Biostatistics (2022) **00**, 00, *pp.* 1–17
https://doi.org/10.1093/biostatistics/kxab046

Separating and reintegrating latent variables to improve classification of genomic data

NORA YUJIA PAYNE*, JOHANN A. GAGNON-BARTSCH

Department of Statistics, University of Michigan, 1085 S. University Ave., Ann Arbor, MI 48109, USA
yujiap@umich.edu

Summary

Genomic data sets contain the effects of various unobserved biological variables in addition to the variable of primary interest. These latent variables often affect a large number of features (e.g., genes), giving rise to dense latent variation. This latent variation presents both challenges and opportunities for classification. While some of these latent variables may be partially correlated with the phenotype of interest and thus helpful, others may be uncorrelated and merely contribute additional noise. Moreover, whether potentially helpful or not, these latent variables may obscure weaker effects that impact only a small number of features but more directly capture the signal of primary interest. To address these challenges, we propose the cross-residualization classifier (CRC). Through an adjustment and ensemble procedure, the CRC estimates and residualizes out the latent variation, trains a classifier on the residuals, and then reintegrates the latent variation in a final ensemble classifier. Thus, the latent variables are accounted for without discarding any potentially predictive information. We apply the method to simulated data and a variety of genomic data sets from multiple platforms. In general, we find that the CRC performs well relative to existing classifiers and sometimes offers substantial gains.

Keywords: Classification; Gene expression; Linear discriminant analysis.

1. Introduction

High-dimensional classification is a ubiquitous and challenging problem in genomics. Classical methods, such as linear discriminant analysis, cannot be used directly since usual estimates for population parameters are poor when the number of features exceeds the number of observations. Various strategies have been developed to address this issue. In genomics, popular approaches include various forms of dimension reduction (Li, 2010), feature selection (Saeys *and others*, 2007), as well as the "independence rule" (Bickel and Levina, 2004). These strategies are embedded in many popular classifiers (Zou and Hastie, 2005; Dudoit and Fridlyand, 2003; Nguyen and Rocke, 2002; Tibshirani *and others*, 2002).

Many of these approaches work by discarding or disregarding aspects of the data. For example, feature selection methods aim to select a small subset of relevant features while ignoring many null features. Dimension reduction strategies such as principal components analysis (PCA) focus attention on a low-dimensional subspace of the original feature space, disregarding uninformative dimensions that merely

*To whom correspondence should be addressed.

contain noise. The independence rule, which disregards correlations between features, is yet another example. When the assumptions underlying these approaches hold, they can yield substantial improvements in classifier fit and accuracy by reducing the number of parameters one must estimate.

However, these assumptions are not always appropriate for genomic data (Hall *and others*, 2014). Consider a hypothetical data set of methylation signatures generated from a sample of lung tumors. The tumors are of either Type A or Type B, and we wish to train a classifier on the methylation data for the purpose of classifying new lung tumors. Although tumor type is the signal of primary interest in this classification task, it is likely only one of many signals present in the data, many of which arise from various biological and environmental factors. For instance, it has been observed that smokers and nonsmokers differ in their DNA methylation signatures across the whole genome (Wan *and others*, 2012). Air pollution and exposure to fine particulates have also been found to affect gene expression and methylation patterns (Bind *and others*, 2014; Quay *and others*, 1998). Since these variables affect genetic activity across many genes, the data contain widespread correlations across the features and many dense signals in addition to the signal of primary interest (Boyle *and others*, 2017). As an added complication, these additional variables are generally latent. For instance, smoking status and fine particulate exposure are frequently not recorded, difficult to measure, or under-reported. These issues are especially challenging when the signal of primary interest is sparse.

In this setting, the aforementioned strategies for coping with high-dimensionality may not be optimal. Dimension reduction strategies may ignore informative dimensions of the feature space, and feature selection strategies may discard discriminative variables. The independence rule is similarly suboptimal. For example, in Fan *and others* (2012), it is shown that leveraging correlations between variables can yield further reduction in misclassification error, as opposed to simply ignoring them. Thus when dense latent signals are present, commonly used strategies may not only discard noise but relevant information as well.

There is a growing body of work on high-dimensional estimation and prediction in the presence of dense signals (Cook *and others*, 2012; Dobriban and Wager, 2018; Dicker, 2012), and a substantial portion of this literature focuses on such problems as arising from latent variables specifically (Kneip and Sarda, 2011; Zheng *and others*, 2017; Dicker, 2012). One way to account for latent variables consists of performing PCA and using the leading principal components as additional predictors in the model; Kneip and Sarda (2011) discuss this idea in the context of linear regression. Another approach to account for the effects of latent variables is to assume "conditional sparsity" of the coefficient vector or covariance matrix in a linear regression model (Zheng *and others*, 2017; Fan *and others*, 2013). Roughly speaking, these conditional methods model and condition on the presence of latent factors in order to accurately estimate the model parameters of interest.

Our main contribution in this article is a framework for training a classifier in the presence of latent variables. We focus primarily on the setting in which the signal of primary interest is sparse, the latent signals are dense, and the latent signals are potentially correlated with the signal of primary interest. The key idea is to decompose the data into a dense low-rank component and a sparse component, train separate classifiers on these components, and combine these classifiers into a single ensemble. Importantly, the sparse component is adjusted for the dense low-rank component, so that these two components contain distinct information. We propose an algorithm for performing this simultaneous decomposition and adjustment, which we call *cross-residualization*. We also propose a specific instantiation of our framework, the *cross-residualization classifier* (CRC) for the setting in which the class-conditional distributions are Gaussian. In this setting, simple linear classifiers may be trained on each of the separate components. However, the CRC framework is modular, and for more complex settings, it can accommodate flexible implementations with the component-wise classifiers chosen according to the specific application at hand.

The article is organized as follows. Section 2 introduces the model and discusses the challenges and opportunities presented by latent variables in more detail. Section 3 presents the CRC. Section 4 contains simulations which demonstrate the advantages of our approach and offer comparisons to classifiers commonly used in genomic applications. Section 5 applies the method to a diverse collection of genomic data sets. Section 6 concludes.

2. Model and motivation

Here, we present a sequence of models to illustrate the challenges and opportunities that latent variables present for classification of genomic data.

Simple model

Suppose an observation is given by (S, T), where $S \in \mathbb{R}^{1\times p}$ is the feature vector and $T \in \{\pm 1\}$ is the class label. For example, S might be a vector of expression levels for p genes and T might indicate the presence or absence of a disease. Consider a model for S in which no latent factors are present, $S_{1\times p} = T_{1\times 1}\gamma_{1\times p} + \epsilon_{1\times p}$, where $T \in \{\pm 1\}$ with $P(T = 1) = \pi \in (0, 1)$. We assume that $\epsilon \sim N(0, \Sigma)$, where $\Sigma = \text{diag}(\sigma_1^2, \ldots, \sigma_p^2)$. We assume that T is binary and $\pi = 1/2$ in this article for simplicity; however, our discussion and methodology readily generalize to the multiclass setting and to cases in which the prior probabilities are not uniform.

Under this simple model, $S \mid T \sim N(T\gamma, \Sigma)$. The optimal Bayes classifier is $f(x) = \text{sign}\left(w_{\text{simple}} \cdot x\right)$, where $w_{\text{simple}} \propto \Sigma^{-1}\gamma' = (\gamma_1/\sigma_1^2, \ldots, \gamma_p/\sigma_p^2)'$. In many applications, the vector γ is sparse since the variable of primary interest affects only a small fraction of the genes. Since Σ^{-1} is a diagonal matrix, a sparse γ implies the optimal weight vector w_{simple} is also sparse. Thus, methods such as the independence rule and variable selection techniques can be used to efficiently estimate w_{simple}, despite it being a p-dimensional parameter.

Uncorrelated latent variables

In reality, genomic data contain the effects of biological variables other than the variable of primary interest. These variables are generally latent, so observations might be more appropriately modeled via a latent factor model

$$Z_{1\times p} = T_{1\times 1}\gamma_{1\times p} + L_{1\times r}\alpha_{r\times p} + \epsilon_{1\times p}, \tag{2.1}$$

where L represents the latent variables. Here we assume that $L \sim N(0, \Psi)$ and is uncorrelated with T, that is, $L \mid T \sim N(0, \Psi)$. We assume that the latent biological variables tend to affect a large proportion of genes, so the coefficient matrix α is dense. The number of latent variables, r, is unknown, but we assume that $r < n - 1$. Note that the feature vector in model (2.1) is denoted by Z instead of S; we continue to reserve S to denote $S = T\gamma + \epsilon$ (which is unobserved in this model). Similar to before, we can think of Z as representing gene expression and T as disease status, although gene expression is now a function of the latent variables in addition to disease status. Latent variable models similar to (2.1) appear frequently in the batch effect normalization literature, where the latent variables represent unwanted technical factors to be removed (Leek and Storey, 2007, 2008; Listgarten *and others*, 2010; Gagnon-Bartsch and Speed, 2012; Sun *and others*, 2012; Parker *and others*, 2014; Wang *and others*, 2017). Here, however, we consider the latent variables to be biological variables responsible for variation across a large number of features.

Under the uncorrelated factor model in (2.1), $Z \mid T \sim N(T\gamma, \alpha'\Psi\alpha + \Sigma)$. The optimal Bayes classifier is again a linear classifier of the form $f(x) = \text{sign}\,(w_{\text{uncorr}} \cdot x)$, where now $w_{\text{uncorr}} \propto (\alpha'\Psi\alpha + \Sigma)^{-1}\gamma'$. Unlike w_{simple}, the optimal weight vector w_{uncorr} is dense, even if γ is sparse. Consequently, strategies that were effective under the simple model may no longer be ideal.

This has implications for classification analyses of genomic data, since latent variables presumably exist in nearly every genomic data set. For example, in all of the data sets we examine in Section 5, more than half of the variance is captured in the first 10 principal components (see Table S.3 of the supplementary material available at *Biostatistics* online). Even when these latent variables are uncorrelated with the variable of primary interest, their presence will result in an optimal weight vector that is nonsparse. Commonly used strategies such as sparse feature selection will result in many zero feature weights when they are optimally nonzero. Under (2.1), we observe Z but not S. However, observing S would be preferable to observing Z, for the reasons highlighted above. If it were possible to extract S from Z, then classification could be performed on S using usual techniques.

Correlated latent variables

Finally, consider a model in which the latent variables L are correlated with T. We continue to model an observation by $Z = T\gamma + L\alpha + \epsilon$, but now let $L \mid T \sim N(T\eta, \Psi)$, where $\eta \in \mathbb{R}^{1\times r}$. If $\eta = 0$, all r latent variables are uncorrelated with T and this model is equivalent to the uncorrelated latent factor model. Under this model, $Z \mid T \sim N\left(T(\gamma + \eta\alpha), \alpha'\Psi\alpha + \Sigma\right)$. The optimal Bayes classifier is once again linear, but with weights $w_{\text{corr}} \propto (\alpha'\Psi\alpha + \Sigma)^{-1}(\gamma + \eta\alpha)'$. The correlated latent variable model presents the same challenges as the uncorrelated latent variable model. Since the covariance matrix of the class-conditional distribution is nondiagonal, the vector w_{corr} is dense, even if γ is sparse.

However, when the latent variables are correlated with T, they can provide valuable discriminative information, as seen in the form of w_{corr}. The class-conditional means will typically be more separated under the correlated model compared to the uncorrelated model (i.e., $\pm(\gamma + \eta\alpha)$ versus $\pm\gamma$), and classification will generally be easier. If γ is weak and sparse, then the presence of L correlated with T may be especially valuable. As a result, observing S is no longer preferable to observing Z when L and T are correlated, as S no longer contains all the predictive information present in Z.

Discussion

Latent variables present both challenges and opportunities in a high-dimensional classification analysis. In the presence of dense latent signals, the optimal classifier is defined by a dense, p-dimensional vector of weights. Strategies such as feature selection are frequently used to estimate this weight vector, but they may discard relevant information, particularly in the case where the latent variables are correlated with the class label.

Henceforth, we work with the correlated latent factor model, which subsumes the simple and uncorrelated models. If there are no latent factors, then Z is simply S and we can use the simple strategies mentioned above to train a classifier. If L and T are uncorrelated, training a classifier is more complicated, but a possible strategy would be to first isolate S from Z and then train a classifier on S. Such a strategy is reasonable because S contains all of the predictive information available in Z; that is, $(Z \perp\!\!\!\perp T) \mid S$ when L and T are uncorrelated. However, this is no longer true when L and T are correlated, since L now also contains predictive information. While recovering S alone is insufficient under the correlated model, recovering both S and L would be, in the sense that $(Z \perp\!\!\!\perp T) \mid S, L$.

These observations suggest a procedure in practice. Suppose for a moment that L were known. We could use L to estimate α, and then residualize the latent variables from Z to obtain an estimate of S; that is, $\hat{S} = Z - L\hat{\alpha}$. With both $\hat{S}$ and L in hand, we could train separate classifiers on S and L, and then use a metaclassifier to combine them in an ensemble. Of course, L is unobserved under our model and must be estimated in order to carry out this procedure. Fortunately, L can be well-approximated by the top r left singular vectors of Z when γ is sparse and $p \gg n$. In fact, in the limit it is possible to recover L perfectly, up to an arbitrary parameterization (for conditions, see Section S.1 of the supplementary material available at *Biostatistics* online).

Even so, challenges remain. If we train a classifier on $\hat{S}$ and $\hat{L}$, then applying this classifier to out-of-sample observations entails estimating analogous quantities for out-of-sample observations. There is also the question of how to estimate r, the rank of L. Additionally, utilizing an estimate of L to estimate S may lead to downstream overfitting in the classifier trained on $\hat{S}$. The ensemble may suffer as a result. We address these challenges in Section 3, in which we lay out algorithms for training a classifier (the CRC) according to the framework outlined above.

In separating the dense latent signals from the sparse signal of interest, one can view the CRC as performing a kind of forward selection. The former can be viewed as a common effect shared across all features and the latter a feature specific effect (Kneip and Sarda, 2011). By residualizing out $L\alpha$ (or an approximation thereof), we isolate the sparse signals so that we learn which of the p features has an individual effect beyond shared the latent effect. The metaclassifier utilizes this information to improve classification accuracy beyond what can be achieved by a classifier trained on the dense latent signals alone. Because the classifiers in the ensemble are roughly uncorrelated, the utility of the metaclassifier is enhanced.

We now introduce some notation. Let the training set be an i.i.d. sample from the correlated latent factor model. Denote the training observations by $\{(Z_1, T_1), \ldots, (Z_n, T_n)\}$, where $Z_i = T_i\gamma + L_i\alpha + \epsilon_i$ for $i = 1, \ldots, n$. The training data in matrix form is denoted by

$$\boldsymbol{Z}_{n\times p} = [Z_1', \ldots, Z_n']' = \boldsymbol{T}_{n\times 1}\gamma_{1\times p} + \boldsymbol{L}_{n\times r}\alpha_{r\times p} + \boldsymbol{\epsilon}_{n\times p}.$$

We define $\boldsymbol{Z}_{-i}$ to be the $(n-1)\times p$ matrix consisting of all rows in $\boldsymbol{Z}$ except the ith. Finally, denote a generic *target observation*, an out-of-sample observation for which we wish to make a class prediction, by $Z = T\gamma + L\alpha + \epsilon$.

We do not include an intercept term in the model. In practice, we may recenter the columns of $\boldsymbol{Z}$ to have mean 0 and apply the same recentering to Z (see Section S.2 of the supplementary material available at *Biostatistics* online).

3. The CRC

The CRC is an ensemble of two linear discriminant-based classifiers, which we refer to as CRC-L and CRC-S. CRC-L and CRC-S are trained on the dense latent signals and the sparse signal of interest, respectively. A metaclassifier is fit to the resulting discriminant scores to form the ensemble.

3.1. *CRC-L: training a classifier on the latent signals*

One strategy for fitting a classifier on the latent signals is to use principal components linear discriminant analysis (PC-LDA). That is, first estimate $\boldsymbol{L}$ by projecting $\boldsymbol{Z}$ onto the top r principal components of $\boldsymbol{Z}$ and then train an LDA classifier on this estimate. Given that $\boldsymbol{L}$ can be recovered by the principal components (see Section S.1 of the supplementary material available at *Biostatistics* online) and that the class-conditional distribution of $\boldsymbol{L}$ is Gaussian under our model, PC-LDA is a natural strategy. One challenge, however, is that r is unknown and must be estimated. If $\hat{r} < r$, then we may fail to capture all relevant information in $\boldsymbol{L}$. On the other hand, if $\hat{r} > r$, then the estimate $\hat{\boldsymbol{L}}$ will overfit the training data, potentially leading to downstream overfitting in the ensemble. In general, estimating r is a challenging task (Choi *and others*, 2017; Jolliffe, 2002).

We take an alternative approach which does not require an estimate of r or even an explicit estimate of $\boldsymbol{L}$. Specifically, we project $\boldsymbol{Z}$ onto all principal components of $\boldsymbol{Z}$ and then use LDA to train a classifier on the projected data (details in Section S.3 of the supplementary material available at *Biostatistics* online). This approach amounts to PC-LDA using all n principal components, which has good out-of-sample performance despite overfitting the training data. To see this, consider the regression analogue of PC-LDA, principal components regression (PCR). PCR and ridge regression are well-known to be closely related (Jolliffe, 2002; Friedman *and others*, 2001) and are in fact equivalent in a special case. If n principal components are used, PCR is equivalent to ridge regression with the ridge penalty tending to zero, a technique more commonly known as ridgeless regression (Hastie *and others*, 2019). Ridgeless regression interpolates the training data, resulting in perfect overfitting. However, it has been shown to have good out-of-sample predictive accuracy when $p \gg n$ (Hastie *and others*, 2019). This property motivates our approach. So long as we apply CRC-L to out-of-sample observations, we avoid estimating r without suffering too much from the negative effects of overfitting. Overfitting in the training data remains an issue when fitting the ensemble, but this issue is easily addressed via a leave-one-out approach (Section 3.3).

Comment: In this article, we focus on the case in which γ is sparse. The characterization of CRC-L as a classifier trained on the latent variables $\boldsymbol{L}$ changes if γ is not sparse, but this does not necessarily impact performance. For further comments, see Section S.1 of the supplementary material available at *Biostatistics* online.

3.2. *CRC-S: training a classifier on the signal of primary interest*

We next build a classifier on $\boldsymbol{S}$. Diagonal LDA (DLDA) is a natural choice for this classifier, since S is normal with diagonal covariance. Since neither $\boldsymbol{S}$ nor S is known, the first step in training CRC-S is to obtain estimates $\hat{\boldsymbol{S}}$ and $\hat{S}$. We propose two algorithms for doing so: residualization to obtain $\hat{S}$ and cross-residualization to obtain $\hat{\boldsymbol{S}}$.

3.2.1. *Residualization* Recovering S for an out-of-sample observation is one of the key challenges of our approach. Even if $\boldsymbol{S}$ were known in the training data, a classifier trained on $\boldsymbol{S}$ would not be useful unless we could also recover S for an out-of-sample observation Z. Here, we focus on recovering S.

Suppose that L were observed (for both the training and target observations). Additionally, suppose γ were known. In this case, we could recover S in a straightforward manner. Since $\boldsymbol{Z} - \boldsymbol{T}\gamma = \boldsymbol{L}\alpha + \epsilon$, an estimate of α could be obtained by regressing $\boldsymbol{Z} - \boldsymbol{T}\gamma$ onto $\boldsymbol{L}$. Having obtained $\hat{\alpha}$, we could residualize the latent variables from Z to obtain an estimate of S; that is, $\hat{S} = Z - L\hat{\alpha}$.

However, since neither the latent variables nor γ are known this approach must be modified. With regard to the latent variables, one option would be to estimate $\boldsymbol{L}$ and L directly by projecting $\boldsymbol{Z}$ and Z onto the first several principal components of $\boldsymbol{Z}$. We could then replace $\boldsymbol{L}$ and L by their estimates in the procedure described above, and let $\hat{S} = Z - \hat{L}(\hat{\boldsymbol{L}}'\hat{\boldsymbol{L}})^{-1}\hat{\boldsymbol{L}}'(\boldsymbol{Z} - \boldsymbol{T}\gamma)$. Since $\hat{\boldsymbol{L}}$ and $\hat{L}$ are obtained by projecting $\boldsymbol{Z}$ and Z onto the principal components of $\boldsymbol{Z}$, this may be viewed as performing a PCR in which the training set predictors are $\boldsymbol{Z}$, the training set response variables are $\boldsymbol{Z} - \boldsymbol{T}\gamma$, and the resulting fitted model is applied to Z in order to predict $L\alpha$; the predicted $L\alpha$ is then subtracted from Z to give $\hat{S}$. As discussed in the previous section, one challenge of PCR is selecting the number of principal components to use. However, when $p \gg n$ good out-of-sample performance may be achieved by simply using all n principal components. As in the previous section, we adopt this approach both because it avoids the need to obtain an explicit estimate of r and because it provides, after some algebra, a simple closed-form solution

$$\hat{S} = Z - Z\boldsymbol{Z}'(\boldsymbol{Z}\boldsymbol{Z}')^{-1}(\boldsymbol{Z} - \boldsymbol{T}\hat{\gamma}), \tag{3.2}$$

where $\hat{\gamma}$ is an estimate of γ (discussed below). Another advantage of using all n principal components is that it closely parallels the approach taken in CRC-L. This is advantageous because our ultimate goal in computing $\hat{S}$ is to obtain any residual predictive information in Z that is not already captured by CRC-L.

With regard to estimating γ, several options are available. Importantly, any estimate of γ must take into account the presence of the latent variables. In particular, simply taking the difference in class means, i.e., letting $\hat{\gamma} = (\boldsymbol{T}'\boldsymbol{T})^{-1}\boldsymbol{T}'\boldsymbol{Z}$, is not a viable option because this estimate will be biased if $\boldsymbol{L}$ is correlated with $\boldsymbol{T}$. Estimators that account for $\boldsymbol{L}$ can be found in the batch effects normalization literature. The estimator we use, $\hat{\gamma} = [\boldsymbol{T}'(\boldsymbol{Z}\boldsymbol{Z}')^{-1}\boldsymbol{T}]^{-1}\boldsymbol{T}'(\boldsymbol{Z}\boldsymbol{Z}')^{-1}\boldsymbol{Z}$, is from Gagnon-Bartsch *and others* (2013) and has the advantage of having a simple closed-form expression. This estimator may be seen as approximating the ordinary least squares estimate of γ from a regression in which $\boldsymbol{L}$ is known (Section S.4 of the supplementary material available at *Biostatistics* online). Other estimates of γ, similar in spirit, include those obtained via surrogate variables analysis (Leek and Storey, 2007, 2008), the confounder adjusted testing and estimation framework (Wang *and others*, 2017), as well as several others (Listgarten *and others*, 2010; Gagnon-Bartsch and Speed, 2012; Sun *and others*, 2012; Gerard and Stephens, 2021).

3.2.2. *Cross-residualization* In addition to recovering S for an out-of-sample observation, we also wish to recover $\boldsymbol{S}$ in the training data. A natural approach would be simply to apply the residualization procedure to the training data as well. A complication is that the residualization procedure uses PCR with all n principal components, and although this works well for out-of-sample observations, it massively overfits the training data. Indeed, substituting $\boldsymbol{Z}$ for Z in (3.2) to obtain $\hat{\boldsymbol{S}}$ would yield $\hat{\boldsymbol{S}} = \boldsymbol{Z} - \boldsymbol{Z}\boldsymbol{Z}'(\boldsymbol{Z}\boldsymbol{Z}')^{-1}(\boldsymbol{Z} - \boldsymbol{T}\hat{\gamma}) = \boldsymbol{T}\hat{\gamma}$, a rank one matrix in which each column (predictor) is perfectly correlated with $\boldsymbol{T}$. This may be interpreted as the PCR overfitting to the $\boldsymbol{\epsilon}$ term in the training data, and therefore residualizing out the $\boldsymbol{\epsilon}$ term along with $\boldsymbol{L}\alpha$.

However, it is necessary to preserve the $\boldsymbol{\epsilon}$ term in order to properly train a classifier. For example, in the DLDA classifier that we will fit, it is necessary to preserve the $\boldsymbol{\epsilon}$ term in order to estimate the individual feature variances $\sigma_1^2, \ldots, \sigma_p^2$. More generally, overfitting to the training data introduces an asymmetry between the predictors $\hat{\boldsymbol{S}}$ to which the classifier is trained and the predictors $\hat{S}$ to which the classifier is applied. As a result of this asymmetry, we would not expect a classifier trained on $\hat{\boldsymbol{S}}$ to generalize well to $\hat{S}$.

We address this problem by applying the residualization procedure to the training data in a leave-one-out manner. For each $i \in \{1, \ldots, n\}$ we let

$$\hat{S}_i = Z_i - Z_i\boldsymbol{Z}'_{-i}(\boldsymbol{Z}_{-i}\boldsymbol{Z}'_{-i})^{-1}(\boldsymbol{Z}_{-i} - \boldsymbol{T}_{-i}\hat{\gamma}^{(i)}), \tag{3.3}$$

where $\hat{\gamma}^{(i)} = [\boldsymbol{T}'_{-i}(\boldsymbol{Z}_{-i}\boldsymbol{Z}'_{-i})^{-1}\boldsymbol{T}_{-i}]^{-1}\boldsymbol{T}'_{-i}(\boldsymbol{Z}_{-i}\boldsymbol{Z}'_{-i})^{-1}\boldsymbol{Z}_{-i}$. We then let $\hat{\boldsymbol{S}} = [\hat{S}'_1, \ldots, \hat{S}'_n]'$. By computing $\hat{\boldsymbol{S}}$ in this leave-one-out manner, we put $\hat{\boldsymbol{S}}$ and $\hat{S}$ on equal footing; each row of $\hat{\boldsymbol{S}}$ is computed in a manner analogous to the manner in which $\hat{S}$ is computed. As a result, we can expect a classifier trained on $\hat{\boldsymbol{S}}$ to generalize well to $\hat{S}$. We refer to this leave-one-out approach as *cross-residualization*.

Comment. An alternative strategy to address overfitting in the training data would be to explicitly estimate r when performing the PCR in the residualization procedure. If $\hat{r} \ll n$, this would reduce the overfitting. However, we prefer the cross-residualization strategy for several reasons. First, as previously noted, estimating r is a challenging problem. Second, for any $\hat{r}$ we would expect at least some degree of overfitting to the training data, which would introduce asymmetry between the training and target data, thereby impacting the ability of the classifier to generalize to the target data. Finally, we note that because (3.3) has a simple closed-form solution, and in particular because $(\boldsymbol{Z}_{-i}\boldsymbol{Z}'_{-i})^{-1}$ can be computed using a rank-one downdate, cross-residualization can be implemented in a computationally efficient manner. Thus, cross-residualization provides an approach that does not require selecting a tuning parameter, is computationally efficient, and guarantees that the training predictors $\hat{\boldsymbol{S}}$ and target predictors $\hat{S}$ are on equal footing.

3.2.3. *DLDA* Cross-residualization effectively removes the sources of variation common across all features, and the columns of $\hat{\boldsymbol{S}}$ are therefore approximately decorrelated. As a result, we can take advantage of this decorrelation and fit a classifier to $\hat{\boldsymbol{S}}$ using DLDA. When fitting by DLDA, feature selection is an important consideration. Because cross-residualization effectively removes the sources of variation common across all features, we wish to select features based on their individual predictive effects beyond the common latent effect. Selecting too many null features may render the ensemble step less useful. Since the columns of $\hat{\boldsymbol{S}}$ are approximately decorrelated, we utilize a simple marginal screening scheme in which only the features with the N smallest p-values are used for classification (i.e., the "top" N features). Here, $N \in \{1, \ldots, p\}$ is a tuning parameter that is selected via a grid search. For details, see Section S.5 of the supplementary material available at *Biostatistics* online.

3.3. *Fitting the ensemble*

There are many ways in which we might combine CRC-S and CRC-L into an ensemble. Voting is a popular approach (Yang *and others*, 2010), but here there are only two classifiers to be combined. Instead, we adopt elements of the super learning approach of Polley *and others* (2011). In the super learner, the training data are split into several folds, and each classifier in the ensemble is fit to each fold. The resulting fits are used to estimate the optimal weighted combination of classifiers. These estimated weights minimize cross-validated error over all possible weighted linear combinations of the classifiers. The ensemble classifier is then obtained by fitting each of the classifiers to the entire training set and combining them with the estimated weights.

The CRC is well-suited to the super learning approach since cross-residualization is already a leave-one-out procedure. To fit the CRC ensemble, we score the training observations using CRC-S and CRC-L in a leave-one-out manner and then train a metaclassifier on these scores. Under our model, these scores are Gaussian conditional on the class label, and we therefore use LDA for the metaclassifier. Assembling all steps, we arrive at Algorithm 1, which summarizes the CRC.

An alternative interpretation of the CRC arises if we unpack Algorithm 1 and examine how the CRC acts on Z instead of the discriminant scores s^s and s^l. Note that s^s and s^l arise as linear functions of Z. For s^l, this is clear. For s^s, observe that (3.2) may be rewritten as $\hat{S} = Z[I - \boldsymbol{Z}'(\boldsymbol{Z}\boldsymbol{Z}')^{-1}(\boldsymbol{Z} - \boldsymbol{T}\hat{\gamma})]$ and hence $s^s = Z[I - \boldsymbol{Z}'(\boldsymbol{Z}\boldsymbol{Z}')^{-1}(\boldsymbol{Z} - \boldsymbol{T}\hat{\gamma})]w^s$ is a linear function of Z. In addition, $\hat{c}$ is itself a linear classifier and can be expressed as $\hat{c}(s^l, s^s) = \text{sign}\left(b_1 s^s + b_2 s^l\right)$ where b_1 and b_2 are scalar weights. Therefore, $\hat{c}(s^s, s^l) = \text{sign}\,(Zw)$ where $w = b_1[I - \boldsymbol{Z}'(\boldsymbol{Z}\boldsymbol{Z}')^{-1}(\boldsymbol{Z} - \boldsymbol{T}\hat{\gamma})]w^s + b_2 w^l$. The expression for w shows that the CRC is itself a linear classifier that weights each feature of an observation by a weighted average of the CRC-S weights (meant to capture the sparse signals) and CRC-L weights (meant to capture the dense, low-rank latent signals), where the relative weighting is determined by the relative predictive value of the individual classifiers. Features may receive large weights due to the discriminative information they carry about the sparse signals, the latent signals, or both.

4. Simulations

We perform several simulations to illustrate the inner workings of the CRC and to compare the CRC to other classifiers which are frequently used in genomic applications.

We generate data according to the simple, uncorrelated, and correlated models from Section 2. For all three models, we let $\Sigma = I_p$ and $\gamma = (\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, 0, \ldots, 0)$. For the uncorrelated and correlated models, we set $r = 3$ and $\Psi = I_{r \times r}$, and generate i.i.d. α_{ij} from a standard normal distribution. In the correlated

Algorithm 1: Cross-residualization classifier (CRC)

```
Data: $\boldsymbol{Z}_{n\times p}, \boldsymbol{T}_{n\times 1}, Z_{1\times p}$
Result: $\hat{T} \in \{\pm 1\}$
Residualization
    Data: $\boldsymbol{Z}_{n\times p}, \boldsymbol{T}_{n\times 1}, Z_{1\times p}$
    Result: $\hat{S}_{1\times p}$
    $\hat{\gamma} \leftarrow [\boldsymbol{T}'(\boldsymbol{Z}\boldsymbol{Z}')^{-1}\boldsymbol{T}]^{-1}\boldsymbol{T}'(\boldsymbol{Z}\boldsymbol{Z}')^{-1}\boldsymbol{Z}$
    $\widehat{L\alpha} \leftarrow Z\boldsymbol{Z}'(\boldsymbol{Z}\boldsymbol{Z}')^{-1}(\boldsymbol{Z} - \boldsymbol{T}\hat{\gamma})$
    return $\hat{S} \leftarrow Z - \widehat{L\alpha}$
Cross-residualization
    Data: $\boldsymbol{Z}_{n\times p}, \boldsymbol{T}_{n\times 1}$
    Result: $\hat{\boldsymbol{S}}_{n\times p}$
    for $i$ in 1, ..., $n$ do
        $\hat{S}_i \leftarrow$ result of residualization algorithm applied to $(\boldsymbol{Z}_{-i}, \boldsymbol{T}_{-i}, Z_i)$
    end
    return $\hat{\boldsymbol{S}} \leftarrow [\hat{S}_1', \ldots, \hat{S}_n']'$
CRC-L
    Data: $\boldsymbol{Z}_{n\times p}, \boldsymbol{T}_{n\times 1}, Z_{1\times p}$
    Result: $s^l \in \mathbb{R}$
    Use PC-LDA (with $n$ principal components) to train a classifier on $(\boldsymbol{Z}, \boldsymbol{T})$, resulting in weight
      vector $w^l$
    $s^l \leftarrow Zw^l$
    return $s^l$
CRC-S
    Data: $\hat{\boldsymbol{S}}_{n\times p}, \boldsymbol{T}_{n\times 1}, \hat{S}_{1\times p}$
    Result: $s^s \in \mathbb{R}$
    Use DLDA to train a classifier on $(\hat{\boldsymbol{S}}, \boldsymbol{T})$, resulting in weight vector $w^s$
    $s^s \leftarrow \hat{S}w^s$
    return $s^s$
Ensemble classifier
    Data: $\boldsymbol{Z}_{n\times p}, \hat{\boldsymbol{S}}_{n\times p}, \boldsymbol{T}_{n\times p}, s^l, s^s$
    Result: $\hat{T} \in \{\pm 1\}$
    for $i$ in 1, ..., $n$ do
        $s_i^l \leftarrow$ result of CRC-L applied to $(\boldsymbol{Z}_{-i}, \boldsymbol{T}_{-i}, Z_i)$
        $s_i^s \leftarrow$ result of CRC-S applied to $(\hat{\boldsymbol{S}}_{-i}, \boldsymbol{T}_{-i}, \hat{S}_i)$
    end
    Use LDA to train a classifier $\hat{c}$ on predictors $\{(s_i^l, s_i^s)\}_{i=1}^n$, and response $\boldsymbol{T}$
    return $\hat{c}(s^l, s^s)$
```

Notes: Some details are omitted for clarity. In particular, CRC-S has a feature selection step, and the implementations of both CRC-L and CRC-S differ slightly when applied to $n-1$ observations rather than n. See Sections S.3 and S.5 of the supplementary material available at *Biostatistics* online.

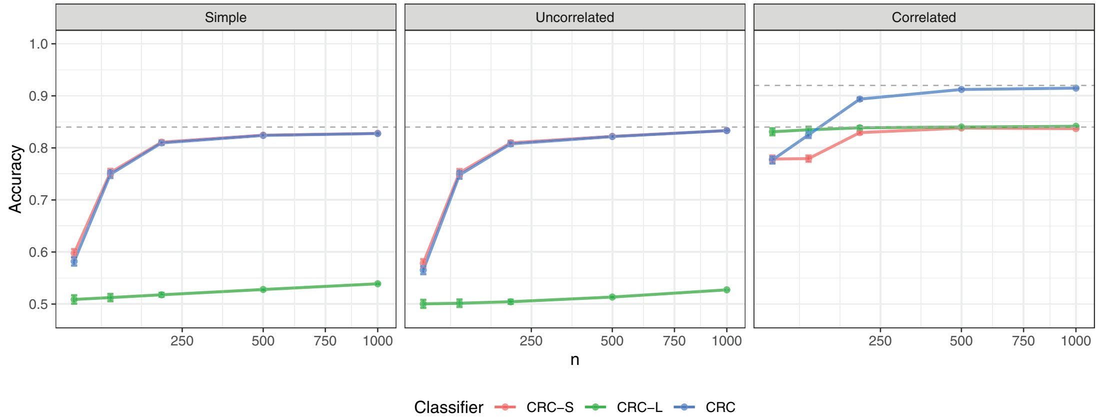

Fig. 1. Mean accuracies of CRC, CRC-S, and CRC-L when $p = 100\,000$. Sample size n is depicted on a square-root scale. The dashed horizontal lines are at $\Phi(1)$ and $\Phi(\sqrt{2})$ and indicate Bayes optimal accuracy rates (as detailed in the text).

model, we additionally set $\eta = (\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}})$. Note that the Bayes optimal accuracy rate is $\Phi(\sqrt{\gamma'\gamma})$ for a classifier trained on S and $\Phi(\sqrt{\eta'\eta})$ for a classifier trained on L. For a classifier trained on (S, L), the Bayes optimal accuracy rate is $\Phi(\sqrt{\gamma'\gamma + \eta'\eta})$. Thus with our parameter choices, the Bayes optimal accuracy rate for a classifier trained on (S, L) is $\Phi(1)$ under the simple model, $\Phi(1)$ under the uncorrelated model, and $\Phi(\sqrt{2})$ under the correlated model.

For each model, we generate a balanced class label vector of dimension n and a feature matrix of dimension $n \times p$, for varying n and p. We let n range from 50 to 1000 and consider $p = 20\,000, 100\,000$, and $500\,000$, roughly corresponding to the number of features found in various types of “omics” data sets. For each n and p, we replicate this procedure several times and average accuracy rates for CRC, CRC-S, and CRC-L across replications. For computational feasibility, we vary the number of replications with n (see Section S.6 of the supplementary material available at *Biostatistics* online). Results for $p = 100\,000$ are depicted in Figure 1; results for $p = 20\,000$ and $p = 500\,000$ can be found in Section S.6 of the supplementary material available at *Biostatistics* online.

To see how the CRC ensemble works to improve classification accuracy, we first look at the performance of its component classifiers, CRC-L and CRC-S. CRC-L has an accuracy rate close to 50% under the simple and uncorrelated models, which can be attributed to the sparsity of γ and the additional noise contributed by the latent variables in the uncorrelated case. The accuracy of CRC-L increases slightly as n increases, as is expected. Although CRC-L performs poorly under the simple and uncorrelated models, it has accuracy close to the Bayes optimal rate (for L) under the correlated model, even when n is relatively small. Looking at the other component classifier, we see that CRC-S performs well across all models. For large n, CRC-S has accuracy close to the Bayes optimal rate for S. In particular, CRC-S exhibits similar performance under the simple and uncorrelated models, suggesting that cross-residualization is providing a reasonable estimate of $\boldsymbol{S}$, allowing CRC-S to effectively pick up on the sparse signal.

CRC behaves as expected in the simple and uncorrelated cases, with accuracies on par with CRC-S. However, it is the correlated case that is particularly illustrative. For large n, CRC-S and CRC-L have accuracies close to the Bayes accuracy rates for S and L, respectively, but both of these accuracies are less than the Bayes accuracy rate for (S, L). The accuracy of the CRC ensemble, however, is close to the Bayes accuracy rate for (S, L) for large n, suggesting that the CRC is making more efficient use of the signal in the data by considering the sparse signals and dense latent signals separately. Whereas the uncorrelated

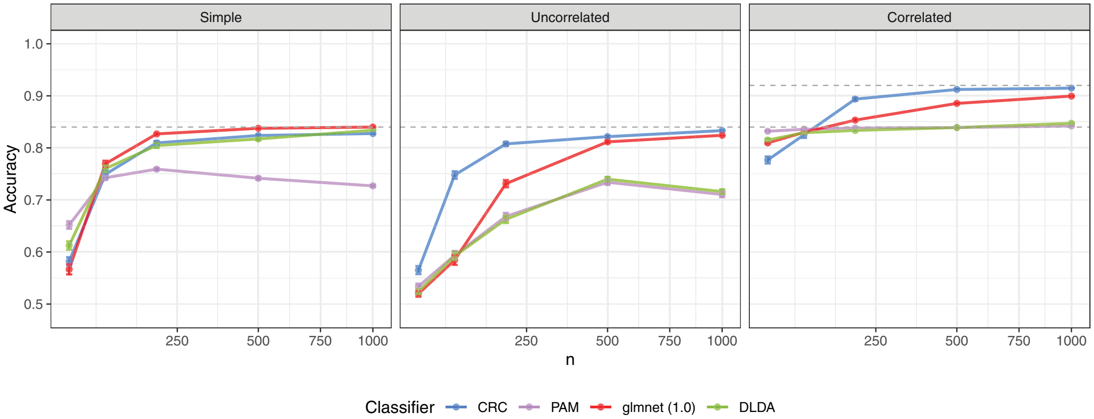

Fig. 2. Mean accuracies of glmnet, PAM, DLDA, and CRC when $p = 100\,000$. Sample size n is depicted on a square-root scale. The dashed horizontal lines are at $\Phi(1)$ and $\Phi(\sqrt{2})$ and indicate Bayes optimal accuracy rates (as detailed in the text).

case demonstrates that improved recovery of the sparse signals can improve classification accuracy, the correlated case demonstrates that there are potential information gains to be made beyond better sparse recovery when dense latent signals exist.

In Figure 2, we repeat the simulation with other classifiers commonly used in genomics: penalized logistic regression (glmnet) (Zou and Hastie, 2005), DLDA (Dudoit and Fridlyand, 2003), and nearest shrunken centroids (PAM) (Tibshirani *and others*, 2002). These classifiers are all linear. In particular, DLDA and PAM are discriminant-based, like the CRC. Both glmnet and PAM have shrinkage parameters which are tuned via cross-validation. The glmnet classifier has an additional parameter, the elastic net mixing parameter α. We set α to various values from 0 to 1 since simulation results are potentially sensitive to the value of this parameter. For visual simplicity, we include only the results for $\alpha = 1$ in Figure 2; this is the value of α for which glmnet generally performs best. Results for glmnet at other values of α can be found in Section S.6 of the supplementary material available at *Biostatistics* online. We use our own implementation of DLDA, which includes a feature selection step that is identical to the feature selection step in CRC-S. As a result, DLDA and CRC-S differ only in that DLDA is fit to $\boldsymbol{Z}$ whereas CRC-S is fit to $\hat{\boldsymbol{S}}$.

Figure 2 displays the results. Under the simple model, the classifiers perform similarly, with the exception of PAM, for which accuracy appears to degrade as the sample size increases. We believe that this is related to PAM selecting a large number of features in the simple setting (see Table S.1 of the supplementary material available at *Biostatistics* online). Under the uncorrelated and correlated models, however, the CRC appears to offer substantial gains over the other methods, particularly for moderate sample sizes. PAM and DLDA perform similarly under the correlated and uncorrelated models, which is perhaps expected since PAM is essentially DLDA with an ℓ_1 penalty to perform feature selection.

It is interesting to compare CRC-S and DLDA under the three models. Recall that in our simulation, the two methods differ only in that DLDA is fit to $\boldsymbol{Z}$, whereas CRC-S is fit to $\hat{\boldsymbol{S}}$. As expected, DLDA behaves much like CRC-S under the simple model. However, this is no longer the case when latent factors are present. Under the uncorrelated model, DLDA is outperformed by CRC-S despite both classifiers selecting a similarly small number of features (see Table S.1 of the supplementary material available at *Biostatistics* online), suggesting that feature selection alone is not necessarily effective when the optimal feature weights are nonsparse. As noted in Section 2, sparsity in S does not necessarily imply sparsity in

Table 1. *Data sets. In the "Classes" column, the number of samples in each class is given in parentheses. The Alzheimer's and Sepsis data sets contain multiple samples per subject (individual person), so we additionally give the number of subjects. For details, see Section S.7 of the supplementary material available at Biostatistics online.*

Name	Type	Platform	n	p	Classes
Alzheimer's	Methyl.	HM 450k	190	485 512	Disease (106; 54), Control (84; 42)
Asthma	Methyl.	HM 450k	115	485 512	Asthma (74), Control (41)
Colorectal Cancer	Expr.	U133 Plus 2.0	200	54 675	Cancer (100), Normal (100)
Crohn's (Methyl.)	Methyl.	EPIC	238	504 790	CD (164), Non-IBD Control (74)
Crohn's (Expr.)	Expr.	HiSeq 2000	304	13 151	CD (254), Non-IBD (50)
FASD	Methyl.	HM 450k	103	485 512	FASD (39), Control (64)
Sepsis	Expr.	HiSeq 2500	217	27 670	Healthy (58; 20), Critically ill nonseptic (63; 22), Sepsis (96; 29)

Z, even if L and T are uncorrelated. Under the correlated model, DLDA performs similarly to CRC-S, but both are outperformed by the CRC ensemble, highlighting the effectiveness of cross-residualization when used in conjunction with an ensemble strategy.

In Section S.9 of the supplementary material available at *Biostatistics* online, we examine the robustness of the CRC to violations of our model, including sparse α, nonnormally distributed latent variables, and nonnormal errors. In general, we find the CRC to be robust to violations of our model, an encouraging conclusion for a method we hope to be of use in a wide variety of genomic applications.

5. APPLICATIONS TO GENOMIC DATA

We apply the CRC to a variety of publicly available genomic data sets. These data sets cover a broad range of phenotypes, including Alzheimer's disease, asthma, cancer, Crohn's disease, fetal alcohol spectrum disorder (FASD), and sepsis (for details, see Section S.7 of the supplementary material available at *Biostatistics* online). The data sets also encompass several technologies and include gene expression data from multiple platforms (Illumina HiSeq 2000 and 2500, and Affymetrix HG U133 Plus 2.0 microarray) as well as methylation data from two platforms (Illumina Infinium 450K and EPIC BeadChip arrays). See Table 1 for a summary, and Section S.7 of the supplementary material available at *Biostatistics* online for additional details. We select these particular data sets because they have a relatively large number of samples within each class and a clearly defined phenotype.

We compare the CRC to the same classifiers from Section 4. For each data set, we assess the performance of each classifier as follows. Let $n_0 = \min\{n_1, n_2\}$, where n_1, n_2 are the class sizes (number of observations in each class). To form the training set, we randomly sample $\lfloor 0.8 \cdot n_0 \rfloor$ observations from each class. Of the remaining observations, we sample $n_0 - \lfloor 0.8 \cdot n_0 \rfloor$ from each class to form the test set. This results in balanced training and test sets, so that the baseline accuracy rate is 50% across data sets and interpretation of accuracies is not complicated by class imbalances. We create 200 train-test splits in this way and compute the mean test accuracy of each classifier across replications (Table 2). For the Alzheimer's and Sepsis data sets, we modify the procedure slightly to account for the fact that there are multiple observations per individual; see Section S.8 of the supplementary material available at *Biostatistics* online.

The CRC generally performs well relative to other classifiers. CRC-L generally performs well across the data sets, suggesting that there may be dense latent factors that are predictive of the class labels. CRC-S generally outperforms DLDA, highlighting the value of (cross-)residualization, and suggesting

Table 2. *Results on various genomic data sets. Boldface denotes the highest accuracy rate(s). Standard errors are all less than 0.004.*

Data set	glmnet	PAM	DLDA	CRC	CRC-S	CRC-L
Alzheimer's	0.74	0.72	0.70	**0.78**	0.78	0.79
Asthma	0.75	**0.78**	0.75	0.77	0.76	0.76
Colorectal cancer	0.91	0.74	0.76	**0.94**	0.94	0.88
Crohn's (Expr.)	**0.91**	0.86	0.88	**0.91**	0.88	0.91
Crohn's (Methyl.)	0.87	0.86	0.85	**0.88**	0.88	0.80
FASD	0.77	0.79	0.76	**0.84**	0.84	0.78
Sepsis (Healthy vs. Crit-Ill)	0.85	0.79	0.80	**0.90**	0.89	0.84
Sepsis (Sepsis vs. Crit-Ill)	0.70	0.70	0.69	**0.77**	0.76	0.77
Sepsis (Sepsis vs. Healthy)	0.94	0.91	0.92	**0.95**	0.95	0.94

once again that the signal of primary interest may be sparse in S but dense in Z. As with the simulations of Section 4, the relative performance of CRC-S and DLDA on these data sets suggests that feature selection in conjunction with cross-residualization can be more effective than feature selection alone when dense latent variation is present. Overall, the CRC generally performs as well as the better of its two component classifiers.

Another promising result is that the CRC appears to perform well within subgroups of interest. In the Alzheimer's data set, there are four types of samples: purified glia cells, purified neuron cells, bulk tissue taken from the temporal cortex, and bulk tissue taken from the frontal cortex. In Section S.8 of the supplementary material available at *Biostatistics* online, we report accuracy rates within each subgroup for the analysis performed in this section. Notably, the CRC performs as well or better than the other classifiers within each cell type. For certain cell types and classification tasks, there is a marked improvement in accuracy rates. We see similar results in the Sepsis data set, in which samples occur across three cell types (CD4, CD14, and CD8).

We also investigate the performance of other classifiers including k-nearest neighbors (kNN), random forest, and support vector machines (SVM) on these data sets. These simulation results can be found in Section S.11 of the supplementary material available at *Biostatistics* online. Compared to the additional methods we examined, the CRC achieved the highest accuracy rate across most of the data sets, although SVM with a linear kernel performed as well as the CRC on several of the data sets.

6. Discussion

The CRC ensemble derives its strength from the separation of the sparse signal of interest and dense latent signals. While ensembles in general typically offer some benefit by combining multiple classifiers, the CRC fully exploits this benefit by combining classifiers trained on different sources of information. In particular, we are able to better leverage information contained in the sparse signal, which may be obscured by the dense latent signals.

In addition to its statistical performance, the CRC offers several practical advantages, many of which stem from its modularity. There are four individual prediction algorithms used within the CRC—the PC-LDA in CRC-L, the PCR in the residualization step, the DLDA in CRC-S, and the LDA that defines the ensemble. There are several distinct benefits to this modular structure. In particular, modularity makes it easier to conceptualize an approach as well as easier to implement and debug (Gerard and Stephens, 2021). In addition, modularity produces additional opportunities for diagnostics, as there are intermediate outputs that can be inspected and visualized. Finally, because individual algorithms are easily swapped out, the CRC can be readily refined and adapted to new settings.

For example, we may wish to modify CRC-L and replace PC-LDA with some other algorithm if there is reason to believe that the latent variables are not normally distributed. This might occur, for example, if there are subgroups within a class (e.g., subtypes of a cancer). In this case, it may be desirable to fit CRC-L using PC-kNN or some other nonparametric method. To maintain the complementary nature of CRC-S and CRC-L, we might choose to use PC-kNN within the residualization and cross-residualization algorithms as well, so that the same signals captured by CRC-L are those residualized out of Z.

Improvements and adaptations of CRC-S are also possible. For example, we could do away with a marginal screening procedure for feature selection, and instead, utilize shrinkage penalties. We believe that such alternatives have the potential to improve the accuracy of CRC-S; the fact that glmnet outperforms CRC-S in the simple simulation (left panel of Figure 2) suggests that there is indeed room for improvement. In addition, CRC-S could be generalized to accommodate more general error structures. For example, even after removing the latent variables, correlations may remain between a few individual genes whose functions are tightly related. In such cases, it may make sense to model Σ as a sparse but not strictly diagonal matrix.

Finally, neither residualization nor cross-residualization require T to be a binary variable. By taking T to be continuous, we can extend CRC to the regression setting, provided that we modify CRC-S and CRC-L accordingly (i.e., substituting regression-based equivalents). Again, the modularity of the CRC makes this relatively straightforward from both a conceptual and practical standpoint.

The specific algorithms that we have chosen to use in the CRC also offer practical advantages of their own. In particular, our implementations of PC-LDA and PCR use all n principal components and therefore do not require the selection of a tuning parameter. The resulting methods are computationally efficient and have closed-form solutions. In using all n principal components, we take advantage of the fact that PCR and PC-LDA continue to have good out-of-sample predictions despite severely overfitting to the training data. This strategy is made possible by the leave-one-out manner in which we apply these algorithms, which allows us to avoid issues that would otherwise arise due to the overfitting of the training data. In particular, the leave-one-out nature of cross-residualization allows us to preserve the ϵ term, which is critical to properly fitting CRC-S. In addition, the leave-one-out manner in which we obtain the CRC-L and CRC-S scores allows us to properly weight CRC-L and CRC-S in the ensemble. The net effect of this approach is that we replace tuning with over-parameterization and leave-one-out fits; we are hopeful that this general strategy may be useful in other contexts as well. The leave-one-out fits within the ensemble provide another benefit; namely, built-in estimates of error rates for the individual components and the CRC as a whole.

Regardless of the specific implementation, we believe the analyses in this paper highlight the importance of accounting for the latent variables that are prevalent in genomic data. By doing so, weaker biological signals which may be less prominent but equally as important can be better incorporated into analyses of such data.

Finally, we note that we have assumed throughout this article that the data generating distribution is stable, that is, that the distribution of (Z, T) is the same in the training and target sets. This stability assumption is typical and underlies many commonly used statistical prediction methods. Moreover, this stability assumption is reasonable in many genomic applications, at least in cases where the classifier is applied to a target population that is representative of the training population. However, there are important cases in which this assumption is questionable.

Consider, for example, batch effects. In a well-designed study, some care may be taken to ensure that batch is not correlated with biology (e.g., by randomizing the run order of the samples), but in practice, this may not always be feasible and batch may indeed end up being correlated with biology. Moreover, the correlation between batch and biology may differ between the training and target sets, resulting in distribution shift, i.e., the joint distribution of (Z, T) will differ in the training and target sets. Such distribution shift presents a major challenge; efforts to address it include transfer learning approaches (Pan and Yang, 2010; Weiss *and others*, 2016) and normalization approaches (McCall *and others*, 2010; Parker *and others*, 2014).

We believe the CRC may also offer additional opportunities to address the issue of nonstability. In particular, we suspect that by separating the dense signals in L from the sparse signals in S we will often incidentally separate nonstable signals from stable ones as well. More specifically, we suspect that nonstable signals such as those arising from batch effects or differences between the training and target populations will typically impact a large number of features and therefore appear in L. Conversely, we suspect that the sparse signals that appear in S will typically have a biological origin directly related to the phenotype of interest and will therefore be largely stable. The resulting separation would only be partial, however—although S might be mostly stable, L would presumably contain both stable and nonstable signals. Additional strategies will therefore be needed to further decompose L into stable and nonstable components, or to otherwise accommodate potential distribution shift in L.

7. Software and code

An R package implementing the CRC, as well as code for reproducing the results in this article, is publicly available at https://github.com/yujiap/crc_code.

Supplementary material

Supplementary material is available at http://biostatistics.oxfordjournals.org.

Acknowledgments

The authors would like to thank Zhihao Guo, Greg Hunt, Kristen Hunter, Dan Kessler, Zoe Rehnberg, Kerby Shedden, Terry Speed, and Jonathan Terhorst for helpful comments.

Conflict of Interest: None declared.

Funding

N.Y.P. was supported by a National Science Foundation Graduate Research Fellowship (DGE 1841052) and National Science Foundation RTG grant (DMS 1646108).

References

Bickel, P. J. and Levina, E. (2004). Some theory for Fisher's linear discriminant function, 'naive Bayes', and some alternatives when there are many more variables than observations. *Bernoulli* **10**, 989–1010.

Bind, M., Lepeule, J., Zanobetti, A., Gasparrini, A., Baccarelli, A. A., Coull, B. A., Tarantini, L., Vokonas, P. S., Koutrakis, P. and Schwartz, J. (2014). Air pollution and gene-specific methylation in the Normative Aging Study: Association, effect modification, and mediation analysis. *Epigenetics* **9**, 448–458.

Boyle, E. A., Li, Y. I. and Pritchard, J. K. (2017). An expanded view of complex traits: From polygenic to omnigenic. *Cell* **169**, 1177–1186.

Choi, Y., Taylor, J. and Tibshirani, R. (2017). Selecting the number of principal components: Estimation of the true rank of a noisy matrix. *Annals of Statistics* **45**, 2590–2617.

Cook, R. D., Forzani, L. and Rothman, A. J. (2012). Estimating sufficient reductions of the predictors in abundant high-dimensional regressions. *Annals of Statistics* **40**, 353–384.

Dicker, L. (2012). Optimal estimation and prediction for dense signals in high-dimensional linear models. *arXiv:1203.4572*.

Dobriban, E. and Wager, S. (2018). High-dimensional asymptotics of prediction: Ridge regression and classification. *Annals of Statistics* **46**, 247–279.

Dudoit, S. and Fridlyand, J. (2003). Classification in microarray experiments. In: Speed, T. (editor), *Statistical Analysis of Gene Expression Microarray Data*. Chapman & Hall/CRC, pp. 93–158.

Fan, J., Feng, Y. and Tong, X. (2012). A road to classification in high dimensional space: The regularized optimal affine discriminant. *Journal of the Royal Statistical Society: Series B* **74**, 745–771.

Fan, J., Liao, Y. and Mincheva, M. (2013). Large covariance estimation by thresholding principal orthogonal complements. *Journal of the Royal Statistical Society: Series B* **75**, 603–680.

Friedman, J., Hastie, T. and Tibshirani, R. (2001). *The Elements of Statistical Learning*, Volume 1. Springer Series in Statistics. New York: Springer.

Gagnon-Bartsch, J. A., Jacob, L. and Speed, T. P. (2013). Removing unwanted variation from high dimensional data with negative controls. *Technical Report*, UC Berkeley Department of Statistics.

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

Gerard, D. and Stephens, M. (2021). Unifying and generalizing methods for removing unwanted variation based on negative controls. *Statistica Sinica* **31**, 1145–1166.

Hall, P., Jin, J. and Miller, H. (2014). Feature selection when there are many influential features. *Bernoulli* **20**, 1647–1671.

Hastie, T., Montanari, A., Rosset, S. and Tibshirani, R. J. (2019). Surprises in high-dimensional ridgeless least squares interpolation. *arXiv preprint arXiv:1903.08560*.

Jolliffe, I. T. (2002). *Principal Component Analysis*. New York: Springer.

Kneip, A. and Sarda, P. (2011). Factor models and variable selection in high-dimensional regression analysis. *Annals of Statistics* **39**, 2410–2447.

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

Leek, J. T. and Storey, J. D. (2008). A general framework for multiple testing dependence. *Proceedings of the National Academy of Sciences* **105**, 18718–18723.

Li, L. (2010). Dimension reduction for high-dimensional data. *Statistical Methods in Molecular Biology*, **620**, 417–434.

Listgarten, J., Kadie, C., Schadt, E. E. and Heckerman, D. (2010). Correction for hidden confounders in the genetic analysis of gene expression. *Proceedings of the National Academy of Sciences* **107**, 16465.

McCall, M. N., Bolstad, B. M. and Irizarry, R. A. (2010). Frozen robust multiarray analysis (fRMA). *Biostatistics* **11**, 242–253.

Nguyen, D. V. and Rocke, D. M. (2002). Tumor classification by partial least squares using microarray gene expression data. *Bioinformatics* **18**, 39–50.

Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering* **22**, 1345–1359.

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

POLLEY, E. C., ROSE, S. AND VAN DER LAAN, M. J. (2011). Super learning. In: *Targeted Learning: Causal Inference for Observational and Experimental Data*. New York: Springer, pp. 43–66.

QUAY, J. L., REED, W., SAMET, J. AND DEVLIN, R. B. (1998). Air pollution particles induce IL-6 gene expression in human airway epithelial cells via NF-κ B activation. *American Journal of Respiratory Cell and Molecular Biology* **19**, 98–106.

SAEYS, Y., INZA, I. AND LARRAÑAGA, P. (2007). A review of feature selection techniques in bioinformatics. *Bioinformatics* **23**, 2507–2517.

SUN, Y., ZHANG, N. R. AND OWEN, A. B. (2012). Multiple hypothesis testing adjusted for latent variables, with an application to the AGEMAP gene expression data. *Annals of Applied Statistics* **6**, 1664–1688.

TIBSHIRANI, R., HASTIE, T., NARASIMHAN, B. AND CHU, G. (2002). Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proceedings of the National Academy of Sciences* **99**, 6567–6572.

WAN, E. S., QIU, W., BACCARELLI, A., CAREY, V. J., BACHERMAN, H., RENNARD, S. I., AGUSTI, A., ANDERSON, W., LOMAS, D. A. AND DEMEO, D. L. (2012). Cigarette smoking behaviors and time since quitting are associated with differential DNA methylation across the human genome. *Human Molecular Genetics* **21**, 3073–3082.

WANG, J., ZHAO, Q., HASTIE, T. AND OWEN, A. B. (2017). Confounder adjustment in multiple hypothesis testing. *Annals of Statistics* **45**, 1863–1894.

WEISS, K., KHOSHGOFTAAR, T. M. AND WANG, D. (2016). A survey of transfer learning. *Journal of Big Data* **3**, 1–40.

YANG, P., HWA YANG, Y., ZHOU, B. B. AND ZOMAYA, A. Y. (2010). A review of ensemble methods in bioinformatics. *Current Bioinformatics* **5**, 296–308.

ZHENG, Z., LV, J. AND LIN, W. (2017). Nonsparse learning with latent variables. *arXiv:1710.02704*.

ZOU, H. AND HASTIE, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B* **67**, 301–320.

[*Received February 20, 2021; revised November 9, 2021; accepted for publication November 24, 2021*]