1 Introduction

Microarrays are the most mature and widely used high-throughput technology (Allison et al., 2006; Eisen et al., 1998). Sophisticated preprocessing and normalization techniques have been developed, and are widely used, to clean data from microarrays before analysis is performed (Allison et al., 2006). However, most of these preprocessing techniques and significance analyses do not take into account non-biological variables in the design of high-throughput studies (Mecham et al., 2010). Batch effects - the collective set of unmeasured non-biological variables associated with the batch in which a given array is run - have been identified as a major problem in the analysis of data from these studies (Leek et al., 2010). Specifically, it has been shown that ignoring non-biological artifacts like batch in the analysis of high-throughput data can result in misleading or incorrect results (Akey et al., 2007; Baggerly et al., 2004).

Although batch effects are now recognized as a major problem in the analysis of high-throughput data, until now there has been no thorough examination of the impact of batch effects on building genomic predictors. Batch could negatively impact prediction by obfuscating and washing out any predictive power of useful biological variations between certain outcomes. Furthermore, prediction accuracy could be erroneously overstated if batch and outcome are highly correlated, and batch proves to be easily predicted. In this scenario, prediction models would appear highly accurate, even under cross-validation in one study. However, the out-of-sample performance of these predictors would be considerably worse. Here we investigate the role of batch effects on building and evaluating predictors based on genomic data. To investigate the relationship between batch and prediction, we collected data from two publicly available Affymetrix gene expression studies. For each study, we collected information on an outcome for prediction and defined batches based on the date the microarrays were processed.

Using these data we first evaluated the relative impact of different prediction algorithms and normalization techniques on prediction. We examined Top-Scoring Pairs (TSP) - a popular rank-based prediction algorithm (Geman et al., 2004), and Prediction Analysis of Microarrays - a popular continuous prediction algorithm (Tibshirani et al., 2002). As a benchmark, we also examined the impact on the least angle regression implementation of the least absolute shrinkage and selection operator (Lasso) (Tibshirani, 1996; Efron et al., 2004). We examined Robust Multi-Array Average (RMA) - a multi-sample preprocessing technique (Irizarry et al., 2003), and Frozen Robust Multi-Array Analysis (fRMA) - a single-sample preprocessing technique (McCall et al., 2010). RMA borrows information across samples in a single study in order to normalize data. fRMA borrows information from historical studies, allowing for it to be applied to single microarrays at a time. This

feature of fRMA could be especially useful in cases where samples are examined individually, rather than in batches.

We used a cross-validation design to observe the above preprocessing methods and predictive algorithms in three different scenarios: (1) building and testing predictors on data collected within a single batch, (2) building predictors on one batch and testing predictors on a different batch, and (3) building and testing predictors on data collected from a mixture of batches. For all combinations of preprocessing and predictor building algorithms, prediction accuracy was worse when predicting across batches. However, on average batch did not have a substantial impact on prediction accuracy when the batch variable and the outcome variable were not highly correlated.

There are currently various methods that have been developed in order to correct for batch effects in the context of significance analysis, including SVA (Leek and Storey, 2007) and Combat (Johnson et al., 2007). However, these rely on populations of microarrays to estimate the effect of batch on individual measurements. Since the goal of our analysis is, in part, to determine methods that will work on studies with perfect confounding of batch and outcome, or on individual samples, we do not further consider these methods. Similarly, a previous study has examined the performance of a range of predictive algorithms after batch-affect removal using basic standardization and mean-removal methods (Luo et al., 2010). Our study differs because we compare pre-processing methods designed for multi-sample (RMA) and single-sample (fRMA) analysis. Using simulated study designs, we also consider the importance of the correlation between batch effects and outcome variables, which was not considered in Luo et al. (2010). Finally, we consider the potential for removing batch effects using sequence-based models. Although our results are negative, we believe they are an important contribution to the growing body of knowledge about how to model batch effects.

Since batch effects were only recently identified as playing a major role in the analysis of genomic data, much of the archived data in databases like the Gene Expression Omnibus (GEO) (Edgar et al., 2002) or ArrayExpress (Parkinson et al., 2009) come from studies where batch and outcome may be highly correlated. There continues to be strong interest in developing and validating predictors using these data. We designed a second cross-validation scheme to evaluate the impact of batch effects on prediction when batch and outcome were perfectly correlated. Using this cross-validation technique we showed that when batch and outcome are highly correlated: (1) there is a much stronger adverse effect of batch on prediction, (2) if the probes most highly associated with batch are known, removing them from the analysis improves prediction.

Taken together our results suggest that a critical component of building accurate genomic predictors is to develop training datasets where batch and other non-

biological variables are not highly correlated with the outcome of interest. Furthermore, our results suggest that using normalization techniques specifically designed to handle one sample at a time may lead to more accurate predictions. We have also shown that it may still be possible to build and evaluate predictors in data where batch and outcome are highly correlated, if the batch-affected probes are known in advance. Our results suggest that a fruitful avenue for future research may be to develop methods to identify and adjust or remove batch-affected probes when building genomic predictors.

2 Microarray case studies

We evaluated the role of batch on prediction using a set of simulated study designs to mimic a typical real-life development and application of a genomic predictor. The simulated designs are comprised of actual measured gene expression levels from two studies performed using the hgu133a Affymetrix platform. We downloaded the raw intensity measurements and phenotype data from the Gene Expression Omnibus (GEO) (Edgar et al., 2002). We determined batch using the date on which the array was processed, which is embedded in the raw data files. While date has been shown to be a good proxy for batch (Leek et al., 2010), it is possible that there is further unwanted variation within the batches defined by date. Therefore, our results may be conservative. In our simulated study designs, we varied the type of preprocessing method, the composition of the training and test sets, and the predictive algorithms.

2.1 Case Study 1 - Wang et al. (2005)

Our first dataset is from a large study of relapse-free survival in breast cancer samples (Wang et al. 2005; Carroll et al. 2006, accession GSE2034). The data set consists of 286 lymph-node negative samples with 180 patients relapse free after 5 years and 106 patients with a distant metastasis. Distant metastasis is a subtle phenotype and extremely challenging to predict, even without accounting for potential batch effects. So for the purpose of illustration we chose a more easily predicted phenotype, estrogen receptor (ER) status, as the outcome for our study. There were 209 ER+ patients and 77 ER- patients in the study.

Information about batches was not recorded in the metadata available for this study from GEO. However, the processing date of each microarray is included in the raw intensity Affymetrix CEL files. We extracted the sample processing dates from the CEL files and assigned samples into three batches based on clusters of these processing dates (Figure 1a). Batch A had 102 arrays (68 ER+, 34 ER-).

Batch B had 87 arrays (64 ER+, 23 ER-). Batch C had 97 arrays (77 ER+, 20 ER-). For the analyses presented, we use only the data from batches A and B. We chose these batches because the distribution of outcomes is more comparable in these two batches, and the span of dates included in each batch is roughly the same.

2.2 Case Study 2 - Minn et al. (2005)

The second case study is based on a medium-sized microarray study of breast cancer (Minn et al. 2005, accession GSE2603). We again used ER status as the outcome for illustrative purposes. There were 57 ER+ and 42 ER- patients (we excluded cell line data). We again assigned samples into four batches based on clusters of processing dates (Figure 1b). Batch A had 34 arrays with outcome information (15 ER+, 19 ER-). Batch B had 38 arrays with outcome information (23 ER+, 15 ER-). Batch C had 27 arrays with outcome information (19 ER+, 8 ER-). For the analyses presented, we use only the data from batches A and B. We chose these batches because both the span of the dates and the distribution of outcomes in these two batches is more comparable.

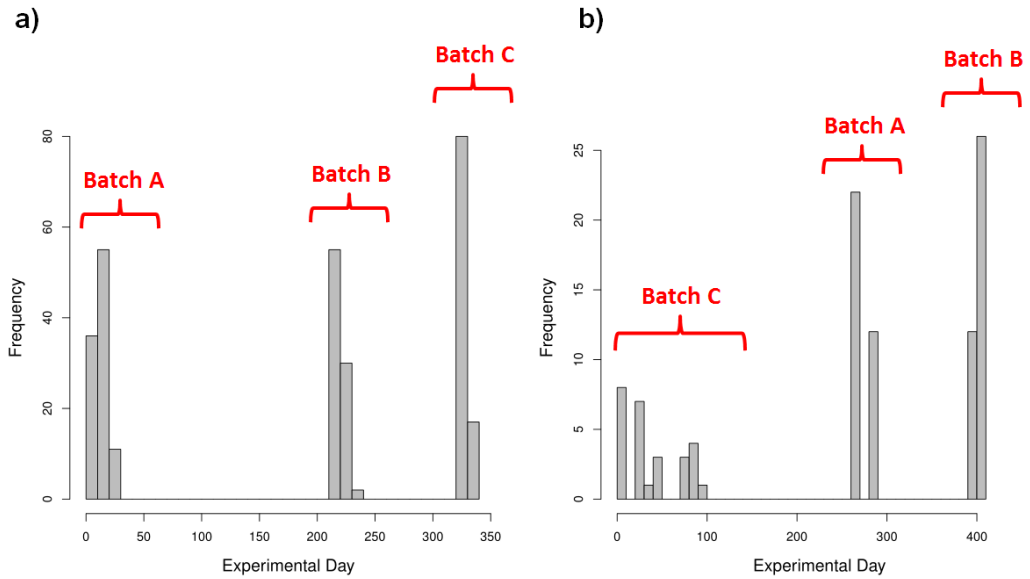


Figure 1: **Assignment of batch by date of microarray studies.** We assigned batches as indicated, based on the histogram of array dates for (a) Wang et al. (2005) and (b) Minn et al. (2005) data.

3 Simulated Study Designs

Genomic predictors are typically built on data from a single study and cross-validation accuracy is reported. The data within a single study are typically generated by the same personnel, using the same facilities, and under similar conditions. A second step in developing a genomic predictor is to evaluate the prediction accuracy on an independent validation set. In many publications, the validation samples are collected by the same group and using the same technology as the original samples. In clinical applications, each sample might be collected by different individuals using different technology and facilities. We created our simulated design to mimic this clinical application.

We performed a leave-half-out cross-validation procedure on a series of simulated study designs, comprised of data from our two case studies (Figure 2, pseudocode in Appendix). In each simulated study design, we generated multiple training and test sets. The predictive model was built on the training set and applied to the test set to get a prediction accuracy. In the first simulated study design, all samples in the training set came from a single batch and the test set samples came from the same batch ("within-batch" cross-validation). This design mimics the scenario where a single lab has collected all of the data for a study. Second, we simulated a study design where the training set samples were drawn from one batch, and then used to predict outcomes in a test set where samples came from a different batch ("between-batch" cross-validation). This mimics an independent validation where the predictive model is built in one lab or one study, and then applied to a separate study. In the third simulated study design, we built and tested the predictive model on training and test sets built from a mixture of batches ("pooled-batch" cross validation). This design allows us to evaluate the impact of ignoring batch when building and testing predictors in large studies with multiple batches.

In order to simulate both the within-batch and between-batch study designs described above, for both the Wang et al. (2005) and Minn et al. (2005) datasets we selected batches A and B (Figure 1). Then, for each of the batches we chose two subsets of the data - a training subset and a testing subset, each of which is equal to half of the size of the smaller batch in the dataset. In addition, for each of these subsets, we sampled such that the outcome variable was balanced and proportional to the mixture of outcomes present in the entire dataset. Thus, for each study there are four total subsets, each of which is preprocessed separately with either RMA or fRMA normalization.

After preprocessing, we then built a model on the training subsets for each batch within the study, creating one prediction model for each batch. Models were built using the defaults from the R-packages. For the within-batch scenario, we then tested each model using the testing data from the same batch. For the between-batch

scenario, we tested each model using the testing data from the other batch.

We repeated the above described procedure 100 times. That is, we performed 100 random selections of training and testing data, and then built and tested the predictive models on each of these subsets of data to obtain a cross-validated prediction accuracy measure.

In order to test against a model which ignored the effect of batch, we also performed the analysis above on the data from each study regardless of batch. That is, we sampled training data and testing data from the entire dataset, making sure to take samples from both batches. We then built the predictive model on the training data, and measured its predictive accuracy on the testing data.

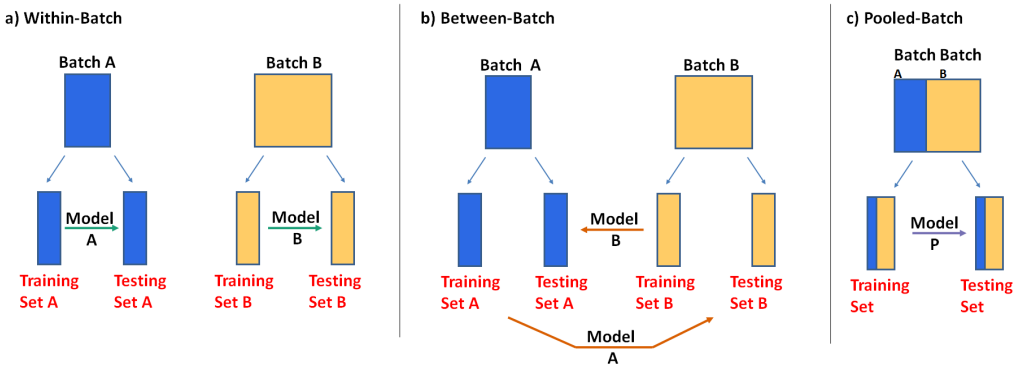


Figure 2: **Design of cross-validation for prediction accuracy, allowing for (a) within, (b) between and (c) pooled batches.** We randomly selected two mutually exclusive training and testing subsets of arrays from each batch - all four had the same number of samples, with proportional mixes of each outcome. These subsets were preprocessed separately. We built a predictive model on the training set, and then either tested it on a) the testing set of the same batch, or b) the testing set of the other batch. We iterated this process 100 times to obtain robust accuracy rates for the models. c) In addition, an internal control was created which pooled the batches together. We randomly selected two mutually exclusive training and testing equal-sized subsets of the arrays, with a mix of batch and outcome proportional to the entire dataset. We again built a predictive model on the training set and tested it on the test set, and iterated 100 times.

3.0.1 Batch only slightly impacts prediction accuracy

For both the Wang et al. (2005) and Minn et al. (2005) datasets, the prediction accuracy was only slightly impacted when models were build on one batch and

then tested on another (Table 1, boxplots presented in Appendix Figure 9). In all combinations of preprocessing method and prediction algorithm, the median cross-validated accuracy level was decreased by no more than 6 percentage points total. The interquartile range was increased in these cases by no more than 6 percentage points as well.

In general, PAM prediction performed with higher accuracy than TSP and Lasso prediction. PAM likely outperformed TSP due to the fact that the PAM model utilized more probe sets in its predictor. fRMA normalization also consistently performed either the same as, or better than, RMA normalization. These improvements were no greater than 12 percentage points.

The performance of within-batch prediction and pooled-batch prediction was also similar. This suggests that, with study designs that ensure small correlation between batch and outcome, batch does not dramatically impact prediction accuracy.

4 Batch strongly negatively impacts prediction when outcome and batch are highly correlated

Often in high-throughput studies, batch and outcome are perfectly confounded. This could occur, for example, if a laboratory first obtains diseased tissues from patients, and then later matches them to controls.

In order to examine the above described situation, we simulated a study design with the Wang et al. (2005) dataset (Figure 3). We obtained two mutually-exclusive training and testing data subsets from the Wang et al. (2005) dataset. However, for the training set, we obtained ER- samples exclusively from batch A, and ER+ samples exclusively from batch B (thus perfectly confounding batch and outcome). The testing data, however, contained a mixture of ER+ and ER- samples from each batch that reflected the original proportions in the dataset. The training and testing sets were normalized separately using RMA or fRMA preprocessing methods. Then, the predictive algorithm PAM or TSP was build on the training set. The predictive accuracy of the method was then measured using the testing set. This process was iterated 100 times in order to obtain robust results.

4.1 Prediction accuracy patterns show that batch and outcome information are conflated by predictive algorithms

We obtained 100 measures of cross-validated prediction accuracy rate for the four combinations of preprocessing method and prediction algorithm (Figure 4, Ap-

Wang et al. (2005)

	Within	Between	Pooled
RMA-Lasso	**0.81** (0.77, 0.84)	**0.81** (0.74, 0.86)	**0.81** (0.79, 0.86)
fRMA-Lasso	**0.81** (0.77, 0.84)	**0.83** (0.76, 0.88)	**0.84** (0.79, 0.86)
RMA-TSP	**0.79** (0.73, 0.83)	**0.77** (0.71, 0.84)	**0.77** (0.72, 0.84)
fRMA-TSP	**0.79** (0.73, 0.84)	**0.79** (0.72, 0.84)	**0.79** (0.72, 0.84)
RMA-PAM	**0.88** (0.84, 0.91)	**0.86** (0.83, 0.91)	**0.86** (0.84, 0.91)
fRMA-PAM	**0.88** (0.84, 0.91)	**0.88** (0.86, 0.91)	**0.87** (0.84, 0.91)

Minn et al. (2005)

	Within	Between	Pooled
RMA-Lasso	**0.94** (0.82, 1.00)	**0.88** (0.82, 0.94)	**0.94** (0.82, 0.94)
fRMA-Lasso	**0.94** (0.82, 1.00)	**0.89** (0.83, 0.94)	**0.94** (0.88, 0.94)
RMA-TSP	**0.88** (0.81, 0.94)	**0.82** (0.75, 0.89)	**0.88** (0.82, 0.88)
fRMA-TSP	**0.88** (0.81, 0.94)	**0.82** (0.77, 0.89)	**0.88** (0.77, 0.88)
RMA-PAM	**0.94** (0.88, 0.94)	**0.89** (0.82, 0.94)	**0.94** (0.88, 1.00)
fRMA-PAM	**0.94** (0.88, 1.00)	**0.94** (0.82, 1.00)	**0.94** (0.88, 1.00)

Table 1: **Predicting between batches does not largely impact accuracy.** Summaries of 100 cross-validated prediction accuracy rates, as found using the design above (Figure 2), are shown for the Wang et al. (2005) and Minn et al. (2005) datasets. Results are displayed as median (25th percentile, 75th percentile). Prediction accuracy was measured within batches, between batches, and pooling batches, in order to assess the role that batch plays in prediction. Data were preprocessed with two commonly-used preprocessing methods - RMA and fRMA - in order to see the affect of preprocessing on batch. We see that in general, prediction accuracy is not largely impacted by batch for studies with little correlation between batch and outcome.

pendix Figure 10). We divided the results into the four combinations of outcome and batch (as defined in Figure 3). The prediction accuracy is highest in the data that had the same batch/outcome combination as the data used to train the predictive model. This is because the algorithm is using information from the batch in creating the predictor - that is, the algorithm is predicting batch in addition to predicting outcome. The prediction accuracy is substantially lower in batch/outcome combinations that were not included in the training data - that is, when the algorithm cannot predict both batch and outcome simultaneously. Additionally, the overall accuracy of the predictive algorithm over the entire testing dataset when

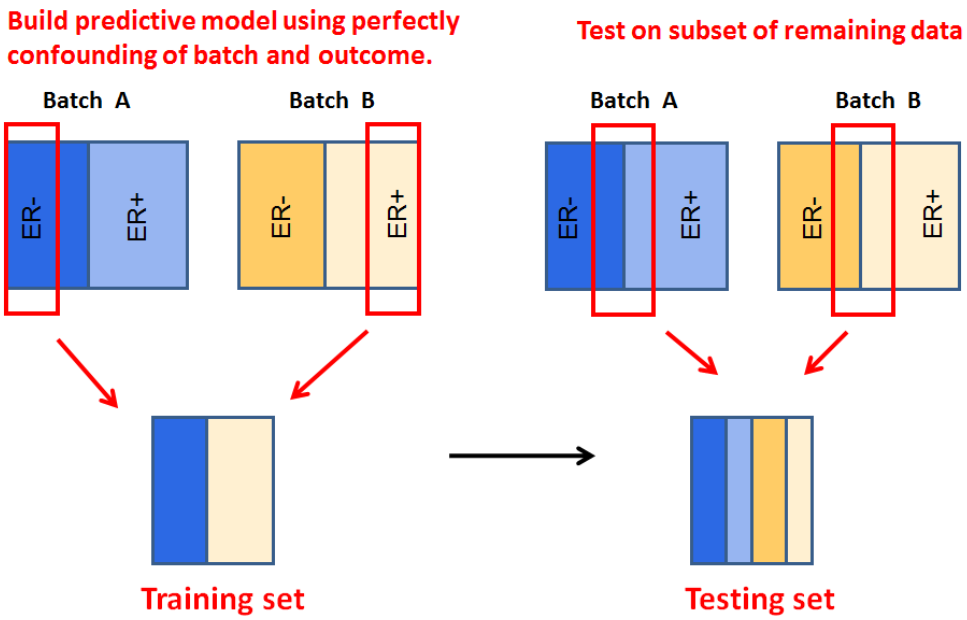


Figure 3: **Simulated design allows for predictive model to be built on subset of data with batch and outcome perfectly confounded.** We built the model on a subset of the data, using ER- samples only from batch A, and ER+ samples only from batch B. We then tested the accuracy of the model on a subset of the data from each batch and outcome combination and report the accuracy in Figure 4.

batch is perfectly confounded with outcome is lower than when batch and outcome are not confounded (as previously reported in Table 1, Appendix Figure 9). Overall, it is quite problematic when batch and outcome are confounded when building a predictive model using these algorithms, because the algorithms have no way of separating which parts of the genomic signal are due to the outcome and which are due to the batch.

For the PAM classification algorithm, we found that fRMA-normalization and RMA-normalization perform quite similarly. Both combinations out-performed TSP classification. This was likely due to the fact that PAM classification used more probes to build the predictor than TSP.

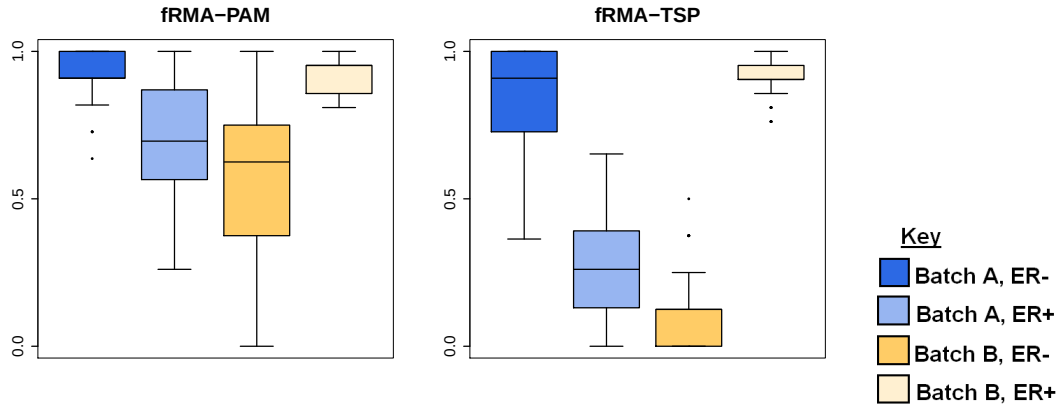


Figure 4: **Prediction accuracy rates for perfect confounding simulated design show that batch and outcome information is conflated.** The study design is presented above (Figure 3), and prediction accuracy rates are shown as boxplots for the accuracy measurements taken from the 100 iterations. Results are shown only for fRMA-preprocessing. RMA-preprocessing results are shown in the appendix, and are very similar to the fRMA-preprocessing results. The results show that batch/outcome combinations used in the training dataset (Figure 3) perform much better than batch/outcome combinations not used in the training dataset. This suggests that batch information is heavily used by the predictive algorithm when there is high confounding between batch and outcome.

5 Correcting for batch

5.1 Empirical correction for batch by removal of batch-affected probes using a hard threshold

We next examined the contribution of individual probes to batch effects. Identifying probes that are consistently affected by batch might lead to generalizable rules or sets of probes that should be removed to improve prediction accuracy.

To do this, we first examined the role that batch plays in prediction when there is perfect confounding between batch and outcome in a dataset, and saw how prediction changed as we removed what we deemed to be batch-affected probes.

We repeated the study design described above (Figure 3), but with a reduced expression matrix with batch-affected probes removed. To identify batch-affected probe sets, we fit a model to the non-testing data (as defined in Figure 3). This ensured that feature selection was not being performed on the test data (Smialowski et al., 2010). Thus for each of the two studies, we identified 100 sets of batch-

affected probes - one for each iteration of randomly choosing training and testing data. We identified batch-affected probes by fitting probe set expression levels, Y_{ij} with the model

$$Y_{ij} = \beta_{0i} + \beta_{1i}\text{batch}_\text{j} + \beta_{2i}\text{outcome}_\text{j} + \varepsilon_{ij} \tag{1}$$

for probe set i on array j. We then defined a probe set to be batch-affected based on the following hard threshold:

$$BA_i = \begin{cases} 1 & \text{if } p_i < Q_x(\mathbf{p}) \\ 0 & \text{if } p_i > Q_x(\mathbf{p}) \end{cases} \tag{2}$$

where p_i is the p-value for β_{1i}, and Q_x is the $x\%$ empirical quantile of $\mathbf{p}$, the set of p_i values for $i \in (1, 22283)$, where 22283 is the total number of probe sets on the array.

For each iteration of randomly building the simulated design (as described in Figure 3), we removed the batch-affected probes from both the training and testing sets, with Q_x ranging from 10% to 60%. We then calculated the prediction accuracy of predictors on the testing subset of the data.

The above-described procedure was performed on the Minn et al. (2005) and Wang et al. (2005) datasets. Thus, we determined 100 candidate set of batch-affected probes for each of the studies. We then sought to determine the predictability of these batch-affected probes, both by comparing the sets found using the Minn et al. (2005) and Wang et al. (2005) datasets, and by creating a model, based on probe sequence, to describe the obtained p-value of the batch coefficient in model 1.

5.1.1 Removal of batch-affected probes increases prediction accuracy for Wang et al. (2005) data

We performed the described simulated design using the Wang et al. (2005) dataset with all pairwise combinations of preprocessing technique (RMA, fRMA) and prediction algorithm (PAM and TSP). We consistently saw improvement in overall prediction accuracy as batch-affected probes were removed (Figure 6. In general, we saw greater improvement with TSP prediction than PAM prediction. For both methods of preprocessing, after removing batch-affected probes, TSP prediction performs approximately the same as PAM prediction. Recall that PAM prediction performed more accurately than TSP when no batch-affected probes were removed.

We first examined a density plot of the parameter estimates β_1 and β_2 for model 1 (Figure 5 (a)). Note that this is a display of the parameter estimates from

all of the 100 iterations of fitting model 1. We saw that the coefficients associated with batch had a wider distribution than those associated with outcome - that is, it appears that batch has a larger effect on expression than outcome for this dataset. Notably, neither density plot displays a normal distribution, which may be evidence that our model is not perfectly parametrized.

5.1.2 Removal of batch-affected probes slightly increases prediction accuracy for Minn et al. (2005) data

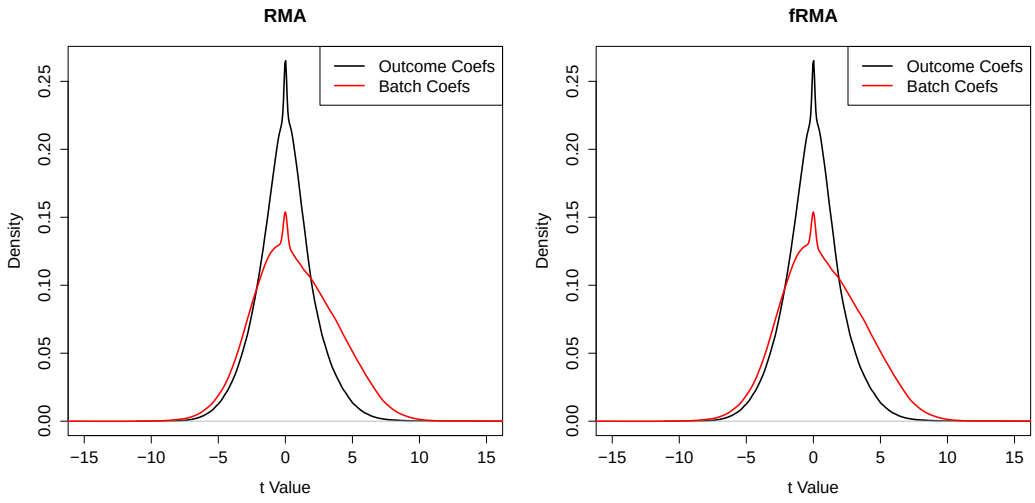
For the Minn et al. (2005) dataset, the removal of batch-affected probes by the method described improved prediction slightly in three of the four combinations of preprocessing and prediction algorithm (Appendix Figure 11). The improvement was not as dramatic as that for the Wang et al. (2005) dataset. A density plot (Figure 5 (b)) of the parameter estimates from the model 1 show very similar distributions for the parameter estimates β_1 and β_2. Thus, for this data the signal and the batch effects had a very similar intensity - something that is not often seen in studies with more subtle outcomes. This may play a role in why our method of removing batch-affected probes does not work for the fRMA-PAM combination.

5.1.3 There is little overlap between the batch-affected probes found using the Wang et al. (2005) and Minn et al. (2005) datasets

We next sought to determine any similarities in the batch-affected probes determined by the Wang et al. (2005) and Minn et al. (2005) RMA-preprocessed datasets. First, we compared within each study the pairwise overlap of batch-affected probes between each of the 100 iterations fitting model 1 to the non-testing data (Figure 2). In general we found that the overlap did not exceed what would happen by chance if randomly selecting probes and calling them batch-affected (Appendix Table 3).

We also compared the batch-affected probes identified in the Wang et al. (2005) and Minn et al. (2005) datasets. To do this, we identified probes that were called batch-affected in at least 50% of the 100 iterations. We then compared these probes between the two studies. We found little overlap between the batch-affected probes (Table 2). Even in the case of a very conservative definition of batch-affected (that is, defining 60% of the probes in a study as batch-affected, as described previously in model 1), we see only a 36% overlap. We have concluded that batch-affected probes cannot be generalized across samples or studies.

(a) Wang et al. (2005)



(b) Minn et al. (2005)

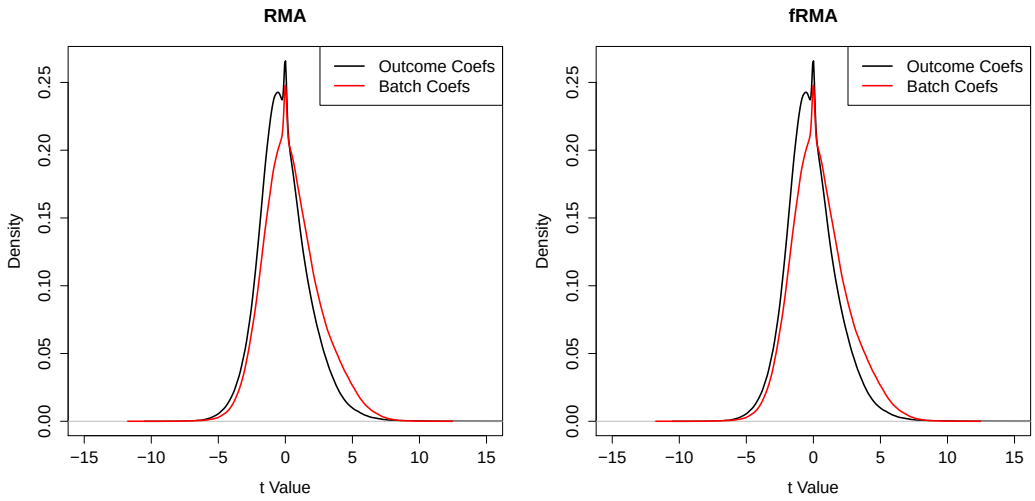


Figure 5: Density plots of β_1 and β_2 estimates from the fitted model 1 on 100 iterations of the simulated design (Figure 3), using the a) Wang et al. (2005) data, and b) Minn et al. (2005) data. We utilized both RMA- and fRMA- normalization for each study.

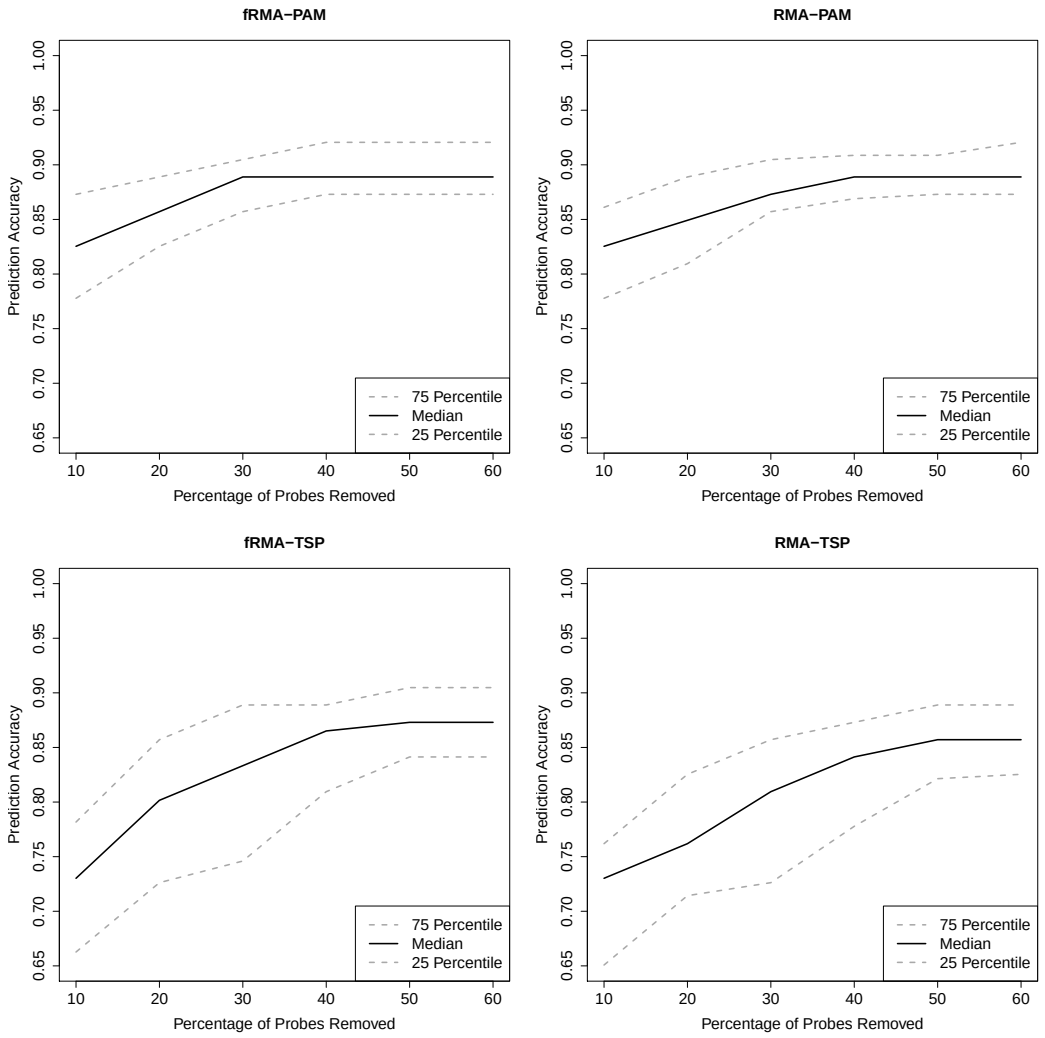


Figure 6: **Prediction accuracy improves as batch-affected probes are removed.** Batch-affected probes in the Wang et al. (2005) dataset were determined by fitting model 1 and selecting probes with the most significant β_1 estimates.

5.1.4 Batch-affected probes cannot be predicted using linear modeling based on probe sequence

We next sought to assess whether or not the extent to which probes are affected by batch might be a function of the nucleotide sequence of the probes. Nucleotide sequence has previously been shown to affect probe expression levels, likely because of the physical properties of the different nucleotides (see for example Wu et al.

Percentage Batch-Affected	RMA Agreement	fRMA Agreement
10%	0.02	0.02
20%	0.06	0.05
30%	0.11	0.10
40%	0.18	0.16
50%	0.26	0.25
60%	0.36	0.35

Table 2: **Proportion agreement for batch-affected probes for Wang et al. (2005) and Minn et al. (2005) shows little overlap beyond what is expected by chance.** Because batch-affected probes were determined using the non-testing data for each of the 100 random iterations, we determined 100 different sets of batch-affected probes. Results for the 100 iterations are displayed as median (25th percentile, 75th percentile). These results call into question whether batch-affected probes can be generalized beyond a single study, regardless of preprocessing method.

2004; Johnson et al. 2006). We defined the extent to which a probe is batch-affected as the $-\log_{10}(p_k)$, where p_k is the p-value for the batch coefficient in model 1 for probe k. Our model was

$$-\log_{10}(p_k) = \sum_i \sum_j \mu_{ij} I(B_i = j) \tag{3}$$

for basepair j (A, T, C or G) and position i (1-25) on probe k. Note that to find the p-value from equation 1, we were required to use probe-level expression data (since the sequence information is applicable at the probe-level, not the probe-set-level). Thus, we could not apply RMA and fRMA preprocessing to candidate datasets, since both methods summarize probe-level data into probe sets. Instead of preprocessing, we performed GCRMA-background correction (as described in Wu et al., 2004) to correct for probe-effects, and took the log_2 of the expression values.

To assess the performance of this model, we performed a meta-analysis using 262 microarray studies, all of which used the Affymetrix hgu133plus2 platform and had at least 6 microarrays in the study. (Specific study names can be found on the author's website, `http://biostat.jhsph.edu/~jleek/PracticalBatch/`.) We divided each study into two batches. To determine these two batches, we found the median array date within each study, and assigned arrays with dates earlier than or equal to the median to be in batch A, and arrays with dates later than the median to be in batch B. Furthermore, we specified that each batch must have at least 25% of the arrays. Thus, studies with mostly one batch (for example, studies performed

mostly on one day) were excluded. We chose to define batches in this conservative way in order to maximize the number of studies we analyzed while still maintaining a suitable, consistent and reproducible estimation of batch by date.

Because we did not have one consistent outcome throughout the 262 studies, we excluded the outcome variable when fitting model 1, thus fitting a simple linear model with batch as the only covariate. We also fit the model to the Wang et al. (2005) and Minn et al. (2005) datasets, run on the hgu133a Affymetrix platform. For ease of comparison, for these two studies we also only used the batch covariate and not the outcome covariate when fitting model 1. For these two studies we used the same batch covariate found by the array date histogram (Figure 1).

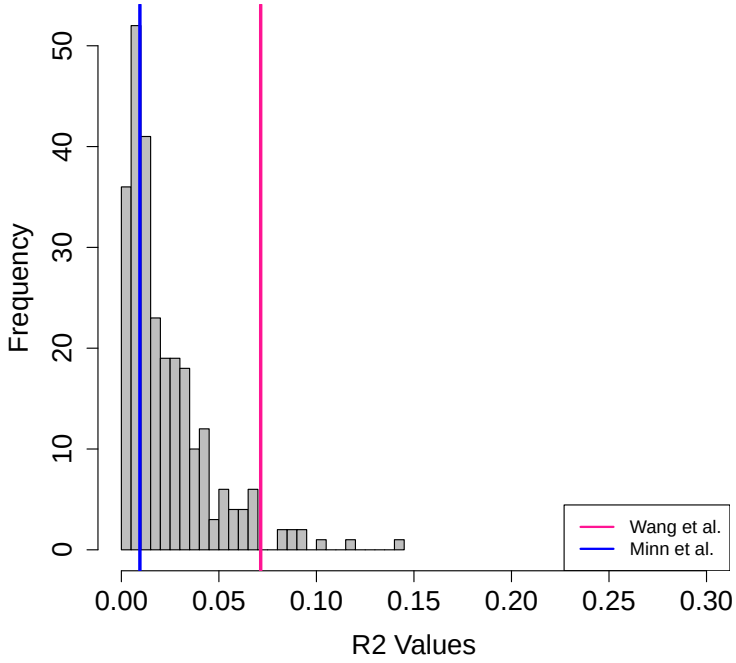


Figure 7: **Histogram of R^2 values from model 3 for background-adjusted probe-level data from 262 microarray studies, as well as the Wang et al. (2005) and Minn et al. (2005) datasets.** R^2 values show that no more than 15% of the variation in the measure of batch-affectedness in the probes is due to the probe sequence.

In general, we found poor model fit for the 262 studies, as well as the Wang et al. (2005) and Minn et al. (2005) datasets. The models predicted less than 10% of the variability in the p-values associated with batch (Figure 7). We conclude

that nucleotide sequence has little predictive power to identify our estimated batch-affected probes. That is, it appears that nucleotide sequence of specific probes is not a useful predictor of which probes will be susceptible to batch effects.

We next examined specific parameter estimates for each of these studies. In general, the parameter estimates were quite close to zero for the vast majority of the studies (Figure 8). In some cases, the parameter estimates differed from zero. However, they did not differ in a consistent way, leading us to conclude that whether or not a probe is batch-affected can not be consistently estimated using model 3. Graphs of parameter estimates for each of the 262 studies can be found on the author's website (`http://biostat.jhsph.edu/~jleek/PracticalBatch/`).

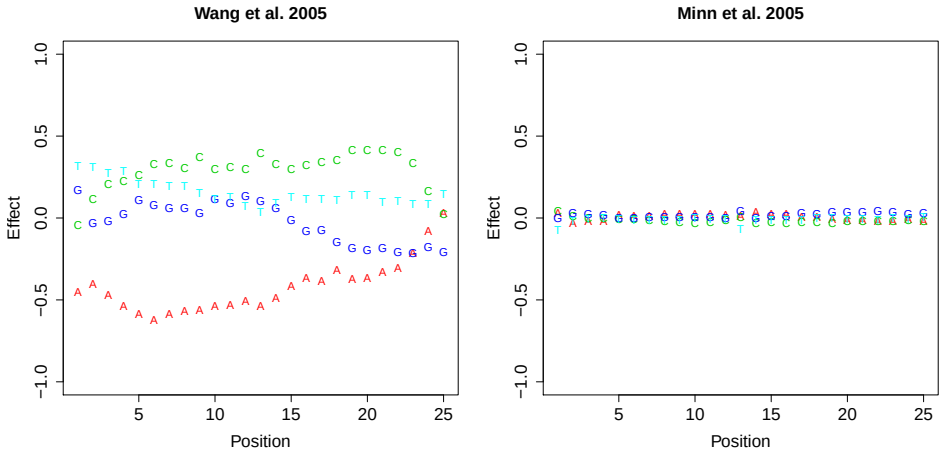


Figure 8: **Examples of parameter estimates from model 3 for the Wang et al. (2005) and Minn et al. (2005) datasets, as well as 262 others, do not show a consistent pattern.** Model 3 was fit on GCRMA-background corrected probe-level data with the outcome coefficient excluded. The vast majority of the models displayed coefficients close to zero, as in the Minn et al. (2005) dataset. Some studies did show coefficient patterns, such as in the Wang et al. (2005) dataset. However, these patterns were not consistent from study to study.

6 Conclusions

We have found that prediction is negatively affected by batch. In the case of study designs in which batch and outcome are not confounded, we see that prediction is only somewhat negatively impacted across all preprocessing techniques and prediction algorithms. In the case of study designs in which batch and outcome are

perfectly confounded, we see that prediction accuracy is substantially reduced by batch.

Furthermore, we show that the removal of empirically-determined batch-affected probes can greatly increase prediction accuracy when batch and outcome are perfectly confounded. This shows promise that certain probes are more susceptible to batch than others within a study. However, we show very limited success in determining batch-affected probes using modeling. Our results suggest that batch-effect removal for prediction is still critical, but can not be accomplished through sequence modeling alone. Instead, new methods are required to identify and remove batch effected probes in individual studies.

7 Appendix

In addition to the graphical design of the leave-half-out cross validation design presented in Figure 3, we created pseudocode that describes the process.

Within- and Between-Batch Cross-Validation

1. For the batch with the smaller sample size (batch A), randomly choose half of the arrays from outcome 1 (n_1) and half of the arrays from outcome 2 (n_2). This will be the training data for batch A.
2. For the batch with the larger sample size (batch B), randomly chose n_1 arrays from outcome 1, and n_2 arrays from outcome 2, where n_1 and n_2 are defined above. This will be the training data for batch B.
3. Preprocess the training data for batch A and batch B separately using the preprocessing algorithm of your choosing.
4. Build a predictive model for each of the batches, thus creating model A and model B.
5. Create test data for batch A by randomly choosing n_1 arrays from outcome 1 (that were not chosen in the training data) and n_2 arrays from outcome 2. Note that if batch A has an even number of arrays for each outcome, then the testing data will just be the remaining arrays that were not in the training set.
6. Create a test data for batch B by randomly choosing n_1 arrays from outcome 1 (that were not chosen in the training data) and n_2 arrays from outcome 2.
7. *For Within-Batch Cross-validation:* Predict the outcomes for test data A using model A, and for test data B using model B.
8. *For Between-Batch Cross-validation:* Predict the outcomes for test data A using model B, and for test data B using model A.
9. Repeat 100 times to obtain robust accuracy rates.

Pooled Batch Cross-Validation

1. Do not subdivide data into batches. Randomly choose half of the arrays from outcome 1 (N_1) and half of the arrays from outcome 2 (N_2). This will be the training data.
2. Preprocess the training data.
3. Build a predictive model using the training data.
4. Create test data by randomly choosing N_1 arrays from outcome 1 and N_2 arrays from outcome 2. Note that if the data has an even number of arrays for each outcome, then the testing data will just be the remaining arrays that were not in the training set.
5. Predict the outcomes for the test data using the model built on the training data.
6. Repeat 100 times to obtain robust accuracy rates.

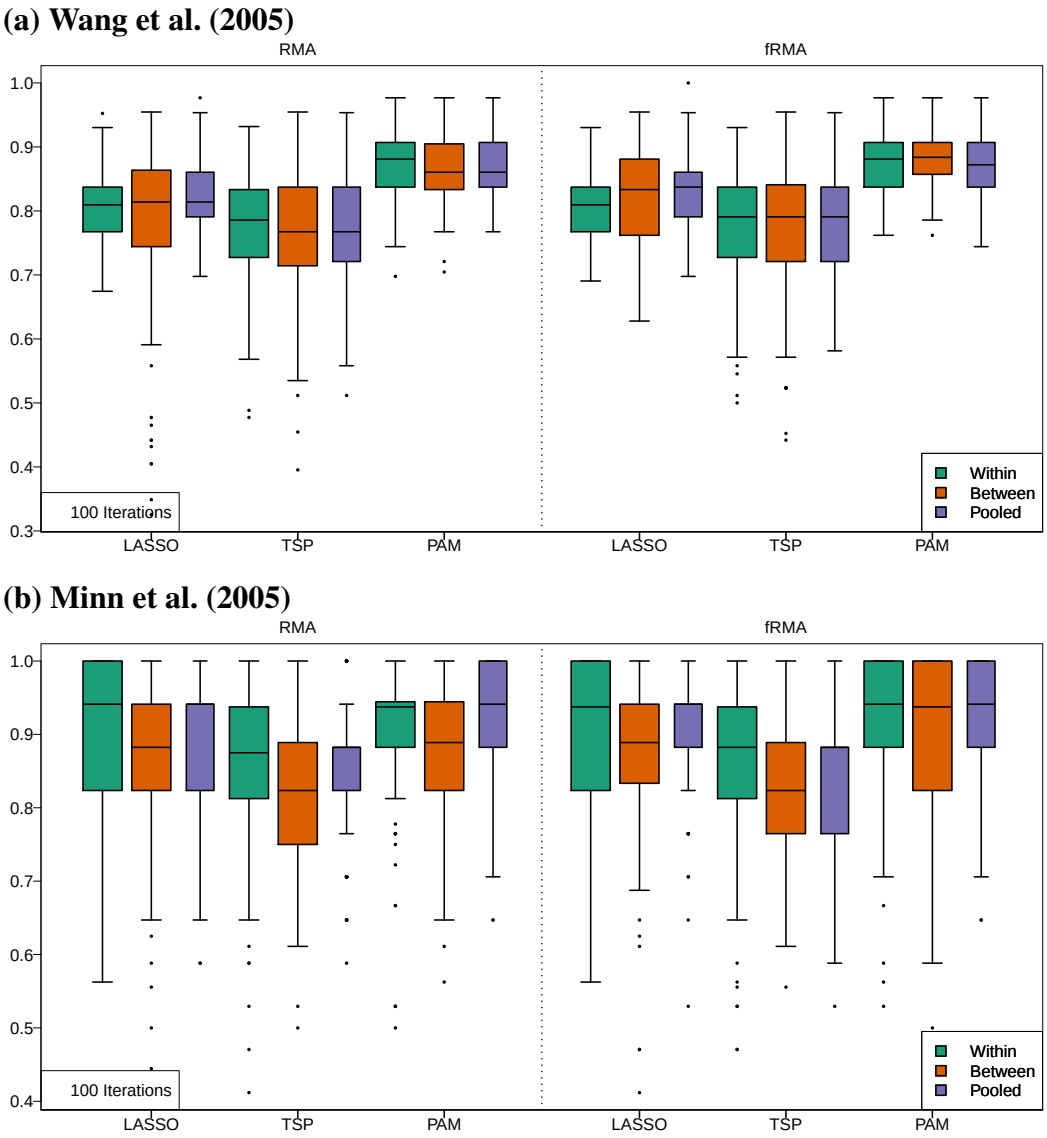


Figure 9: **Predicting between batches increases variance and decreases accuracy.** Boxplots of 100 cross-validated prediction accuracy rates, as found using the cross-validation design described in the paper are shown for a) Wang et al. (2005) and b) Minn et al. (2005) datasets. Prediction accuracy was measured within batches, between batches, and pooling batches, in order to assess the role that batch plays in prediction. Data were preprocessed with two commonly-used preprocessing methods - RMA and fRMA - in order to see the affect of preprocessing on batch. We see that in general, the variance of accuracy rates increases when building the model on one batch and testing it on another (between-batch).

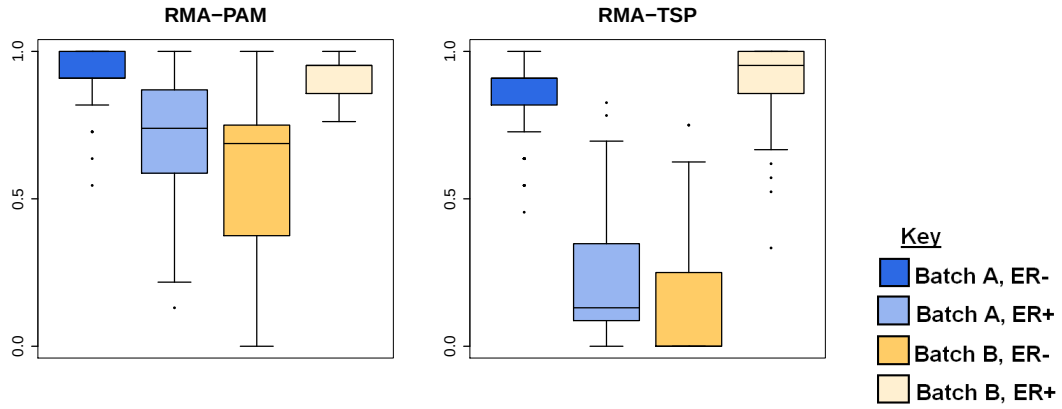


Figure 10: **Prediction accuracy rates for perfect confounding simulated design show that fRMA-PAM combination is optimal.** The study design is presented above (Figure 3), and prediction accuracy rates are shown as boxplots for the accuracy measurements taken from the 100 iterations. Results are shown only for RMA-preprocessing here. fRMA-preprocessing results are shown in Figure 4.

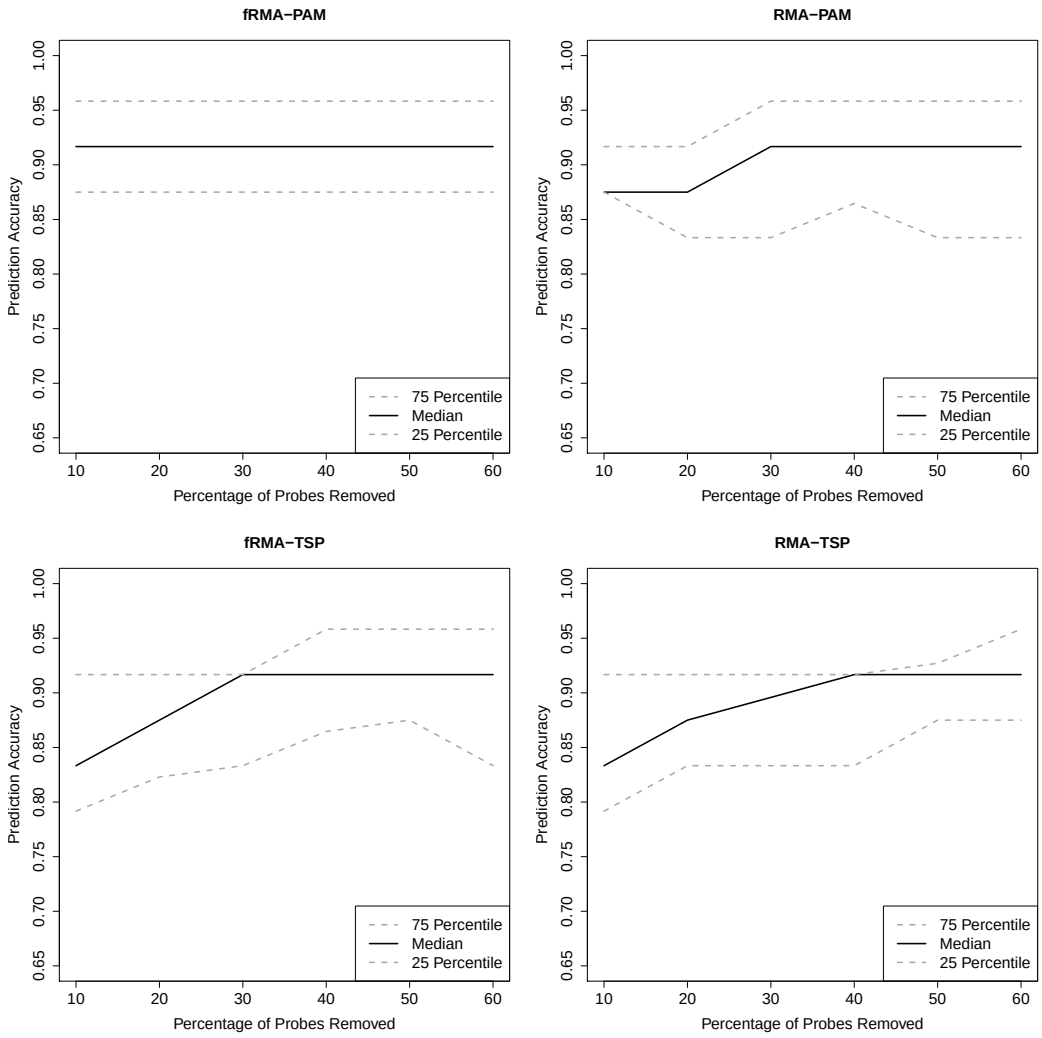


Figure 11: **Prediction accuracy only somewhat improves as batch-affected probes are removed.** Batch-affected probes in the Minn et al. (2005) dataset were determined by fitting model 1 and selecting probes with the most significant β_1 estimates.

Wang et al. (2005)

Percent Batch-Affected	RMA	fRMA
10%	**0.08** (0.08, 0.08)	**0.08** (0.08, 0.08)
20%	**0.16** (0.16, 0.17)	**0.17** (0.16, 0.17)
30%	**0.25** (0.25, 0.26)	**0.25** (0.25, 0.25)
40%	**0.34** (0.34, 0.35)	**0.34** (0.33, 0.34)
50%	**0.43** (0.43, 0.44)	**0.43** (0.42, 0.43)
60%	**0.53** (0.52, 0.53)	**0.52** (0.52, 0.53)

Minn et al. (2005)

10%	**0.07** (0.07, 0.08)	**0.07** (0.07, 0.07)
20%	**0.15** (0.15, 0.15)	**0.15** (0.15, 0.15)
30%	**0.23** (0.23, 0.24)	**0.23** (0.23, 0.23)
40%	**0.31** (0.31, 0.32)	**0.31** (0.31, 0.32)
50%	**0.40** (0.39, 0.40)	**0.40** (0.39, 0.40)
60%	**0.48** (0.48, 0.49)	**0.49** (0.48, 0.49)

Table 3: **Proportion overlap of batch-affected probes in pairwise comparisons of the 100 iterations shows little consistency.** 100 sets of batch-affected probes were generations from model 1, based on the simulated design described in figure 3. We then made pairwise comparisons of the proportion overlap of batch-affected probes, and report the median (25th percentile, 75th percentile).

References

Akey, J. M., S. Biswas, J. T. Leek, and J. D. Storey (2007): "On the design and analysis of gene expression studies in human populations." *Nature Genetics*, 39, 807–8; author reply 808–9, URL http://dx.doi.org/10.1038/ng0707-807.

Allison, D. B., X. Cui, G. P. Page, and M. Sabripour (2006): "Microarray data analysis: from disarray to consolidation and consensus." *Nature Reviews Genetics*, 7, 55–65, URL http://dx.doi.org/10.1038/nrg1749.

Baggerly, K. A., S. R. Edmonson, J. S. Morris, and K. R. Coombes (2004): "High-resolution serum proteomic patterns for ovarian cancer detection." *Endocrine-Related Cancer*, 11, 583–4; author reply 585–7, URL http://dx.doi.org/10.1677/erc.1.00868.

Carroll, J. S., C. A. Meyer, J. Song, W. Li, T. R. Geistlinger, J. Eeckhoute, A. S. Brodsky, E. K. Keeton, K. C. Fertuck, G. F. Hall, Q. Wang, S. Bekiranov, V. Sementchenko, E. A. Fox, P. A. Silver, T. R. Gingeras, X. S. Liu, and M. Brown (2006): "Genome-wide analysis of estrogen receptor binding sites." *Nature Genetics*, 38, 1289–97, URL http://dx.doi.org/10.1038/ng1901.

Edgar, R., M. Domrachev, and A. E. Lash (2002): "Gene Expression Omnibus: NCBI gene expression and hybridization array data repository," *Nucleic Acids Research*, 30, 207–10, URL http://dx.doi.org/10.1093/nar/30.1.207.

Efron, B., T. Hastie, I. Johnstone, and R. Tibshirani (2004): "Least angle regression," *The Annals of Statistics*, 32, 407–499, URL http://dx.doi.org/10.1214/009053604000000067.

Eisen, M. B., P. T. Spellman, P. O. Brown, and D. Botstein (1998): "Cluster analysis and display of genome-wide expression patterns," *Proceedings of the National Academy of Sciences*, 95, 14863–8, URL http://dx.doi.org/10.1073/pnas.95.25.14863.

Geman, D., C. D'Avignon, D. Q. Naiman, and R. L. Winslow (2004): "Classifying gene expression profiles from pairwise mRNA comparisons," *Statistical Applications in Genetics and Molecular Biology*, 3, 19, URL http://dx.doi.org/10.2202/1544-6115.1071.

Irizarry, R. A., B. M. Bolstad, F. Collin, L. M. Cope, B. Hobbs, and T. P. Speed (2003): "Summaries of Affymetrix GeneChip probe level data," *Nucleic Acids Research*, 31, e15, URL http://dx.doi.org/10.1093/nar/gng015.

Johnson, W. E., C. Li, and A. Rabinovic (2007): "Adjusting batch effects in microarray expression data using empirical Bayes methods." *Biostatistics*, 8, 118–27, URL http://dx.doi.org/10.1093/biostatistics/kxj037.

Johnson, W. E., W. Li, C. a. Meyer, R. Gottardo, J. S. Carroll, M. Brown, and X. S. Liu (2006): "Model-based analysis of tiling-arrays for ChIP-chip." *Proceedings of the National Academy of Sciences of the United States of America*,

103, 12457–62, URL http://dx.doi.org/10.1073/pnas.0601180103.

Leek, J. T., R. B. Scharpf, H. C. Bravo, D. Simcha, B. Langmead, W. E. Johnson, D. Geman, K. A. Baggerly, and R. A. Irizarry (2010): "Tackling the widespread and critical impact of batch effects in high-throughput data." *Nature Reviews Genetics*, 11, 733–9, URL http://dx.doi.org/10.1038/nrg2825.

Leek, J. T. and J. D. Storey (2007): "Capturing heterogeneity in gene expression studies by surrogate variable analysis." *PLoS Genetics*, 3, 1724–35, URL http://dx.doi.org/10.1371/journal.pgen.0030161.

Luo, J., M. Schumacher, A. Scherer, D. Sanoudou, D. Megherbi, T. Davison, T. Shi, W. Tong, L. Shi, H. Hong, C. Zhao, F. Elloumi, W. Shi, R. Thomas, S. Lin, G. Tillinghast, G. Liu, Y. Zhou, D. Herman, Y. Li, Y. Deng, H. Fang, P. Bushel, M. Woods, and J. Zhang (2010): "A comparison of batch effect removal methods for enhancement of prediction performance using MAQC-II microarray gene expression data," *The Pharmacogenomics Journal*, 10, 278–91, URL http://dx.doi.org/10.1038/tpj.2010.57.

McCall, M. N., B. M. Bolstad, and R. A. Irizarry (2010): "Frozen Robust Multi-Array Analysis (fRMA)," *Biostatistics*, 11, 242–253, URL http://dx.doi.org/10.1093/biostatistics/kxp059.

Mecham, B. H., P. S. Nelson, and J. D. Storey (2010): "Supervised normalization of microarrays," *Bioinformatics*, 26, 1308–15, URL http://dx.doi.org/10.1093/bioinformatics/btq118.

Minn, A. J., G. P. Gupta, P. M. Siegel, P. D. Bos, W. Shu, D. D. Giri, A. Viale, A. B. Olshen, W. L. Gerald, and J. Massagué (2005): "Genes that mediate breast cancer metastasis to lung," *Nature*, 436, 518–24, URL http://dx.doi.org/10.1038/nature03799.

Parkinson, H., M. Kapushesky, N. Kolesnikov, G. Rustici, M. Shojatalab, N. Abeygunawardena, H. Berube, M. Dylag, I. Emam, A. Farne, E. Holloway, M. Lukk, J. Malone, R. Mani, E. Pilicheva, T. F. Rayner, F. Rezwan, A. Sharma, E. Williams, X. Z. Bradley, T. Adamusiak, M. Brandizi, T. Burdett, R. Coulson, M. Krestyaninova, P. Kurnosov, E. Maguire, S. G. Neogi, P. Rocca-Serra, S.-A. Sansone, N. Sklyar, M. Zhao, U. Sarkans, and A. Brazma (2009): "ArrayExpress update–from an archive of functional genomics experiments to the atlas of gene expression," *Nucleic Acids Research*, 37, D868–72, URL http://dx.doi.org/10.1093/nar/gkn889.

Smialowski, P., D. Frishman, and S. Kramer (2010): "Pitfalls of supervised feature selection." *Bioinformatics (Oxford, England)*, 26, 440–3, URL http://dx.doi.org/10.1093/bioinformatics/btp621.

Tibshirani, R. (1996): "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society: Series B*, 58, 267–288, URL http://www.jstor.org/stable/2346178.

Tibshirani, R., T. Hastie, B. Narasimhan, and G. Chu (2002): "Diagnosis of multiple cancer types by shrunken centroids of gene expression," *Proceedings of the National Academy of Sciences*, 99, 6567–72, URL http://dx.doi.org/10.1073/pnas.082099299.

Wang, Y., J. G. M. Klijn, Y. Zhang, A. M. Sieuwerts, M. P. Look, F. Yang, D. Talantov, M. Timmermans, M. E. Meijer-van Gelder, J. Yu, T. Jatkoe, E. M. J. J. Berns, D. Atkins, and J. A. Foekens (2005): "Gene-expression profiles to predict distant metastasis of lymph-node-negative primary breast cancer," *The Lancet*, 365, 671–9, URL http://dx.doi.org/10.1016/S0140-6736(05)17947-1.

Wu, Z., R. A. Irizarry, R. Gentleman, F. Martinez-Murillo, and F. Spencer (2004): "A Model-Based Background Adjustment for Oligonucleotide Expression Arrays," *Journal of the American Statistical Association*, 99, 909–917, URL http://dx.doi.org/10.1198/016214504000000683.