*Bioinformatics*, 37(11), 2021, 1521–1527
doi: 10.1093/bioinformatics/btaa986
Advance Access Publication Date: 27 November 2020
Original Paper

Gene expression

# Robustifying genomic classifiers to batch effects via ensemble learning

**Yuqing Zhang**[1], **Prasad Patil**[2], **W. Evan Johnson** [2,3] **and Giovanni Parmigiani**[4,5,*]

[1]Clinical Bioinformatics, Gilead Sciences, Inc., Foster City, CA 94404, USA, [2]Department of Biostatistics, Boston University School of Public Health, Boston, MA 02118, USA, [3]Division of Computational Biomedicine, Boston University School of Medicine, Boston, MA 02118, USA, [4]Department of Data Sciences, Dana-Farber Cancer Institute, Boston, MA 02215, USA and [5]Department of Biostatistics, Harvard T.H. Chan School of Public Health, Boston, MA 02115, USA

*To whom correspondence should be addressed.

Associate Editor: Janet Kelso

Received on February 15, 2020; revised on October 20, 2020; editorial decision on November 9, 2020; accepted on November 13, 2020

## Abstract

**Motivation:** Genomic data are often produced in batches due to practical restrictions, which may lead to unwanted variation in data caused by discrepancies across batches. Such 'batch effects' often have negative impact on downstream biological analysis and need careful consideration. In practice, batch effects are usually addressed by specifically designed software, which merge the data from different batches, then estimate batch effects and remove them from the data. Here, we focus on classification and prediction problems, and propose a different strategy based on ensemble learning. We first develop prediction models within each batch, then integrate them through ensemble weighting methods.

**Results:** We provide a systematic comparison between these two strategies using studies targeting diverse populations infected with tuberculosis. In one study, we simulated increasing levels of heterogeneity across random subsets of the study, which we treat as simulated batches. We then use the two methods to develop a genomic classifier for the binary indicator of disease status. We evaluate the accuracy of prediction in another independent study targeting a different population cohort. We observed that in independent validation, while merging followed by batch adjustment provides better discrimination at low level of heterogeneity, our ensemble learning strategy achieves more robust performance, especially at high severity of batch effects. These observations provide practical guidelines for handling batch effects in the development and evaluation of genomic classifiers.

**Availability and implementation:** The data underlying this article are available in the article and in its online supplementary material. Processed data is available in the Github repository with implementation code, at https://github.com/zhangyuqing/bea_ensemble.

**Contact:** gp@jimmy.harvard.edu

**Supplementary information:** Supplementary data are available at *Bioinformatics* online.

## 1 Introduction

Statistical learning models based on genomic information have been widely used for prognostication and prediction across a range of precision medicine applications, including cancer (Golub *et al.*, 1999; Papaemmanuil *et al.*, 2016; Riester *et al.*, 2014; Silvestri *et al.*, 2015) and infectious diseases (Leong *et al.*, 2018; Seib *et al.*, 2009), and have shown great potential in facilitating clinical and preventative decision making (Badani *et al.*, 2015). To fully achieve such potential, it is critical to develop prediction algorithms with generalizable prediction performance on independent data, which are transferable to clinical use (Simon *et al.*, 2003). However, the presence of 'study effects', or heterogeneity across genomic studies, makes it challenging to develop generalizable prediction models. In particular, it has been established that cross-study validation performance of genomic classifiers is often inferior to internal cross-validation (Bernau *et al.*, 2014; Chang and Geman, 2015; Ma *et al.*, 2014), and that this gap cannot be entirely explained by the most easily identifiable sources of study heterogeneity (Zhang *et al.*, 2018b). Further research is needed to better understand and address the impact of heterogeneity on predictor performance.

In this study, we focus on a particular component of study effects, known as batch effects (Leek *et al.*, 2010), and aim to address its unwanted impact in binary classification problems. Batch effects are variation across batches of data that are unrelated to the biological question of interest. Strictly, batch effects refer to technical differences across experimental processing batches. However, in practice, batch effects are often used in a broader sense, with varied definitions for different contexts (Lazar *et al.*, 2013). It is common to use batch adjustment methods for integrating not only technical batches, but also datasets across experimental platforms or types of biological samples, for the same purpose of analysis (Bobak *et al.*, 2019; Butler *et al.*, 2018; Larsen *et al.*, 2014; Zhang *et al.*, 2018a). Existence of batch effects threatens the reproducibility of genomic findings (Kupfer *et al.*, 2012). And therefore, it is necessary to develop efficient methods to remove the undue influence of batch effects.

Many batch effect adjustment methods have been proposed for gene expression microarray (Benito *et al.*, 2004; Gagnon-Bartsch *et al.*, 2013; Gagnon-Bartsch and Speed, 2012; Johnson *et al.*, 2007; Leek and Storey, 2007) and sequencing data (Leek, 2014; Risso *et al.*, 2014). These methods share the general strategy to first merge all batches, estimate parameters representing differences in batch distributions, and then remove them from the data, resulting in a single adjusted dataset for downstream analysis. Here, we adopt a different perspective, and propose to address batch effects with ensemble learning. Contrary to the traditional batch effect adjustment methods, our proposed framework is based on the integration of predictions rather than the integration of data. This is a simpler task for prediction, as it operates in one dimension rather than many. Also, it is possible to reward predictors that show good performance across batches and thus altogether ignore, rather than trying to repair, features that are preferentially affected by batch effects.

Although ensemble learning is a well-established method, it has only recently been discussed in the context of training replicable predictors. Patil and Parmigiani (2018) found that ensembles of learners trained on multiple studies generate predictions with more replicable accuracy. In their framework, a cross-study learner (CSL) is specified by three choices: (i) a data subsetting strategy; (ii) a list of one or more single-study learners (SSLs), which can be any machine learning algorithm producing a prediction model using a single study; and (iii) a combination approach utilizing multiple prediction models to deliver a single prediction rule. In our case, we subset the data by batch, and use the same CSL with batches in place of studies. Guan *et al.* (2019) provide theoretical insights in the comparison between merging and ensembling in training learners from multiple studies, and conclude that although merging is better than ensembling when the studies are relatively homogeneous, ensembling yields better performing models at higher levels of study heterogeneity.

In this article, we explore using ensemble learning in the context of batch effect adjustment for the first time. We provide both realistic simulations and real data examples to demonstrate the utility of our ensembling framework, and compare it with traditional merging strategies for addressing batch effects.

## 2 Materials and methods

### 2.1 Addressing batch effects via ensemble learning

We structure the problem as follows: in a binary classification problem on genomic data, we have a training set for learning prediction models ($S_{\text{trn}}$), and another independent test set ($S_{\text{tst}}$) where predictions are to be made. Subjects in the training set are associated with a binary label indicating their phenotype. Examples of the phenotype could be disease status (e.g. cancer versus normal) or response to a treatment. In addition, expressions of genes for all individuals are profiled for use as predictors. Individuals in the test set also have measured gene expressions, and the goal is to train a model that can accurately predict the disease label for them based on their gene expression profiles. We assume that the training set is generated in $B$ batches $S^1_{\text{trn}}, S^2_{\text{trn}}, \dots, S^B_{\text{trn}}$, possibly due to practical or technical restrictions. We assume that each batch contains a sufficient number of samples to train a prediction model and that there are both additive and multiplicative batch effects (Johnson *et al.*, 2007), which cause differences in the mean and the variance of gene expressions across batches.

Consider a collection of $L$ learning algorithms to use for training. Multi-study learning begins by training each of the algorithms within each of the batches. This results in the collection $\hat{Y}^l_b(\mathbf{x})$, $l = 1, \dots, L$ and $b = 1, \dots B$, where $\hat{Y}^l_b(\mathbf{x})$ is the prediction function trained on batch $b$ with learning algorithm $l$. In the binary classification setting, $\hat{Y}$ are the probabilities of samples belonging to the positive class. The final cross-study learner's (CSL) prediction is calculated by a weighted average of predictions from each model, that is:

$$\hat{Y}(\mathbf{x}) = \sum_{l=1}^{L} \sum_{b=1}^{B} w_{lb} \hat{Y}^l_b(\mathbf{x}). \quad (1)$$

The performance of a CSL relies critically on the weights $w_{lb}$, as these have the function of rewarding elements of the ensemble which show stable predictive performance across batches. We explore five weighting strategies, which fall into three categories, as described in Patil and Parmigiani (2018). The first are sample size weights, which use scaled batch sizes as weights for models trained in the corresponding batches, and make no direct effort to reward robustness to batches via weights. The second are cross-study weights, for which we evaluate how well each learned model performs when applied to the other batches within the training set, and assign higher weights to models that have better prediction performances. The last category is stacking regression weights (Breiman, 1996), for which we use each model to make predictions of the training data, and estimate the weights as regression coefficients between stacked predictions of the training samples and their labels. The association coefficients are estimated using non-negative least squares. A more detailed description of the weighting strategies is available in the Supplementary Materials.

### 2.2 Data

We use a collection of 7 RNA-Seq and microarray studies targeting subjects infected with tuberculosis (TB) to apply and evaluate our ensemble learning method, and make comparisons with the traditional strategy of merging followed by batch adjustment ('merging'). Subjects involved in this collection of studies can be divided in three phenotypes based on their disease progression status: (i) latent infection (LTBI)/non-progressing, (ii) in 'progression', or those that will progress to disease in the near future and (iii) active TB disease. For simplicity and sample size considerations, we use two types of patients in each analysis to form a binary classification problem, and focus on different phenotypes for simulation and application of our methods. In simulations, we focus on predicting progressors against not progressors, as this separation yields the largest sample size in the training set. In real data, since there are no known batch effects within any single study, we aim to separate subjects with latent infection from those with active disease instead, so we have a sufficient number of studies in the collection. We merge studies for training while treating the differences between studies as 'batch effects'.

Table 1 summarizes information on the samples used in simulation studies and real data analysis. This collection of studies share a similar purpose: to develop genomic biomarkers for patients with different tuberculosis progression status. It is common in practice to consider them as 'batches' for the uniform purpose. However, these studies differ in both biological and technical aspects. For example, study A (Zak *et al.*, 2016) targets only adolescents between the ages of 12 and 18, and study C (Anderson *et al.*, 2014) measures only children under the age of 15. The remaining studies contain subjects in a much wider age spectrum, including both children and adults. In the collection, the geographical source of populations includes India (study D), the United States (study E) and different regions in Africa (studies A, B, C, F, G). Studies are also generated from various array and sequencing platforms (Table 1). We recommend caution when interpreting the results of analyses that consider studies as batches, as study differences may be a mixture of biological and artifactual differences.

**Table 1.** Summary statistics of the TB datasets used in simulation studies and real data applications

| | Data | Platform | | No. of subjects | No. of Pos (prevalence) | %age $\leq$ 18 |
|---|---|---|---|---|---|---|
| Simulations | A (Train) | RNA-Seq | Illumina HiSeq 2000 | 181 | 77 (42.5%) | 100% |
| | B (Test) | RNA-Seq | Illumina HiSeq 2000 | 399 | 95 (23.8%) | 26.3% |
| Application | A | RNA-Seq | Illumina HiSeq 2000 | 120 | 16 (13.3%) | 100% |
| | C | microarray | Illumina HumanHT-12 V4.0 expression beadchip | 70 | 20 (28.6%) | 100% |
| | D | RNA-Seq | Illumina NextSeq 500 | 44 | 25 (56.8%) | 2.3% |
| | E | microarray | Affymetrix Human Gene 1.1 ST Array | 70 | 35 (50%) | 0% |
| | F | microarray | Illumina HumanHT-12 V4.0 expression beadchip | 94 | 46 (48.9%) | 0% |
| | G | microarray | Illumina HumanHT-12 V4.0 expression beadchip | 86 | 51 (59.3%) | 0% |

*Note*: No. of subjects: total number of individuals in each study. No. of Pos: number of positive samples. They refer to progressors in simulation studies, and active patients in real data application. The negative samples are subjects with latent infections in both cases. Prevalence: percentage of positive samples in the study. Study A targets only adolescents between ages of 12 and 18. Study C measures only children under the age of 15. The remaining studies contain both children and adults. %age $\leq$ 18 is the percentage of samples of age 18 or less. Studies F and G are published together as two separate studies with different populations. References for the datasets are available in Supplementary Table S1.

## 2.3 Simulation for comparing merging and ensembling

We consider two datasets in simulation, A and B, which are collected from the African population, with the goal of predicting TB progression, as described in Table 1. We use study A for training, and study B for independent validation. For the training set only, we randomly assign individuals to disjoint subsets which will be simulated to be batches, and simulate differences in the moments of gene expression distributions across batches as described below. No batch effects are added to the validation set. We train predictors using both merging and ensembling on this dataset with simulated batch effects, then make predictions in the other independent study. We evaluate the two approaches using discrimination in the independent study.

### 2.3.1 Simulation of batch effects

We transformed the sequencing data into the logarithm of fragments per kilobase of transcript per million mapped reads (logFPKMs), and selected the top 1000 genes with the highest variances for building the classifiers. Then, we randomly took subsets of individuals from the training set to form 3 batches, each batch containing 10 non-progressors and 10 progressors. We then simulated batch effects across the 3 batches.

Our data generating model for batch effects is the linear model assumed in the ComBat batch adjustment method (Johnson *et al.*, 2007). Specifically, we estimate two components from the original training data: (i) the expression of gene $g$ among the negative samples, and (ii) the biological effect (i.e. the expression changes due to biological perturbations or conditions of interest). We then specify batch effect parameters affecting the mean ($\gamma_{\mathrm{gb}}$) and the variance ($\delta_{\mathrm{gb}}$) of expression in gene $g$ caused by batch $b$. As in Johnson *et al.* (2007), $\gamma_{\mathrm{gb}}$ and $\delta_{\mathrm{gb}}$ are randomly drawn from hyper-distributions

$$\gamma_{gb} \sim \mathcal{N}(\eta_b, \tau_b^2), \quad \delta_{gb} \sim \mathrm{InvGamma}(\lambda_b, \theta_b). \tag{2}$$

ComBat assumes an additive batch effect for the mean, and a multiplicative batch effect for the variance. To set the hyper-parameters, we first specify a value to represent the severity of batch effects, as reported in columns $sev_{\mathrm{mean}}$ and $sev_{\mathrm{var}}$ in Supplementary Table S2. We selected three severity levels ($sev_{\mathrm{mean}} \in \{0, 3, 5\}$) for batch effect on the mean, and five levels ($sev_{\mathrm{var}} \in \{1, 2, 3, 4, 5\}$) for batch effect on variance. Given a severity level for batch effects, we fixed values for $\tau_{\mathrm{b}}$s and $\theta_{\mathrm{b}}$s, so that the variance of $\gamma_{\mathrm{gb}}$ and $\delta_{\mathrm{gb}}$ over genes are 0.01. We varied the mean of these two parameters, so that the hyper mean $\eta_{\mathrm{b}}$ is $(-sev_{\mathrm{mean}}, 0, +sev_{\mathrm{mean}})$, and the hyper variance $\lambda_{\mathrm{b}}$ is $(1/sev_{\mathrm{var}}, \frac{1/sev_{\mathrm{var}}+sev_{\mathrm{var}}}{2}, sev_{\mathrm{var}})$ for the three batches. The parameters are then added or multiplied to the expression mean and variance of the original study. The characteristics of simulated batches are also summarized in Supplementary Table S2. Supplementary Figure S2 shows an example training set where we simulated three batches with both mean and variance differences. The magnitudes of batch effects are selected based on what we previously observed in real data (Zhang *et al.*, 2018a, 2020).

### 2.3.2 Comparing ensemble learning with merging

We then use the dataset with simulated batch effects to train classifiers for predicting patient phenotypes. We perform the ensemble learning strategy as described above. For the merging strategy, we pooled the three batches together, and applied ComBat to remove batch effects. We then used the whole adjusted data to train a single model, and make one set of predictions on the independent test set. We trained learners LASSO (Tibshirani, 1996), Random Forest (RF, Breiman, 2001) and Support Vector Machines (SVM, Cortes and Vapnik, 1995), after performing the two batch adjustment strategies. In ensemble learning, we evaluated aggregating predictions both from a single learning algorithm ($L = 1$), and across all algorithms ($L = 3$). The accuracy of performance was measured by the area under ROC curve (AUC). We repeated batch correction and predictions to generate 100 discrimination scores, and compared them between the two strategies.

## 2.4 Applying ensemble learning in real data batch effects

To demonstrate a realistic application setting for our ensemble learning method, we took 6 TB studies as summarized in Table 1. Our goal in this context will be to distinguish between active and latent TB. We iteratively treat each of these studies as the independent test set. The remaining studies are used as batches forming the training set.

The original data contains more than 15 000 genes, resulting in too unfavorable a situation to support method comparisons. For example, LASSO was not able to get a prediction AUC above 0.5 on the independent samples. We thus prefiltered genes to select a subset of the 1000 most highly variable, and used the same subsets of genes for both ensembling and merging.

We trained three learning algorithms: LASSO, RF and SVM, and integrated predictions both from each single learning algorithm, and from all three learning algorithms in the ensemble framework. The remaining methods for batch effect adjustment, predictions and model evaluations are the same as those for simulation studies. We performed 100 bootstrap replicates on the test set, to obtain a confidence interval for model performance scores. Performance is evaluated using both AUC and mean cross-entropy loss, defined as

$$mxe = -\frac{1}{N}\sum_{i=1}^{N}(y_i \log(p_i) + (1 - y_i)\log(1 - p_i)),$$

where $N$ is the number of samples, $y_i$ is the true label for sample $i$, (0 if latent TB, 1 if active TB) and $p_i$ is the predicted probability of sample $i$ having active TB. We show the cross-entropy loss in the main paper, and include AUC results in Supplementary Materials. Compared to AUC, cross-entropy loss is sensitive to changes in predicted probabilities which may not affect the overall ranking of prediction. Cross-entropy loss is also a proper scoring rule (Gneiting and Raftery, 2007), unlike the AUC.

## 3 Results

### 3.1 Impact of mean and variance batch effects on discrimination of predictions

Figure 1 summarizes results over 100 simulated datasets, representative of the patterns we observe across simulation studies. We used a Random Forests learner for both merging and ensembling. Random Forests achieve a 0.685 AUC on the test set when using the original training study without simulated batch effects. When adding batch effects to the data, we observed drops in discrimination in the test set. Mean and variance batch effects affect prediction performance in different ways. Model discrimination is not strongly affected by batch if batch effect only affects the variance. A sufficient size of the mean differences across batches in the training set is necessary to cause a drop in prediction accuracy. On the other hand, when batch affects the mean, an increase in variance differences across batches will lead to a further drop in discrimination.

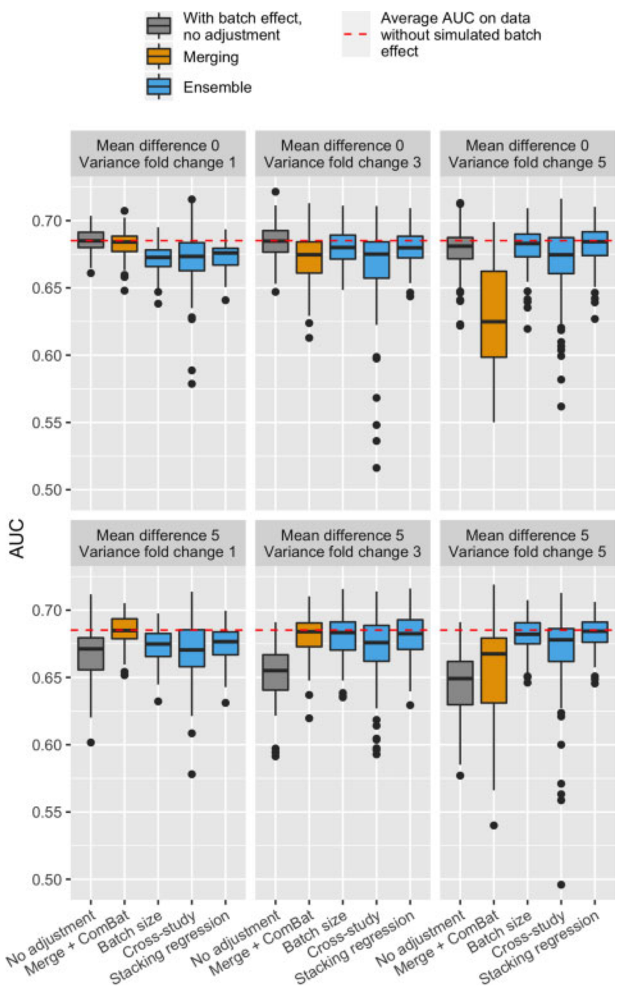

**Fig. 1.** Comparison between ensembling and merging when using Random Forests. Three out of our five choices of ensembling weights are displayed: batch size weights, cross-study weights and stacking regression weights (see Section 2 for details)

### 3.2 Ensemble learning achieves more stable discrimination than merging at high severity of batch effects

Supplementary Table S2 shows the levels of batch differences we created in simulation studies, corresponding to the results in Figure 1. At a low level of batch effects, with no mean difference and a variance fold change smaller than three, the merging method yields better discrimination. However, we observed a turning point in the severity of batch effect, after which ensemble learning starts to achieve higher discrimination. This turning point is characterized by general stability or small improvements in the performance of the ensembling approaches and large drops in the performance of the unadjusted and batch-adjusted merged data approaches. We considered ensembling both using a single learning algorithm, and using all three learning algorithms together. We observed similar performance gains across levels of batch effects. Rather than comparing the absolute performance of algorithms, our observation highlights the robustness of ensembling compared to merging across increasing degree of batch effects, a common pattern shared by LASSO, SVM and RF.

The turning point in the magnitude of batch effects differs by the selected learning algorithms. Supplementary Figures S3, S4 and S5 show the simulation results of training with all algorithms. When training with SVM, for example, we see that discrimination from ensemble learning is already comparable with that using the merging method in data with no mean batch effect and a variance fold change of two, though at this level, merging still out-performs ensembling when training with Random Forests. Also, at high level of batch discrepancies, stacking regression weights yield better prediction results than the sample-size weights when building ensemble with SVM alone, though they are more comparable when using Random Forests.

We also observed that ensembling across different learning algorithms does not necessarily improve the final prediction compared to ensembling with a single algorithm. We see in Figure 1 that ensemble learning across algorithms generates worse performance than using only Random Forests. The optimal weighting approach also depends on the learners involved in the ensemble. When using Random Forests only, the sample-size weights and the stacked regression weights generate better accuracy than the cross-study weights. But the latter is better in integrating across all algorithms. Note that despite the difference in rankings of the three types of weights, all three ensemble weighting methods out-perform merging and batch correction with ComBat at high level of batch differences.

Finally, we repeated the simulations with a larger sample size, and larger number of batches. To increase batch size, we took 3 subsets as batches, each containing 20 non-progressors and 20 progressors, in contrast to our previous results which use 10 individuals per condition per batch. For a larger number of batches, we took five subsets of subjects as simulated batches. We observed consistent patterns in both situations, as shown in Supplementary Figures S6 and S7.

### 3.3 Predicting tuberculosis disease phenotype

We now present a classification case study using real data with batch effects. Specifically, we selected six studies designed to discover biomarkers of TB disease (versus control). We iteratively treat one as the test set, and use the others for training. The study label serves as the batch. We applied both ensembling and merging to address batch effects, and trained three types of learning algorithms: LASSO, RF and SVM. Ensemble predictions are aggregated using each learning algorithm, and from all algorithms. That is three $L = 1$ experiments, each with one of the algorithms, and another $L = 3$

experiment ensembling across all algorithms. To obtain a bootstrap confidence interval of model performance, we generated 100 bootstrap samples from the test set. Since study effects are often considered to be more severe and complicated than technical batch effects, this application example represents a high level of batch differences.

The cross-entropy loss and the weights assigned to learners trained from each training batch, using only random forest learners, are shown in Supplementary Figure S8. Whenever study A is included as a batch in training, the stacking weights are dominated by weights assigned to the learner from study A. Meanwhile, cross-study weights consider it to be the worst-performing when generalizing to the other batches in the training set. In addition, though such patterns in the assigned weights are consistent across test sets, we observe different rankings of merging and the three ensembling methods in mean cross-entropy loss when different studies are used as the test set. Cross-study weights achieve the lowest average loss among all methods on studies D, F and G, and among the three ensemble methods on study E, while stacking regression weights yield the worst loss. This ranking is reversed when A or C is used as the test set. These observations of model performance are consistent when aggregating predictions from all three learning methods, as shown in Figure 2. The corresponding results using AUC as the evaluation metric are shown in Supplementary Figure S9. We also explored the options of choosing the top 10, 100 and 10 000 genes with the highest marginal variances (Supplementary Fig. S10), instead of the 1000 used for the main analysis. These choices do affect the absolute performance of models, but the general conclusions about the relative merits of ensembling and merging strategies remain valid.

To investigate the reasons behind these observations, we have noticed and mentioned that the six studies include populations in different age ranges. The two studies which generate different ranking of methods, A and C, only include children or adolescents. It is well-established that young age is a risk factor for progression from TB infection to active disease (Narasimhan *et al.*, 2013). Tuberculosis in childhood and adulthood is also different in clinical features and pathogenesis (Alcaïs *et al.*, 2005). When each of the six studies are used as the test set, we merged the other five studies after z-score standardization within each study, and selected 1000 genes with the highest marginal variance. Among these 1000 genes, we used limma (Smyth, 2005) to perform differential expression in each study and identified the 50 most significant genes ranked by FDR-adjusted *P*-values. We then took the overlapping genes across test studies, resulting in 11 up-regulated and 16 down-regulated genes that are common across studies. We measured the average expression of these up- and down-regulated genes in each study, as summarized in Supplementary Table S3, and visualized their expression distribution in Supplementary Figure S11. This analysis shows that the top differentially expressed genes are not those with the clearest biological signal in studies A and C. These results suggest that the difference in age may be confounded with batch, which may explain the inconsistent observations in the six-study analysis. For example, when training in the other batches and validating in A or C, the training data consists of a mixture of children and adults, in which we are not able to develop a predictor that generalizes well specifically for children/adolescents.

We then took the four studies with both children and adults, namely D, E, F and G, and re-applied both ensembling and merging strategies. In this case, the four studies are more representative of what are usually considered ‘batch effects’, as they serve the same purpose of developing TB biomarkers across children and adults. The bottom row in Figure 2 shows the mean cross-entropy loss of predictions, where ensemble predictions are aggregated across all three algorithms. We observed that the average performance of ensemble learning approaches is consistently better than that of merging in all test studies. When training with Random Forests as the sole learner, the mean cross-entropy loss and the weights assigned to each learner are shown in Supplementary Figure S12.

Due to the relatively small sample sizes, we also observed a high variance in model performance. Therefore, we also compared ensembling against merging within each bootstrap sample in the four-study analysis. The proportions of bootstrap samples where each method achieves the lowest cross-entropy loss are summarized and annotated in Figure 2, which shows that in the four-study analysis, in almost all situations, addressing batch effects via ensemble learning yields more robustly performing models with respect to discrimination. These results are broadly consistent with our observations in simulations that when there are severe differences across batches, ensemble learning is a more robust strategy for addressing the impact of batch effects on prediction.

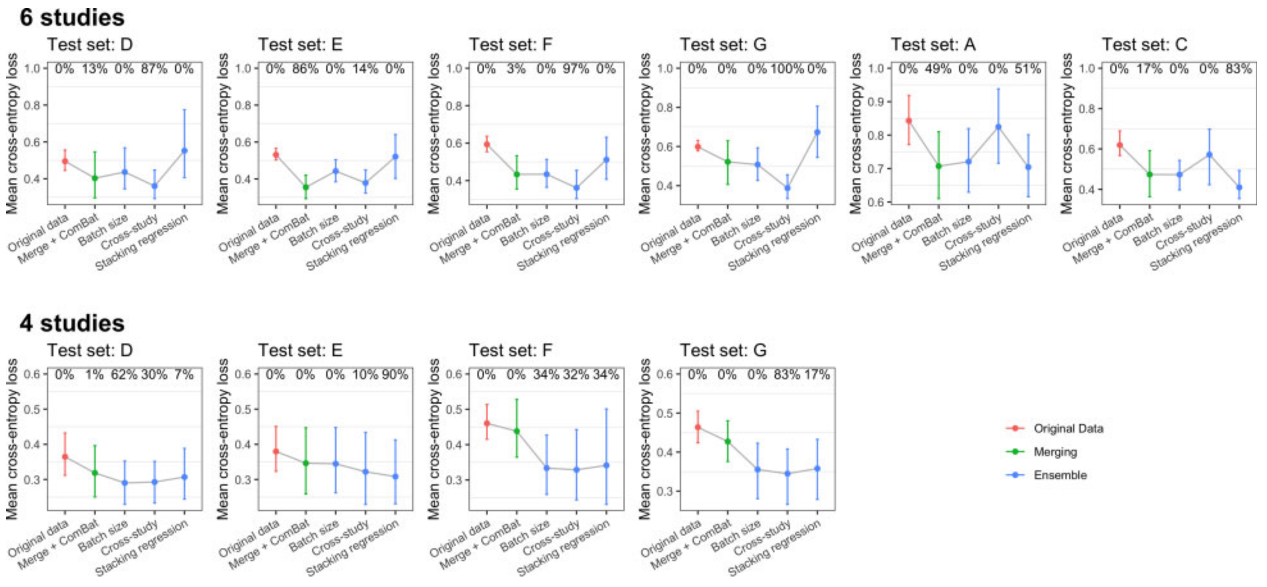

**Fig. 2.** Application of ensemble learning to predicting active TB against latent infection. We iteratively selected one of the studies in Table 1 as the independent test study. The remaining studies are viewed as ‘batches’ in the training set. We trained LASSO, Random Forest and SVM, then aggregated predictions from all three algorithms to construct the ensemble. The figure shows average prediction performance over 100 bootstrap samples of the test data, with error bars showing 95% confidence intervals. Above the bars we note the percentage of bootstrap experiments where each method achieves the lowest mean cross-entropy loss. When the four homogeneous studies are used, the average performance using the three ensemble strategies are better than the merging strategy, which is consistent with observations from the simulation study at high severity of batch effects. Different ensemble methods can be the best in a different test set (the optimal study—ensemble combination: D—batch-size weights, E—stacking regression weights, G—cross-study weights). For study F, the three ensemble methods are roughly equal, each wins 33% of the time

Finally, the changes in results from using six studies compared to those using four are largely due to the fact that study A contains a different population from the other studies, is highly imbalanced, and is also much larger in size than the other studies. Having only 16 active TB samples among 120 subjects in study A makes it challenging to develop a generalizable predictor from study A. However, this predictor is assigned more weight by the stacking regression method due to the large proportion of the total of the observations across all studies, which are aggregated for the stacking regression. To verify this conjecture, we upsampled the smallest study (D) in the four-study collection to be five times its original size, and repeated the experiments. In another set of experiments we downsampled study A to have 16 active and 16 latent patients. Comparing the upsampling (Supplementary Fig. S13) and the downsampling (Supplementary Fig. S14) results with the weight assignment in the main results, we see that stacking weights assigned to study D indeed increased after upsampling, and the weights for study A decreased after downsampling. This suggests that the stacking method can indeed be sensitive to the size of the studies.

## 4 Discussion and conclusions

We proposed a novel perspective for addressing batch effects when developing genomic classifiers. Our proposal is to use multi-study ensemble learning, treating batches as separate studies. We provided both realistic simulations and real data applications to compare ensembling with the traditional approach, which is to analyze all batches together to remove batch effects, and use the adjusted data for prediction. We observed in both simulations and real data that, though merging is able to generate better performing models when batch effects are modest, ensemble learning achieves more robust discrimination in independent validation across different levels of severity of batch effects. Our observations suggest that ensembling is more likely to provide robust prediction performance when the heterogeneity level is unknown, while merging plus batch correction comes with the risk of substantial performance loss. These findings are consistent with those described in Guan *et al.* (2019), who provides theoretical insights into the comparison between merging and ensembling in training cross-study learners. We explored different training algorithms, different batch sizes and number of batches, and observed consistent patterns of such transition.

The philosophy behind the standard approach of merging and batch adjustment is to remove undesired batch-associated variation from as many of the genomic features as feasible, and then use the 'cleaned' data in classification as though the batch effects never existed. This has been the standard in the literature and can be quite successful (Engchuan *et al.*, 2016; Luo *et al.*, 2010; Riester *et al.*, 2014). Multi-study ensemble learning provides a different perspective: ensemble weights reward prediction functions that, while trained in one batch, continue to predict well in other batches. These are likely to avoid using features affected by batch effect, in contrast to cleaning them and potentially losing useful biological signal.

In our application with four studies, we observed patterns broadly consistent with those seen in the simulations. In contrast, in the analysis with six studies, the advantages and disadvantages of ensembling and merging are less clear, because studies A and C contain radically different populations compared to the others. Furthermore, study A is large in size, causing the stacking regression strategy to assign a high weight to the learner trained from A whenever it is included as a training batch. Here one batch is large and confounded with another variable which strongly influences the prediction. Then, ensembling does not provide a model with generalizable performance. We therefore recommend to thoroughly evaluate the available information in the data, and identify any confounders for prediction. The studies or batches of data to be jointly used should contain the samples and study designs suitable for addressing the same biological question.

In simulations, we observed that ensembling generates more robust performance compared to merging, especially at high levels of batch effects. The specific level where ensembling begins to outperform merging was different for each of the three learning algorithms we explored. A natural explanation for this observation is that different learning algorithms may be affected by batch effects to different extents. The relationship between the learning algorithms used and the merging versus ensembling choice is complex and worthy of further methodological study. Ensembling over multiple learners provides a logical strategy when the implications of using specific learners are not well understood, and would deserve further study.

We compared five kinds of weighting strategies to integrate the predictions. The relative performance of different weighting strategies also depends on the specific dataset and algorithm. Additionally, there could be ways to further improve the ensemble performance by developing other weighting methods that are not considered in this study. All ensembles considered are very small because the number of batches *B* and the number or learners *L* are both small. Recently, Ramchandran *et al.* (2019) compared an ensembling strategy based on RF to one based on cross-study weighting of the component trees in the RF, showing improvements from direct reweighting of trees. This method may also prove effective in our context, as individual trees are more parsimonious than the whole forest, and may more effectively avoid features affected by batches. Lastly, we explored, but did not highlight, combining both the batch-correction and ensembling strategies (Supplementary Fig. S15). In this case, data is first merged and batch-corrected with ComBat, then ensemble learning is performed by separating the adjusted data by the original batches. We observed that combining both strategies did not improve the prediction performance over either strategy alone. We conjecture that batch adjustment may be removing some of the biologically relevant study heterogeneity which the CSLs take advantage of to improve generalizability.

Our study has several limitations. First, the ensemble learning approach requires that each batch contains sufficient samples to train a prediction function. This assumption may limit our approach to sufficiently large datasets. However, most if not all batch effect adjustment strategies require a reasonable number of samples in each batch to accurately and robustly estimate batch effects. Having limited number of samples in a batch will negatively affect not only our proposed methods, but also traditional methods based on merging. We speculate that methods like ComBat might, however, be able to effectively operate with smaller batches than ensembling. Still, in the context of prediction, it is usually challenging to train models on small datasets.

We focused on using ComBat for batch effect adjustment after merging. ComBat is not the only option, but remains one of the most popular batch effect adjustment methods, especially in the case where batch effect sources are known. Our simulations of batch effects are all based on the generative model of ComBat, which, while plausible, is one among many possible models. Using the same batch effect model for data generation and analysis provides a lower bound to the effectiveness of our proposal, as any other data generating approach would be less favorable to ComBat than the one we used. In data drawn from other generating mechanisms, the advantages of ensembling should be more pronounced, and may set in at lower levels of batch effects.

We evaluated our approach in binary prediction. While simple, binary classification remains one of the most commonly encountered tasks in clinical applications. We provided a realistic application example using the tuberculosis studies. However, our observations and conclusions may not extrapolate to multi-class or time-to-event scenarios. Patil and Parmigiani (2018) discussed an ensemble learning strategy for predicting continuous outcomes. Further research is needed to extend this approach to multi-class prediction.

Our tuberculosis application illustrates a situation with relatively small batch sizes in several batches. It would be interesting to further explore our approach on data with larger batch sizes, or a larger number of batches. Larger batches facilitate both the estimation of batch effects and the training of batch-specific predictors. A larger number of batches facilitates learning about the higher-level distributions in ComBat, and would afford ensembling a better opportunity to find stable signal across a larger number of batches, a strength of the method that is not highlighted here.

Related, treating studies as batches mimics a high level of batch differences, for the discrepancy across the ‘batches’ in this setting includes both biological and technical differences. Thus we include more sources of heterogeneity than normally considered as a batch effect. Specifically, a batch effect is generally defined as variations originated from technical differences across repeated experiments performed on the same platform, such as differences in lab environment, protocols or reagents. In our application example, however, each batch targets a different population, which means there likely exists additional genetic variations across the groups. Also, the TB studies are generated on different platforms. Differences across these technologies are also beyond what is typically considered as batch effects. Despite that, using batch correction software for study integration is common in practice, in which studies are often considered as ‘batches’ for performing a uniform biological analysis. We believe this example is representative of common practice and helpful to illustrate our methodology, and our observations offer valuable practical guidelines in addressing batch effects in genomic classifier development. We have shown that when batch differences are small, the batch adjustment software may be sufficient for mitigating the negative impact on prediction performance. Ensembling provides a more robust and stable option, which is useful when batch effects are strong or unknown.

## Funding

This work was supported by the grants Division of Mathematical Sciences, National Science Foundation (NSF-DMS) 1810829, The National Cancer Institute, National Institutes of Health (NIH-NCI) 4P30CA006516-51 (G.P.) and 5R01GM127430-02 (W.E.J.).

*Conflict of Interest*: none declared.

## References

Alcaïs,A. *et al.* (2005) Tuberculosis in children and adults: two distinct genetic diseases. *J. Exp. Med.*, **202**, 1617–1621.

Anderson,S.T. *et al.* (2014) Diagnosis of childhood tuberculosis and host RNA expression in Africa. *N. Engl. J. Med.*, **370**, 1712–1723.

Badani,K.K. *et al.* (2015) Effect of a genomic classifier test on clinical practice decisions for patients with high-risk prostate cancer after surgery. *BJU Int.*, **115**, 419–429.

Benito,M. *et al.* (2004) Adjustment of systematic microarray data biases. *Bioinformatics*, **20**, 105–114.

Bernau,C. *et al.* (2014) Cross-study validation for the assessment of prediction algorithms. *Bioinformatics*, **30**, i105–i112.

Bobak,C.A. *et al.* (2019) Comparison of common machine learning models for classification of tuberculosis using transcriptional biomarkers from integrated datasets. *Appl. Soft Comput.*, **74**, 264–273.

Breiman,L. (1996) Stacked regressions. *Mach. Learn.*, **24**, 49–64.

Breiman,L. (2001) Random forests. *Mach. Learn.*, **45**, 5–32.

Butler,A. *et al.* (2018) Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nat. Biotechnol.*, **36**, 411–420.

Chang,L.-B. and Geman,D. (2015) Tracking cross-validated estimates of prediction error as studies accumulate. *J. Am. Stat. Assoc.*, **110**, 1239–1247.

Cortes,C. and Vapnik,V. (1995) Support-vector networks. *Mach. Learn.*, **20**, 273–297.

Engchuan,W. *et al.* (2016) Handling batch effects on cross-platform classification of microarray data. *Int. J. Adv. Intell. Paradigms*, **8**, 59–76.

Gagnon-Bartsch,J.A. and Speed,T.P. (2012) Using control genes to correct for unwanted variation in microarray data. *Biostatistics*, **13**, 539–552.

Gagnon-Bartsch,J.A. *et al.* (2013) Removing unwanted variation from high dimensional data with negative controls. *Tech reports.* Dep Stat Univ California, Berkeley, pp. 1–112.

Gneiting,T. and Raftery,A.E. (2007) Strictly proper scoring rules, prediction, and estimation. *J. Am. Stat. Assoc.*, **102**, 359–378.

Golub,T.R. *et al.* (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, **286**, 531–537.

Guan,Z. *et al.* (2019) Merging versus ensembling in multi-study machine learning: theoretical insight from random effects. *arXiv preprint arXiv : 1905.07382*.

Johnson,W.E. *et al.* (2007) Adjusting batch effects in microarray expression data using empirical bayes methods. *Biostatistics*, **8**, 118–127.

Kupfer,P. *et al.* (2012) Batch correction of microarray data substantially improves the identification of genes differentially expressed in rheumatoid arthritis and osteoarthritis. *BMC Med. Genomics*, **5**, 23.

Larsen,M.J. *et al.* (2014) Microarray-based rna profiling of breast cancer: batch effect removal improves cross-platform consistency. *BioMed. Res. Int.*, **2014**, 1–11.

Lazar,C. *et al.* (2013) Batch effect removal methods for microarray gene expression data integration: a survey. *Brief. Bioinf.*, **14**, 469–490.

Leek,J.T. (2014) Svaseq: removing batch effects and other unwanted noise from sequencing data. *Nucleic Acids Res.*, **42**, e161–e161.

Leek,J.T. and Storey,J.D. (2007) Capturing heterogeneity in gene expression studies by surrogate variable analysis. *PLoS Genet.*, **3**, e161.

Leek,J.T. *et al.* (2010) Tackling the widespread and critical impact of batch effects in high-throughput data. *Nat. Rev. Genet.*, **11**, 733–739.

Leong,S. *et al.* (2018) Existing blood transcriptional classifiers accurately discriminate active tuberculosis from latent infection in individuals from south india. *Tuberculosis*, **109**, 41–51.

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

Ma,S. *et al.* (2014) Measuring the effect of inter-study variability on estimating prediction error. *PLoS One*, **9**, e110840.

Narasimhan,P. *et al.* (2013) Risk factors for tuberculosis. *Pulmonary Med.*, **2013**, 1–11.

Papaemmanuil,E. *et al.* (2016) Genomic classification and prognosis in acute myeloid leukemia. *N. Engl. J. Med.*, **374**, 2209–2221.

Patil,P. and Parmigiani,G. (2018) Training replicable predictors in multiple studies. *Proc. Natl. Acad. Sci. USA*, **115**, 2578–2583.

Ramchandran,M. *et al.* (2019) Tree-weighting for multi-study ensemble learners. *bioRxiv*, **33**, 451–462.

Riester,M. *et al.* (2014) Risk prediction for late-stage ovarian cancer by meta-analysis of 1525 patient samples. *JNCI J. Natl. Cancer Inst.*, **106**, dju048.

Risso,D. *et al.* (2014) Normalization of RNA-seq data using factor analysis of control genes or samples. Nature Biotechnology, **32**, 896–902. 10.1038/nbt.2931

Seib,K.L. *et al.* (2009) The key role of genomics in modern vaccine and drug design for emerging infectious diseases. *PLoS Genet.*, **5**, e1000612.

Silvestri,G.A. *et al.* (2015) A bronchial genomic classifier for the diagnostic evaluation of lung cancer. *N. Engl. J. Med.*, **373**, 243–251.

Simon,R. *et al.* (2003) Pitfalls in the use of DNA microarray data for diagnostic and prognostic classification. *J. Natl. Cancer Inst.*, **95**, 14–18.

Smyth,G.K. (2005) Limma: linear models for microarray data. In: *Bioinformatics and Computational Biology Solutions Using R and Bioconductor*. Springer, New York, NY, pp. 397–420.

Tibshirani,R. (1996) Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B (Methodological)*, **58**, 267–288.

Zak,D.E. *et al.* (2016) A blood RNA signature for tuberculosis disease risk: a prospective cohort study. *Lancet*, **387**, 2312–2322.

Zhang,Y. *et al.* (2018a) Alternative empirical bayes models for adjusting for batch effects in genomic studies. *BMC Bioinformatics*, **19**, 262.

Zhang,Y. *et al.* (2018b) The impact of different sources of heterogeneity on loss of accuracy from genomic prediction models. *Biostatistics (Oxford, England)*, **21**, 253–268.

Zhang,Y. *et al.* (2020) Combat-seq: batch effect adjustment for rna-seq count data. NAR Genomics and Bioinformatics, 2, Iqaa078. 10.1093/nargab/lqaa078.