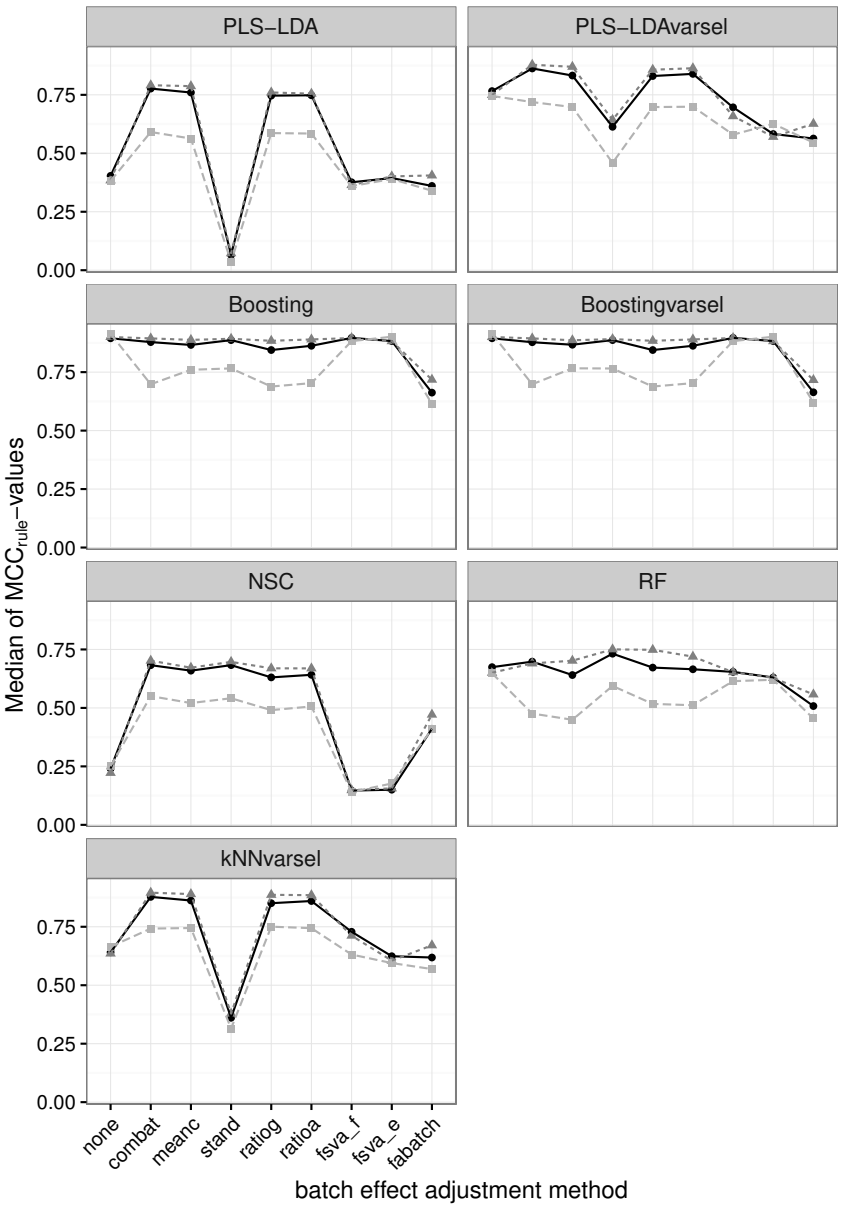

Figure 2: Medians of the MCC_{rule}-values over the 34 datasets for each setting with large training dataset and separate normalization. The solid, dotted and dashed lines show the results obtained when using a large, small and very small test dataset, respectively.

ods have in common that they assimilate the means of the training and test data. Closer inspection of the results revealed that the small 25%-quartiles of the MCC_{rule}-values displayed in the boxplots (Supplementary Figure S6) can be attributed to the results of five training datasets. Here, the results substantially worsened by batch effect adjustment through the methods mentioned further above. These five datasets were imbalanced to a significantly stronger degree than the other datasets (p-value out of two-sided wilcoxon test: 0.0055). When excluding the results corresponding to training datasets with imbalanced class frequencies, batch effect adjustment by the methods mentioned above lead to a quite strong improvement of the prediction accuracy of `RF` (results not shown). The bad performance for these problematic datasets is not directly due to the fact that in these cases the class frequencies are imbalanced in the training data. Instead, the actual reason is that the class frequencies tend to be strongly different in the test datasets than in the training data if the latter is imbalanced. In the discussion we will explain the mechanism by which `RF` in particular suffers by differing class frequencies between training and test data, when used in combination with batch effect adjustment methods which involve an assimilation of the means in training and test data.

For boosting, pre-selection of influential variables as performed by `Boost_varsel` in our study did not further improve results (Figure 1). By contrast PLS-LDA seems to be improved by initial supervised variable selection, which was also found by Li *et al.* (2007).

3.2.3 Performance of individual batch effect adjustment methods

As seen above, we marked down those settings in which addon batch effect adjustment was not valuable. In those settings where it did improve performance there were in general several well-performing methods with no clear ranking between them (Figure 2). Four methods were always among the best here: `combat`, `meanc`, `ratiog` and `ratioa`. While also `stand` was frequently among the best methods, it was very bad for `PLS-LDA`, `PLS-LDA_varsel` and `kNN_varsel`. Thus, the value of this method strongly depends on the classifier used, wherefore it cannot be recommended. In contrast to Luo *et al.* (2010) we could not find that `ratioa` and `ratiog` may be preferable over the other well-performing methods. `fsva_f`, `fsva_e` and `fabatch` did not improve performance in any of the settings and more importantly these methods were often harmful. Therefore these methods should not be used for cross-study prediction. Note that in the paper presenting `fsva_f` and `fsva_e` (Parker *et al.*, 2014) it is stated that these methods rely on similarity between training and test data, an assumption most often not given in cross-study prediction. Our study shows that these methods can also impair performance when the assumption of similarity cannot be made.

4 Discussion

4.1 Reasons for missing benefit from combining the two approaches

Used separately, both addon normalization and addon batch effect adjustment improved the performance of cross-study prediction under the conditions worked out in the previous section. However, we saw no additional gain in prediction performance by using addon batch effect adjustment in combination with addon normalization in comparison to using addon batch effect adjustment alone. Two explanations for this could be the following: 1.) the assimilation of the distribution of the test data to that of the training data by addon batch effect adjustment is not substantially improved by a preceding addon normalization. Generally, addon batch effect adjustment leads to a stronger assimilation of the distribution of the test data to that of the training data than addon normalization. The reason for this is that addon batch effect adjustment explicitly assimilates the distributions of the individual variables in the test data to that in the training data. By contrast, addon normalization merely assimilates the marginal distributions of the values belonging to the individual observations. The latter is however also, implicitely, performed by addon batch effect adjustment; 2.) the variability connected with the adjustment is increased by combining the two procedures.

4.2 Random Forests: impaired performance in the presence of differing class frequencies between training and test data

When the classes were imbalanced in the training data, the performance of `RF` was impaired to the same extent by all addon batch effect adjustment methods, which involve an assimilation of the variable means in training and test data. We attributed this to the fact that, if the classes are imbalanced in the training data, the class frequencies tend to be different in the test data than in the training data. In the context of conventional batch effect adjustment, Nygaard and Rødland (2016) already noted that mean-centering reduces the class differences when the classes are unevenly represented in the different batches. While all classifiers can be expected to suffer to some extent from variable mean adjustment if the class frequencies between training and test data are different, we expect this to be particularly a problem for random forests. In the following we will describe the mechanism responsible for the latter. The classification trees constituting a random forest iteratively divide the observations into subgroups of decreasing sizes. More precisely, in each iteration the subgroups are split into two smaller subgroups based on a threshold of an individual variable. Here, that threshold of that vari-

able (among a randomly chosen subset) is used that leads to the strongest separation of the two classes by the two resulting subgroups according to a specific criterion. As a result, the splits are performed in each case using that variable (ouf of the candidates), which has the greatest discriminatory power. The stronger the discriminatory power of a variable, the stronger it suffers from an adjustment of the means between training and test data, in case the class frequencies between the two are different. Here, the mean adjustment leads to the split point in the test data, which is actually the best, i.e. that which separates the two classes in the test observations best, being strongly shifted away from the best split point in the training data. The best split points in the test data are always shifted into the direction of the same class, namely that which is more frequent in the training than in the test data. Thus, when splitting the test observations according to the split points found in the training data, many of the test observations which belong to the class less frequent in the training data, are placed into the wrong subnodes. These wrong decisions accumulate as the test observations reach lower layers of the classification trees. In the extreme case, the random forest ultimately classifies all test observations as the class which is more frequent in the training than in the test data. For the five problematic training datasets mentioned above, we investigated whether we can observe this phenomenon in the case of ComBat. Here, indeed ComBat leads to classifiying almost all test observations as the class overrepresented in the training data. The latter was not the case without batch effect adjustment.

4.3 Boosting as a (potentially) robust method avoiding overfitting in the context of cross-study prediction

Boosting without batch effect adjustment almost perfectly predicted the class values across datasets and thus, batch effect adjustment tendentiously worsened the performance. It has been noted in the literature that boosting is quite resistant to overfitting i.e. to an over-adjustment to the training dataset, in particular in classification settings (Bühlmann and Hothorn, 2007). While LogitBoost can be prone to overfitting, this can be efficiently inhibited by early stopping of the boosting iterations (Bühlmann and Yu, 2008), as performed in our study. Conventionally the term "overfitting" refers to the phenomenon that a classifier is overly strongly adjusted to the specific observations in the training data. This can have the effect that it features an increased error frequency when applied to independent test observations following the same distribution as the training data. In the context of cross-study prediction, however, independent test observations follow a different distribution than the training data, which is due to batch effects, as already mentioned. Therefore, we have to consider a different kind of overfitting here. A classifier may not only be overly strong adjusted to the specific observations in the training data, but also to the distribution of the

training data. Such a classifier, which is too much adjusted to the particular behavior of the training data, may feature a bad generalizability to different, albeit similar data distributions. A classifier of this kind would have a low cross-validation error but a large cross-study prediction error. By contrast a classifier which is not overfitting the training data distribution could have quite a large cross-validation error, but a low cross-study prediction error. Accordingly, Bernau *et al.* (2014) found only a weak positive correlation between cross-validation and cross-study validation error in their study. The strong performance of boosting with early stopping suggests that this method may not only be resistant to overfitting the training observations, but also to overfitting the distribution of the training observations. Early stopping of the boosting iterations has the effect that only strong, coarse properties of the relationship between covariates and response in the training data are taken into account. These properties can be expected to be not induced by batch effects but common to all datasets studying the biological phenomenon of interest. As the number of boosting iterations increases, the classifier is increasingly well adjusted to the training data distribution. This, together with the fact that boosting is more prone to overfitting for other prediction settings than classification could explain why the CoxBoost algorithm was less suitable for cross-study prediction in the study by Bernau *et al.* (2014) than LogitBoosting was in our study. Similar to the number of iterations in boosting, also other classification methods feature tuning parameters which control the degree to which the algorithm adjusts itself to the training observations and in consequence also to the distribution of the training data. Examples include the shrinkage parameter Δ of `NSC` or the penalization parameter λ in $L1$- and $L2$-penalized logistic regression. Further research could focus on the influence of such parameters on the cross-study prediction performance of these methods. We feel that the number of iterations in boosting could be especially useful in this context. Firstly, this parameter has been seen to greatly influence the performance, see e.g. Seibold *et al.* (2016). Moreover, in each iteration the influence of only one variable is updated, wherefore boosting is not strongly dependent on the specific correlation structure of the dataset. Instead, new variables are consecutively taken into the model based on their importance with respect to explaining the target variable and the iterations are stopped as soon as the model is deemed complex enough.

4.4 Further possibilities for application

ComBat holds a special place among the four well-performing batch effect adjustment methods, because of the peculiarity that the training data is not altered in any way by the adjustment. As a consequence, ComBat addon adjustment could be employed to improve the prediction performance of already existing prediction rules in case the following requirements are met:

the training data which had been used to learn the prediction rule must be available and the observations to predict must arrive in groups of sufficient sizes.

In the analysis performed in this paper we have considered quantile normalization as part of RMA for Affymetrix data. However, quantile normalization is also used for many other biomolecular data types (Okoniewski and Miller, 2008; Schmid *et al.*, 2010; Bullard *et al.*, 2010; Staaf *et al.*, 2008; 't Hoen *et al.*, 2008). Therefore addon quantile normalization can also be used for other data types than Affymetrix data to improve the cross-study prediction performance of prediction rules obtained from these data types.

5 Conclusions

Assimilating the test data to the training data before the application of prediction rules obtained from gene expression data can considerably improve prediction accuracy. In this endeavor, both addon normalization and addon batch effect adjustment are recommendable, however not the combination of these two approaches. A requirement for addon batch effect adjustment to be effective is that the test observations are available in groups of sufficient sizes. In the latter case, addon batch effect adjustment using an appropriate method is preferable over addon normalization. The following addon batch effect adjustment methods are recommended and perform comparably well: `combat`, `meanc`, `ratiog` and `ratioa`. Strongly differing class frequencies between training and test data should be avoided, especially when using random forests as classification method. All methods applied in our study for assimilating training and test data are available in the R-package `bapred`, version 1.0 (Hornung and Causeur, 2016), available from CRAN.

Acknowledgements We thank Sarah Tegenfeldt for making valuable language corrections. This work was supported by the German Science Foundation (DFG-Einzelförderung BO3139/3-1 and BO3139/2-3 to Anne-Laure Boulesteix).

References

Bernau, C., Riester, M., Boulesteix, A.-L., Parmigiani, G., Huttenhower, C., Waldron, L., and Trippa, L. (2014). Cross-study validation for the assessment of prediction algorithms. *Bioinformatics*, **30**, i105–i112.

Bolstad, B. M., Irizarry, R. A., Åstrand, M., and Speed, T. P. (2003). A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, **19**, 185–193.

Boulesteix, A.-L. (2013). On representative and illustrative comparisons with real

data in bioinformatics: response to the letter to the editor by Smith et al. *Bioinformatics*, **29**, 2664–2666.

Boulesteix, A.-L., Lauer, S., and Eugster, M. J. (2013). A plea for neutral comparison studies in computational sciences. *PLoS ONE*, **8**, e61562.

Boulesteix, A.-L., Hable, R., Lauer, S., and Eugster, M. J. A. (2015). A statistical framework for hypothesis testing in real data comparison studies. *Am Stat*, **69**, 201–212.

Bühlmann, P. and Hothorn, T. (2007). Boosting algorithms: regularization, prediction and model fitting. *Stat Sci*, **22**, 477–505.

Bühlmann, P. and Yu, B. (2008). Response to Mease and Wyner, evidence contrary to the statistical view of boosting. *J Mach Learn Res*, **9**, 187–194.

Bullard, J. H., Purdom, E., Hansen, K. D., and Dudoit, S. (2010). Evaluation of statistical methods for normalization and differential expression in mRNA-Seq experiments. *BMC Bioinformatics*, **11**, 94.

Castaldi, P. J., Dahabreh, I. J., and Ioannidis, J. P. (2011). An empirical assessment of validation practices for molecular classifiers. *Brief. Bioinform*, **12**, 189–202.

Gatto, L., Hansen, K. D., Hoopmann, M. R., Hermjakob, H., Kohlbacher, O., and Beyer, A. (2016). Testing and validation of computational methods for mass spectrometry. *J. Proteome Res.*, **15**, 809–814.

Hansen, K. D. and Irizarry, R. A. (2012). Removing technical variability in RNA-seq data using conditional quantile normalization. *Biostatistics*, **13**, 204–216.

Hornung, R. and Causeur, D. (2016). *bapred: Batch effect removal and addon normalization (in phenotype prediction using gene data)*. R package version 1.0.

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

Irizarry, R. A., Hobbs, H., Collin, F., Beazer-Barclay, Y. D., Antonellis, K. J., Scherf, U., and Speed, T. P. (2003). Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, **4**, 249–264.

Johnson, W. E., Li, C., and Rabinovic, A. (2007). Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics*, **8**, 118–127.

Kolesnikov, N. *et al.* (2015). ArrayExpress update–simplifying data submissions. *Nucleic Acids Res*, **43**, D1113–D1116.

Kostka, D. and Spang, R. (2008). Microarray based diagnosis profits from better documentation of gene expression signatures. *PLoS Comput Biol*, **4**, e22.

Leek, J. T. and Storey, J. D. (2007). Capturing heterogeneity in gene expression studies by surrogate variable analysis. *PLoS Genet.*, **3**, 1724–1735.

Li, G.-Z., Zeng, X.-Q., Yang, J. Y., and Yang, M. Q. (2007). Partial least squares based dimension reduction with gene selection for tumor classification. In J. Y. Yang, M. Q. Yang, M. M. Zhu, Y. Zhang, H. R. Arabnia, Y. Deng, and N. G. Bourbakis, editors, *Proceedings of the 7th IEEE International Conference on Bioinformatics and Bioengineering. Boston*, pages 1439–1444.

Luo, J. *et al.* (2010). A comparison of batch effect removal methods for enhancement of prediction performance using MAQC-II microarray gene expression data. *Pharmacogenomics J*, **10**, 278–291.

Nygaard, V. and Rødland, E. A. (2016). Methods that remove batch effects while retaining group differences may lead to exaggerated confidence in downstream analyses. *Biostatistics*, **17**, 29–39.

Okoniewski, M. J. and Miller, C. J. (2008). Comprehensive analysis of affymetrix exon arrays using BioConductor. *PLoS Comput Biol*, **4**, e6.

Parker, H. S., Bravo, H. C., and Leek, J. T. (2014). Removing batch effects for prediction problems with frozen surrogate variable analysis. *PeerJ*, **2**, e561.

Pohjalainen, J., Räsänen, O., and Kadioglu, S. (2015). Feature selection methods and their combinations in high-dimensional classification of speaker likability, intelligibility and personality traits. *Comput Speech Lang*, **29**, 145–171.

Scheerer, A., editor (2009). *Batch Effects and Noise in Microarray Experiments: Sources and Solutions*. Wiley Series in Probability and Statistics. Wiley, Hoboken.

Schmid, R. *et al.* (2010). Comparison of normalization methods for Illumina BeadChip HumanHT-12 v3. *BMC Genomics*, **11**, 349.

Seibold, H., Bernau, C., Boulesteix, A.-L., and De Bin, R. (2016). On the choice and influence of the number of boosting steps. Technical report 188, Department of Statistics, LMU.

Sonka, M., Hlavac, V., and Boyle, R., editors (2014). *Image Processing, Analysis, and Machine Vision*. Cengage Learning, Boston.

Staaf, J., Vallon-Christersson, J., Lindgren, D., Juliusson, G., Rosenquist, R., Höglund, M., Borg, Å., and Ringnér, M. (2008). Normalization of Illumina Infinium whole-genome SNP data improves copy number estimates and allelic intensity ratios. *BMC Bioinformatics*, **9**, 409.

't Hoen, P. A. C. *et al.* (2008). Deep sequencing-based expression analysis shows major advances in robustness, resolution and inter-lab portability over five microarray platforms. *Nucleic Acids Res*, **36**, e141.

van't Veer, L. J. and Bernards, R. (2008). Enabling personalized cancer medicine through analysis of gene-expression patterns. *Nature*, **452**, 564–570.