Comparison of merging strategies for building machine learning models on multiple independent gene expression data sets

Jessica Krepel | Magdalena Kircher | Moritz Kohls | Klaus Jung®

Institute for Animal Breeding and Genetics, University of Veterinary Medicine Hannover, Hannover, Germany

Correspondence
Klaus Jung, Institute for Animal Breeding and Genetics, University of Veterinary Medicine Hannover, Bünteweg 17p, D30559 Hannover, Germany.
Email: klaus.jung@tiho-hannover.de

Funding information
Deutsche Forschungsgemeinschaft, Grant/Award Number: 398066876/GRK 2485/1; Ministry of Science and Culture of the State Lower Saxony (Germany)

Abstract
High-dimensional gene expression data are regularly studied for their ability to separate different groups of samples by means of machine learning (ML) models. Meanwhile, a large number of such data are publicly available. Several approaches for meta-analysis on independent sets of gene expression data have been proposed, mainly focusing on the step of feature selection, a typical step in fitting a ML model. Here, we compare different strategies of merging the information of such independent data sets to train a classifier model. Specifically, we compare the strategy of merging data sets directly (strategy A), and the strategy of merging the classification results (strategy B). We use simulations with pure artificial data as well as evaluations based on independent gene expression data from lung fibrosis studies to compare the two merging approaches. In the simulations, the number of studies, the strength of batch effects, and the separability are varied. The comparison incorporates five standard ML techniques typically used for high-dimensional data, namely discriminant analysis, support vector machines, least absolute shrinkage and selection operator, random forest, and artificial neural networks. Using cross-study validations, we found that direct data merging yields higher accuracies when having training data of three or four studies, and merging of classification results performed better when having only two training studies. In the evaluation with the lung fibrosis data, both strategies showed a similar performance.

KEYWORDS
artificial neural networks, data fusion, discriminant analysis, gene expression data, high-dimensional data, LASSO, random forest, support vector machines

1 | INTRODUCTION

Differential gene expression analysis [10, 21] and gene set enrichment analysis [13, 15] are based on data from either microarray experiments, which are now conducted for over 20 years, or from RNA-seq experiments, which are now conducted for over 10 years in a wide range of biological questions. These technologies allow the simultaneous collection of gene expression data from several thousand genes which can be used for a better understanding of
molecular processes of pathogenesis or for the purpose of classification and prognosis. Of interest is, for instance, to predict the therapy response or to discover gene expression patterns that are characteristic for a certain disease to determine the disease subtype or differentiate between healthy and diseased individuals. The experimental factor is oftentimes binary, such as treated versus non-treated or diseased versus healthy individuals but can also happen to be multinomial, ordinal (e.g., healthy, mildly diseased, severely diseased), or time-to-event. For classification with high-throughput gene expression data, already at the beginning of the millennium, different classification and machine learning (ML) algorithms were submitted and reviewed. Among them are the discriminant analysis (DA), support vector machines (SVMs), artificial neural networks (ANNs), least absolute shrinkage and selection operator (LASSO), and random forest (RF). For example, Dudoit et al. [9] compared Fisher’s linear DA (LDA), maximum likelihood discriminant rules, nearest-neighbor classifiers, classification trees, and aggregating classifiers. In a recent article, Yu et al. [34] compared different architectures of neural networks for disease classification from omics data with traditional ML methods. Also in the context of classification in gene expression data, Liu et al. [19] combine several neural networks from the same data set in an ensemble. Patil and Parmigiani [24] use Lasso, CART, neural network, Mas-o-Menos, RFs, and model-based boosting to train replicable predictors in multiple studies.

Nowadays, there are gene expression data available in public databases for many biological and medical research questions. Oftentimes, several independent studies are performed on the same or similar question (e.g., due to follow-up studies that were conducted as the costs for the experiments continued to decrease). Their data can jointly be analyzed in a meta-analysis. A meta-analysis is the combined analysis of data from two or more independent studies. At different stages of the analysis, the information of the individual studies can be aggregated, for example, at the beginning by merging the data sets of the individual studies or at the end by combining the results of the separate analyses of all studies. There are a number of reasons for combining studies. For example, analyses of individual studies often have a low power due to small sample sizes versus a large number of genes. Performing a meta-analysis can overcome this problem by increasing the sample size [29, 32]. With respect to classification, analyses based on one study often yield too optimistic estimates of accuracy that cannot be reproduced in inter-study validation by other independent data sets. Meta-analysis overcomes this problem by using the information of multiple studies for the training of the classification model, leading to more generalizability and reliability [16, 32]. Diverse approaches of combining multiple studies for differential expression analyses [26, 32], selection of gene signatures [25], or GSEA [18] have been proposed. Kosch and Jung [18] found that data merging is superior to results merging in many scenarios with respect to gene set enrichment analysis. Although training of classifiers often includes a step of differential expression analysis as does GSEA, however, results are not transferable one by one, because ML also includes additional training of many hyperparameters. Several methods for combining multiple independent gene expression studies for ML have been proposed, but not with the specific comparison of data merging versus results merging. Patil and Parmigiani [24] also study methods for training classifiers on multiple independent gene expression studies but do not cover the approach of direct data merging. In contrast, Chang and Geman [5] use scenarios with merged data sets to compare different validation strategies but do not cover the approach of results merging. Similarly, Rashid et al. [25] focus on data merging but not on results merging.

Besides the above-mentioned ML methods which have been established in the field of gene expression data, a risen interest in using ANN can currently be observed in a wide range of applications. The increased usage of ANNs can be traced back to the increased power of computing infrastructure which is needed to train complex ANN models [6]. ANNs are computing systems that are inspired by the biological neural networks of animal brains [27]. They consist of an input layer, one or several hidden layers, and an output layer, each consisting of several units, also named neurons. The neurons between neighbored layers are connected, where each connection represents a weighted transformation of the output from the precursor neuron. For every connection, the weight is modified while the model is trained, in order to minimize the difference between the outcome of the output layer and the true label of the input samples. To calculate the output of a neuron, the values of all connected neurons of the previous layer are weighted with their weights and summed to form the net input. The net input is assigned to an activation level by the activation function to induce nonlinearity.

In this study, we compare different strategies to build classifiers from multiple independent gene expression data sets. We will only consider the two-group design, for example, with class labels diseased and healthy. The merging pipelines are combined with different methods for classification to identify which combination performs best in certain experimental conditions. The exemplarily considered classification methods are LDA, SVMs, LASSO, RF, and ANNs. There are already published studies that compare such classification methods in scenarios with only one single study, for example, as already mentioned,
Dudoit et al. [9] and Yu et al. [34]. Thus, the primary aim of this study remains the comparison of the merging pipelines. Within each merging strategy, a differential gene expression analysis is performed for dimension reduction, to select a set of genes that is then used as feature subset in the classification. The classification model is then trained on a subset of the gene expression data, the so-called training set. We consider two different merging pipelines with the merging at different stages of the analysis, namely the direct merging of the data sets (strategy A) and the merging of the classification results (strategy B). For validation, we used cross-study validation, such that in each fold one study in turn functions as validation study while the remaining $k - 1$ studies are used for training. Thus, there are as many folds as studies. While randomized cross-validation reports low error rates that cannot be reproduced in independent validation, cross-study validation results in higher, more reasonable estimated error rates [2, 5].

2 | MATERIALS AND METHODS

2.1 | General notations and differential expression analysis

To compare the different merging strategies, we first performed a simulation study with synthetic data, and second, a simulation study with real world data of three studies from the public data repository ArrayExpress [3]. The data and the data sampling process are described in Section 2.3. In each scenario, we used gene expression data from $K$ individual studies, where $n_k$ denotes the sample size of study $k$. Let $x_{gik}$ be the observed expression level for gene $g$ ($g = 1, \ldots, G$), sample $i$ ($i = 1, \ldots, n_k$) and study $k$ ($k = 1, \ldots, K$).

The analysis pipeline was constructed as follows: in each fold of the cross-study validation, feature selection was conducted within the training data, where the features to be used for classifier fitting were selected. In case of gene expression data, the features are the genes. Therefore, a differential gene expression analysis was conducted on the data using the R-package “limma” [28] to find out which genes are differentially expressed between the classes and consequently promising candidates for classification models. Within the limma-package, a linear model for each gene is fitted using moderated $t$-statistics. Next, the genes in the data set were ordered according to their p-values. Data for the first $M$ ($M \leq G$) of the ordered genes with the smallest p-values were used as features in the subsequent training of the classifier. The selected genes were re-indexed according to their ranking order by $g_m = g_1, \ldots, g_M$ with index $m = 1, \ldots, M$.

2.2 | Classifier methods

In this work, we only focus on two-class scenarios. Let $c_{ik} \in \{0, 1\}$ be the class label of sample $i$ in study $k$. Each of the classifier methods we are included in our analysis finally yields the probability that a certain sample $i$ belongs to group 1, denoted by $p(c_{ik} = 1)$, for each sample in the test data, and if $p(c_{ik} = 1) > 0.5$, then sample $i$ is assigned to class 1, otherwise it is assigned to class 0. Depending on the merging strategy, the analyses of the studies are merged at different stages of the general analysis process. The examined merging strategies are the merging of the data sets and the merging of the resulting class probabilities and are described in Section 2.4.

We selected exemplarily five classifier methods to perform the comparison between the two merging strategies: LDA, SVM, LASSO, RF, and ANN. Direct comparisons between these classifier methods have been performed by many others and was not the aim of our work. With the selection of these methods we wanted to cover a wide range of typical classifier techniques used in in the context of high-dimensional gene expression data. For each of the five classifier methods, established R-packages are available and were used for our comparison. In the following, we describe some specific settings we used for each of these methods.

For training of the ANNs, the R-package “Keras” was used [1], which provides an interface to the software Keras which itself reverts to Tensorflow. For our analyses, we used ANNs with one fully connected hidden layer with 24 units and with the activation function “ReLU” (rectified linear unit). For the output layer, the activation function “sigmoid” was used to get an output in the form of class probabilities value between 0 and 1, which in this case represents the probability of group 1, $p(c_{ik} = 1)$. An optimizer that implements the RMSprop algorithm which can deal with the problem of local optima was used to train the weights. The number of epochs depended on an early stopping function that stops the training when the monitored accuracy does not improve anymore to prevent an overfitting to the training data. The maximum number of epochs was set to 50. The binary cross-entropy loss was used as the loss function.

In RF, an ensemble of decision trees is built where each tree votes for a class and the majority decides on which class is predicted. For the training of the RFS, the R-package “randomForest” was used [4]. The number of variables randomly sampled as candidates at each split was set to 2 and the number of trees to grow was set to 100.

For training the LASSO models, we used the R-package “glmnet” [30] to fit a generalized linear model via penalized maximum likelihood where we chose the LASSO penalty. The factor $\lambda$ that specifies the strength of the
penalty was optimized using an additional cross validation within the training data, that is, either in the merged training data or in the individual studies.

The LDA is used to find a linear combination of features that separates two or more classes. Depending on the approach, LDA maximizes the ratio of between-class variance to within-class variance or the ratio of overall class variance to within-class variance. We used the functionality of the R-package “MASS” to train the LDA models.

SVMs choose a hyperplane with the maximum margin from both classes to separate them. If a linear separation is not possible, SVM maps the data into a higher dimensional input space. We used the R-package “e1071” [23] to train the SVM models.

### 2.3 Data sets

#### 2.3.1 Artificial data

The artificial data were generated according to Kosch and Jung [18]. In each of the generated $k$ studies, half of the samples are part of the control group with $c_{ik} = 0$, and the other half of the samples are part of the treatment group with $c_{ik} = 1$. For the control samples in study $k$, data were drawn from the multivariate normal distribution $\mathcal{N}_G(\mu_{c=0,k}, \Sigma)$ where $\mu_{c=0,k}$ the mean vector of the control group, was the null vector $0$, and the covariance matrix $\Sigma$ was constructed in an autoregressive way with elements $a_{gg} = \rho |g - g'|$. We arranged the data in a $(G \times \frac{n_k}{2})$ expression matrix of the control samples in study $k$. For the treatment samples in study $k$, data were drawn from $\mathcal{N}_G(\mu_{c=1,k}, \Sigma)$. We drew the mean vectors $\mu_{c=1,k}$ ($k = 1, \ldots, K$) for the treatment groups of the $k$ studies from the univariate normal distribution $N(0, \tau_k)$ in order to incorporate differentially expressed genes. $\mu_{c=1,k}$ reflects log fold changes in study $k$, which are negative for downregulated genes and positive for upregulated genes. Furthermore, the larger the value of $\tau_k$, the more extreme the fold changes in study $k$ can become. We arranged the data in an $(G \times \frac{n_k}{2})$ expression matrix of the treatment samples in study $k$. We set the number of genes $G = 1000$ and the sample size per study $n_k = 50$.

We added an additive and multiplicative batch effect between the $k$ studies using the model proposed by Johnson et al. [18]: $\gamma_{gk} = (x_{gk} + \gamma_{g,k}) \delta_{gk}$. The additive batch effect $\gamma_{gk}$ was drawn from a normal distribution with mean $y_k$ and variance $\epsilon_k^2$, and the multiplicative batch effect $\delta_{gk}$ was drawn from an inverse Gamma distribution with mean $\lambda_k$ and variance $\theta_k^2$. The batch effect parameters $\gamma_{gk}$ and $\delta_{gk}$ were drawn separately for each study $k$.

In the simulation study, different parameter settings were simulated. The accuracy of the classification was calculated for each of the combinations of different log fold change standard deviations $\tau_k = (0.1, 0.2)$ so that there were two sizes of class differences, different numbers of genes that are used as features for the classification $M = (50, 100, \ldots, 500)$, different number of studies $k = (3, 4, 5)$, and two different versions of batch effect modeling, either “constant batch effect” where mean and variance of the batch effect parameters remained constant over all studies with $y_k = 0$, $\epsilon_k = 1$, $\lambda_k = 1$, $\theta_k = 0.1$ for all $k$, or “variable batch effect” where they differed between the studies with $y_k$ between $0$ and $2$, $\epsilon_k$ between $1$ and $2$, $\lambda_k$ between $1$ and $3$, $\theta_k$ between $0.1$ and $0.2$. In each scenario, cross-study validation was applied to the classification, where in each fold a different study was used as validation study. The simulation was additionally repeated 100 times. In each fold and each repetition, the performance measures accuracy, sensitivity, and specificity were calculated. For the graphics, we took the mean values of the performance measures over the folds of the cross-study validation values in each repetition.

#### 2.3.2 Real world data from Gene Expression Omnibus

As real world data example, we downloaded the data from three studies available from the internet platform Gene Expression Omnibus under Series numbers GSE53845, GSE24206, and GSE17978. The data set GSE53845 was already published by DePianto et al. [8]. It contains microarray data of $40$ patients with idiopathic pulmonary fibrosis (IPF) and $8$ healthy patients. The number of genes that were measured in the study is $G = 41,000$. The data set GSE24206 was originally presented by Meltzer et al. [22]. It contains microarray data of $17$ patients with IPF and $6$ healthy patients, and the number of genes is $G = 54,613$. The data set GSE17978 [12] contains microarray data of $12$ patients with IPF and $6$ healthy patients and $G = 35,121$ genes. To evaluate the performance of the different merging strategies, we used between $M = 10$ and $M = 500$ of the top differentially expressed genes to train the classifiers. Again, cross-study validation was implemented.

### 2.4 Merging strategies

#### 2.4.1 Merge data sets

In merging strategy A, we merged the individual data sets of the studies to a metadata set before starting with the general analysis process (Figure 1). In the analysis with real
world data, in all three studies different microarrays were used and hence different gene identifiers. For merging of the data of both training studies and reduction to common genes in the test study, translation tables were downloaded from ArrayExpress or generated via the functionality of the R-package “biomaRt” [11] and used to translate all gene identifiers into the identifier “Entrez gene ID.” Data of common Entrez gene IDs found in all studies were merged into the common data set which included 17,711 gene expression values per sample and 79 samples (61 of class IPF and 18 of class healthy).

Batch effects between the studies were removed using the R function ComBat from the R-package sva. Between the training studies it was removed before training of the classifier and between the validation study and training studies it was removed before testing. The remaining steps of the analysis were conducted on the metadata set as described above in the general analysis process.

2.4.2 | Merge resulting class probabilities

In this merging strategy (Figure 2), the training data sets were analyzed individually during the complete analysis-pipeline, starting with the differential analysis to select the genes that function as features for the classification and ending with the fitting of the classification model. Before starting with the analysis, the data sets had to be subsetted for common genes, because the features of the resulting classifiers need to be part of the tested data set. But since the classifiers were not merged, the genes that were selected to be features in the classifiers, \( g_m = g_1, \ldots, g_M \), could differ between the classifiers. At the end of the analysis, the test data samples \( j^\text{test}_k = 1, \ldots, n^\text{test}_k \) of all studies \( k \) formed a merged test data set. Let \( \text{mod}_k \) be the individual classification model of study \( k \). All classification models \( \text{mod}_k \) calculated class probabilities \( p_{\text{mod}_k}(c^\text{test}_k = 1) \) for each sample in the test data set. For each test sample, we calculated the mean of the predicted class probabilities of all models to predict a class \( \overline{c^\text{test}_k} \) (Equation (1)). According to Kittler et al. [17] this as “sum rule” defined technique is the best way to combine a classifier (“the sum rule can be viewed to be computing the average a posteriori probability for each class over all the classifier outputs”; “the sum rule outperformed other classifier combinations schemes” [17]).

\[
\text{mean}(p(c^\text{test}_k = 1)) = \frac{\sum_{\text{mod}_k} p_{\text{mod}_k}(c^\text{test}_k = 1)}{K}.
\]  

3 | RESULTS

The two different merging strategies were conducted in combination with the five different classification methods and the different data (artificial and real world data) for all parameter settings described in Sections 2.3.1 and 2.3.2. We first present the results for the artificial data and then for the real world data.

3.1 | Artificial data

Figure 3 shows the results of the classification using ANNs and the merging strategies A (data merging) and B (results
In this merging strategy, separate classifiers are fitted to the data of the individual studies from the training set and applied separately to the test data set. Class probabilities for each sample in the test data set are then averaged.

![Diagram of the merging strategy](image)

**Figure 2** In this merging strategy, separate classifiers are fitted to the data of the individual studies from the training set and applied separately to the test data set. Class probabilities for each sample in the test data set are then averaged.

**Figure 3** Accuracy of the two different merging strategies when being applied with artificial neural nets as classifier on the artificial data sets. Points mark the mean of 100 simulation runs, lower and upper error bars mark the minima and maxima, respectively.

Merging (i.e., variable batch effect), strategy A performs a few percent better than under scenarios with constant batch effects. Strategy B also improves slightly when there are more training studies, the accuracy ranges between 80% and 90% under a high differentiability of data sets and between 60% and 65% under a low differentiability. When there are only two training studies, strategy A performs worse than strategy B, while in cases with three and four training studies, strategy A performs
FIGURE 4  Accuracy of the two different merging strategies when being applied with LASSO as classifier on the artificial datasets. Points mark the mean of 100 simulation runs, lower and upper error bars mark the minima and maxima, respectively.

FIGURE 5  Accuracy of the two different merging strategies when being applied with random forest as classifier on the artificial datasets. Points mark the mean of 100 simulation runs, lower and upper error bars mark the minima and maxima, respectively.

better. Figure S1 shows the sensitivity of the simulation, Figure S2 shows the specificity. Sensitivity and specificity are similar to the accuracy.

Figure 4 shows the results of the classification using LASSO. Like with ANNs, the performance of strategy A is highly influenced by the number of training studies.
while the performance of strategy B is most influenced by the differentiability between the classes. The accuracy of strategy B ranges between 80% and 100% under high differentiability and between 60% and 70% under low differentiability. As opposed to the results seen with ANNs, the accuracy increases only with an increasing gene
number of up to 300–400 genes in the classifier and situational decreases afterwards. The variability of the accuracy is generally much lower here than in the simulations with ANNs. The sensitivity (Figure S3) shows a higher variability while the specificity (Figure S4) is similar to the accuracy.

Figure 5 shows the results of the classification using RF. The tendencies are similar to the results of LASSO as classifier except for about 10% lower overall accuracy with RF. Sensitivity and specificity (Figures S5 and S6) are similar.

Figure 6 shows the results of the classification using SVMs. Here, the performance of the classifier seems to be unaffected by the number of genes in the classifier. The performance of the SVMs with respect to the merging strategies, differentiability, number of training studies, and type of batch effects is similar to the results of the other classifiers. Sensitivity and specificity (Figures S7 and S8) are similar.

Figure 7 shows the results of the classification using LDA. The variability of the accuracy is higher than observed in the simulations with ANNs. With the results merging strategy, the accuracy is highest with 50 genes in the classifier and then decreases with higher gene numbers. In the data merging strategy, the influence of the gene number in the classifier is unclear and there seems to be a
gap in the accuracy at 150 or 200 genes. The results of sensitivity and specificity (Figures S9 and S10) are similar to the accuracy but with much less variability.

### 3.2 Real world data

Figure S21 shows the principal component plots of the merged data of the three independent studies. While studies form individual clusters before batch effect removal, a separation of samples from diseased and healthy individuals can be seen after batch effects removal. Only a small overlap of differentially expressed genes between the three studies can be observed (Figure S22), pointing at a possible weak classification performance by either approach.
The graphic upper left of Figure 8 shows the results of the classification using ANN and the merging strategies A (data merging) and B (results merging) applied to real world data. In the graphics, the lower end of the lines represents the minimum of the three folds in the cross-study validation, the upper end represents the maximum, and the point represents the mean. The accuracy rises until the number of genes in the classifier is 40 in strategy A and 50 in strategy B, afterwards it slowly decreases. In the strategy B, the accuracy reaches another maximum. The overall mean accuracy is 0.89 in strategy A and 0.92 in strategy B and the accuracies range between 0.62 and 1 in strategy A and between 0.77 and 1 in strategy B. Therefore, the performances of the strategies are similar, but the mean accuracy of strategy B was 3% higher. The sensitivity (Figure S11) performs better at low numbers of genes in the classifier while the specificity (Figure S12) is low (around 0.77) at 10 genes in the classifier and strongly increases.

The graphic upper right of Figure 8 shows the results of the classification using LASSO and the merging strategies A (data merging) and B (results merging) applied to real world data. From 10 genes in the classifier to 50 genes, the accuracy of strategy A rises faster than of strategy B, therefore strategy A outperforms strategy B when there are 50 genes in the classifier. From 150 genes in the classifier onwards, the accuracies of the strategies are at the same level again. The sensitivity (Figure S13) is higher (near 1) than the accuracy while the specificity (Figure S14) is lower with a mean around 0.71 and with a high range from minimal 0.05 to maximal 1.

The graphic middle left of Figure 8 shows the results of the classification using RF and the merging strategies A (data merging) and B (results merging) applied to real world data. The accuracy does not change significantly depending on the number of genes in classifier or on the merging strategy. The highest mean accuracy is reached by strategy A at 30 genes in the classifier (0.88) and the lowest is reached by strategy A at 350 genes in the classifier (0.81) therefore the performance of strategy A more depends on the number of genes in the classifier. Strategy B shows a higher range of the accuracy. The sensitivity (Figure S15) is higher (near 1) than the accuracy while the specificity (Figure S16) is lower (mean around 0.63) with a high range from minimal 0 to maximal 1.

The graphic middle right of Figure 8 shows the results of the classification using SVMs and the merging strategies A (data merging) and B (results merging) applied to real world data. The accuracy of strategy B constantly lies 2%–4% above the accuracy of strategy A. The highest mean accuracy of 0.92 is reached by strategy B at 40 genes in the classifier. The sensitivity (Figure S17) is higher (near 1) than the accuracy while the specificity (Figure S18) is lower (mean around 0.66) with a high range from minimal 0 to maximal 1.

The graphic lower left of Figure 8 shows the results of the classification using LDA and the merging strategies A (data merging) and B (results merging) applied to real world data. Here, the performance of strategy B falls from 10 to 50 genes in the classifier from a mean accuracy of 0.87 to 0.53. At 100 genes in the classifier the accuracy is at 0.87 again. The performance of strategy A on the other hand rises from 10 to 50 genes in the classifier and afterwards decreases. The ranges of the accuracy are higher than with the other classification methods. The sensitivity (Figure S19) shows similar pattern but the sensitivity in strategy A is near 1 in the area of 10–50 genes in the classifier and the sensitivity in strategy B is near 1 in the area of 50–500 genes in the classifier. The mean specificity (Figure S20) is 0.71 with a range from 0 to 1.

### TABLE 3

| Classifier | Best merging strategy |
|------------|-----------------------|
| ANN        | Results merging       |
| LASSO      | Data merging          |
| RF         | None                  |
| SVM        | Results merging       |
| LDA        | None                  |

#### 3.3 Comparison of results of the different data sets and different ML methods

In the simulations with artificial data (Tables 1 and 2), in case of three or four training studies, strategy A tends to perform better than strategy B while in case of two training studies, strategy B tends to perform better. In the simulations with real world data (Table 3), there was no strategy clearly superior to the other.

#### 4 DISCUSSION

For a joint analysis of high-dimensional gene expression data, it is often possible to merge the data directly, which corresponds to strategy A. This option is usually not given for meta-analyses of clinical trials which are mostly performed in a two-stage approach by merging the results of the individual studies (corresponding to strategy C). This is often referred to as a one-stage meta-analysis. Although
there are some exceptions where the original data are merged, most meta-analyses of clinical trials are based on results merging.

The focus of this work was to aggregate multiple independent studies for fitting classifier models, thus a somewhat different idea than that of a typical meta-analysis. In our simulation with artificial data, we found that the strategy of data merging outperformed the results merging when the number of training studies was higher than two while mostly the results merging strategy outperformed the data merging when there were two studies for training. A low differentiability between the groups negatively affects the performance of the results merging strategy more than the data merging strategy. In the simulations with the real world data, there was no strategy clearly superior to the other. A benefit of strategy A is that the number of samples available to train the classification model is higher due to the early data merging so that the resulting model is more accurate. Especially neural networks need a great amount of data to work properly so that strategy A has a benefit here. Furthermore, strategy A accounts for batch effects between the studies. Therefore, its performance depends on the performance of the step for batch effect removal. Batch effect removal with the method “Combat” seemed to work well in our analysis. When Taminau et al. [32] compared merging strategies in the context of differential gene expression analysis, they also found that the data merging strategy was leading to a higher power of the analysis. Xu et al. [33] came to similar conclusions of why direct data merging might perform better by stating that “The major limitation of meta-analyses is that the small sample sizes typical of individual studies, coupled with variation due to differences in study protocols, inevitably degrades the results. Also, deriving separate statistics and then averaging is often less powerful than directly computing statistics from aggregated data.”

A general problem in data merging is the lack of common identifiers [7]. When using the merging strategies with several real-world studies, the data sets had to be subsetted for common genes that were found in all studies. The data therefore was reduced to overlapping genes whose gene IDs could be translated via translation tables. This limits to some extend the ability to generalize our conclusions because the translation process could be optimized and thus improve the performance of the classifiers. Also, a well-considered selection of the studies from medical or biological as well as statistical view is indispensable to guarantee a good performance of the classifier. In this study, the methods were applied to microarray data which are usually fluorescence values on a continuous metric scale. In further research, the methods can also be evaluated on RNA-seq data which usually presents as discrete count data. Other methods for batch effect removal would be necessary then [20].

ACKNOWLEDGMENTS
This work was supported by the Deutsche Forschungsgemeinschaft (DFG, German 382 Research Foundation) — 398066876/GRK 2485/1, and by the Ministry of Science and Culture of the State Lower Saxony (Germany) through the project FibrOmics. Open Access funding enabled and organized by Projekt DEAL.

CONFLICT OF INTEREST
The authors declare no potential conflict of interest.

DATA AVAILABILITY STATEMENT
Data sharing not applicable—no new data generated.

ORCID
Klaus Jung https://orcid.org/0000-0001-9955-748X

REFERENCES
1. J. J. Allaire and F. Chollet, keras: R interface to ‘Keras’. R package version 2.3.0.0. https://CRAN.R-project.org/package=keras, 2020.
2. C. Bernau, M. Riester, A.-L. Boulesteix, P. Giovanni, C. Huttenhower, L. Waldron, and L. Trippa, Cross-study validation for the assessment of prediction algorithms, Bioinformatics 30 (2014), i105–i112. https://doi.org/10.1093/bioinformatics/btu279
3. A. Brazma, H. Parkinson, U. Sarkans, M. Shoajatalab, J. Vilo, N. Abeygunawardena, E. Holloway, M. Kapushesky, P. Kemmeren, G. Garcia Lara, A. Oezcimen, P. Roca-Serra, and S.-A. Sansone, ArrayExpress – A public repository for microarray gene expression data at the EBI, Nucleic Acids Res. 31(1) (2003), 68–71.
4. L. Breiman, Random forests, Mach. Learn. 45(1) (2001), 5–32.
5. L.-B. Chang and D. Geman, Tracking cross-validated estimates of prediction error as studies accumulate, J. Amer. Statist. Assoc. 110(511) (2015), 1239–1247. https://www.jstor.org/stable/24739720
6. F. Chollet and J. J. Allaire, Deep learning mit R und Keras, 1st ed., mitp VerlagsGmbH & Co. KG, Frechen, 2018.
7. R. D. De Veaux, R. W. Hoerl, and R. D. Sneek, Big data and the missing links, Stat. Anal. Data Min. 9(6) (2016), 411–416.
8. D. J. DePianto, S. Chandriani, A. R. Abbas, G. Jia, E. N’Diaye, P. Caplazi, S. E. Kauder, S. Biswas, S. K. Karnik, C. Ha, Z. Modrusan, M. A. Matthey, J. Kukreja, H. R. Collard, J. G. Egen, P. J. Wolters, and J. R. Arron, Heterogeneous gene expression signatures correspond to distinct lung pathologies and biomarkers of disease severity in idiopathic pulmonary fibrosis, Thorax 70 (2015), 48–56. https://doi.org/10.1136/thoraxjnl-2013-204596
9. S. Dudoit, J. Fridlyand, and T. P. Speed, Comparison of discrimination methods for the classification of tumors using gene expression data, J. Am. Stat. Assoc. 97(457) (2002), 77–87.
10. S. Dudoit, J. P. Shaffer, and J. C. Boldrick, Multiple hypothesis testing in microarray experiments, Stat. Sci. 18(1) (2003), 71–103.
11. S. Durinck, P. T. Spellman, E. Birney, and W. Huber, *Mapping identifiers for the integration of genomic datasets with the R or Bioconductor package biomaRt*, Nat. Protoc. 4 (2009), 1184–1191.

12. M. C. Emblom-Callahan, M. K. China, O. A. Shlobin, S. Ahmad, E. S. Reese, E. P. R. Iyer, D. N. Cox, R. Brenner, N. A. Burton, G. M. Grant, and S. D. Nathan, *Genomic phenotype of non-cultured pulmonary fibroblasts in idiopathic pulmonary fibrosis*, Genomics 96 (2010), 134–145. https://doi.org/10.1016/j.ygeno.2010.04.005

13. J. J. Goeman and P. Bühlmann, *Analyzing gene expression data in terms of gene sets: Methodological issues*, Bioinformatics 23(8) (2007), 980–987.

14. W. E. Johnson and C. Li, *Adjusting batch effects in microarray expression data using empirical Bayes methods*, Biostatistics 8(1) (2007), 118–127.

15. J. Jung, B. Becker, E. Brunner, and T. Beißbarth, *Comparison of global tests for functional gene sets in two-group designs and selection of potentially effect-causing genes*, Bioinformatics 27(10) (2011), 1377–1383.

16. S. H. Kim, C.-W. Lin, and G. C. Tseng, *MetaTSP: A meta-analytic top-scoring pair method for robust cross-study validation of omics prediction analysis*, Bioinformatics 32(13) (2016), 1966–1973.

17. J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, *On combining classifiers*, IEEE Trans. Pattern Anal. Mach. Intell. 20(3) (1998), 226–239.

18. R. Kosch and K. Jung, *Conducting gene set tests in meta-analyses of transcriptome expression data*, Res. Synth. Methods 10 (2019), 99–112.

19. B. Liu, Q. Cui, T. Jiang, and S. Ma, *A combinatorial feature selection and ensemble neural network method for classification of gene expression data*, BMC Bioinformatics 5 (2004), 136. https://doi.org/10.1186/1471-2105-5-136

20. Q. Liu and M. Markatou, *Evaluation of methods in removing batch effects on RNA-seq data*, Infect. Dis. Transl. Med. 2(1) (2016), 3–9.

21. M. I. Love, W. Huber, and S. Anders, *Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2*, Genome Biol. 15(12) (2003), 550.

22. E. B. Meltzer, W. T. Barry, T. A. D’Amico, R. D. Davis, S. S. Lin, M. W. Onaitis, L. D. Morrison, T. A. Sporn, M. P. Steele, and P. W. Noble, *Bayesian probit regression model for the diagnosis of pulmonary fibrosis: Proof-of-principle*, BMC Med. Genomics 4 (2011), 70. https://doi.org/10.1186/1755-8794-4-70

23. D. Meyer, E. Dimitriadou, K. Hornik, A. Weingessel, and F. Leisch, *e1071: Misc functions of the Department of Statistics, Probability Theory Group (Formerly: E1071)*, TU Wien. R package version 1.7-3. https://CRAN.R-project.org/package=e1071, 2019.

24. P. Patil and G. Parmigiani, *Training replicable predictors in multiple studies*, Proc. Natl. Acad. Sci. U. S. A. 115(11) (2017), 2578–2583. https://doi.org/10.1073/pnas.1708283115

25. N. U. Rashid, Q. Li, J. J. Yeh, and J. G. Ibrahim, *Modeling between-study heterogeneity for improved replicability in gene signature selection and clinical*, J. Am. Stat. Assoc. 115 (2020), 1125–1138. https://doi.org/10.1080/01621459.2019.1671197

26. A. Rau, G. Marot, and F. Jafrezic, *Differential meta-analysis of RNA-seq data from multiple studies*, BMC Bioinformatics 15 (2014), 91.

27. D. Rey and K. F. Wender, *Neuronale Netze. Eine Einführung in die Grundlagen, Anwendung und Datenauswertung*, 3rd ed., Hogrefe Verlag, Bern, 2018.

28. M. E. Ritchie, B. Phipson, D. Wu, Y. Hu, C. W. Law, W. Shi, and G. K. Smyth, *limma powers differential expression analyses for RNA-sequencing and microarray studies*, Nucleic Acids Res. 43(7) (2015), e47. https://doi.org/10.1093/nar/gkv007

29. F. Rohart, A. Eslami, N. Matigian, S. Bougeard, and K.-A. Le Cao, *MINT: A multivariate integrative method to identify reproducible molecular signatures across independent experiments and platforms*, BMC Bioinformatics 18 (2017), 128. https://doi.org/10.1186/s12859-017-1553-8

30. N. Simon, J. Friedman, T. Hastie, and R. Tibshirani, *Regularization paths for Cox’s proportional hazards model via coordinate descent*, J. Stat. Softw. 39(5) (2011), 1.

31. J. A. K. Suykens and J. Vandewalle, *Least squares support vector machine classifiers*, Neural. Process. Lett. 9 (1999), 293–300.

32. J. Taminau, C. Lazar, S. Meganck, and A. Nowé, *Comparison of merging and metaanalysis as alternative approaches for integrative gene expression analysis*, ISRN Bioinformatics 2014 (2014), 345106. https://doi.org/10.1155/2014/345106

33. L. Xu, A. C. Tan, R. L. Winslow, and D. Geman, *Merging microarray data from separate breast cancer studies provides a robust prognostic test*, BMC Bioinformatics 9 (2008), 125.

34. H. Yu, D. C. Samuels, Y.-y. Zhao, and Y. Guo, *Architectures and accuracy of artificial neural network for disease classification from omics data*, BMC Genomics 70 (2015), 167. https://doi.org/10.1186/s12864-019-5546-z

SUPPORTING INFORMATION

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

How to cite this article: J. Crepel, M. Kircher, M. Kohls, and K. Jung. *Comparison of merging strategies for building machine learning models on multiple independent gene expression data sets*, Stat. Anal. Data Min.: ASA Data Sci. J. 15 (2022), 112–124. https://doi.org/10.1002/sam.11549