RubricOE: a learning framework for genetic epidemiology

Subrata Saha[1,†], Aldo Guzmán-Sáenz[2,†], Aritra Bose[2,†], Filippo Utro[2], Daniel E. Platt[2], and Laxmi Parida[2,*]

[1]Department of Systems Biology, Columbia University Medical Center, New York, NY 10032

[2]Computational Genomics, IBM T.J Watson Research Center, Yorktown Heights, NY 10598

[†]Equal contribution

[*]Corresponding author: parida@us.ibm.com

Running Title: RubricOE

Keywords: Association Studies; Machine Learning; Polygenic Risk Score

Abstract

Genetic epidemiology is a growing area of interest in the past years due to the availability of genetic data with the decreasing cost of sequencing. Machine learning (ML) algorithms can be a very useful tool to study the genetic factors on disease incidence or on different traits characterizing a population. There are many challenges that plagues the field of genetic epidemiology including the unbalanced case-control data sets, fallibility of standard genome wide association studies with single marker analysis, heavily underdetermined systems with millions of markers in contrast of a few thousands of samples, to name a few. Ensemble ML methods can be a very useful tool to tackle many of these challenges and thus we propose RubricOE, a pipeline of ML algorithms with error bar computations to obtain interpretable genetic and non-genetic features from genomic or transcriptomic data combined with clinical factors in the form of electronic health records. RubricOE is shown to be robust in simulation studies, detecting true associations with traits of interest in arbitrarily structured multi-ethnic populations.

Introduction

In recent years, with the availability of large cohorts for complex diseases, identifying the associated genetic variants as well as gene-environment interactions has become more challenging. Genome-wide association studies (GWAS) detect associations between genetic variants (single nucleotide polymorphism or SNP) and traits is a powerful tool but is sensitive to sample sizes, population structure, rare variants, environmental factors, etc. However, GWAS has several challenges in disentangling environmental effect from true genetic associations of a disorder leading to spurious associations. It also suffers from the curse of dimensionality with the availability of millions of SNPs for several thousands individuals, leading to the "small n, large p problems" (due to the under-determined nature of the genotype matrix). With the availability of more population controls than cases for binary traits due to low prevalence of many conditions/diseases the genotype data sets are often unbalanced. The scale of the data as well as the unbalanced nature of binary traits pose a substantial challenge for GWAS in large cohorts [1]. The SNPs are often genotyped with linkage disequilibrium (LD) between them resulting in correlated variables, which needs to be taken into account to find true genetic associations. Therefore, standard multi-

variable statistical approaches like linear or logistic regression are not well suited for genome-wide data [2]. Due to these shortcomings there has been a need for sophisticated machine learning approaches.

Machine learning (ML) algorithms can learn from the data without making any model assumptions. It provides several alternatives for performing multi-SNP analyses away from the single marker analysis as done in GWAS. It can employ techniques such as regularization and cross-validation in regression to tackle overfitting issues for under-determined data sets. Machine learning methods have been applied on a broad range of problems in genomics including how to recognize locations of transcriptions start sites, promoters, enhancers, etc in a genome sequence [3]. It has also been employed for genomic selection in plant breeding [4] as well as in GWAS [2] and genetic epidemiology [5]. ML can also elucidate non-linear SNP-SNP epistatic interactions, such as testing for N^2 SNP-SNP combinations for N SNPs. Generally, with multiple hypothesis Bonferroni corrected thresholds, this can be a difficult task with linear regression based techniques. For binary traits of complex diseases, the challenge is to identify clusters of alleles that suggest interactive pathways which are potential therapeutic targets.

In order to do that, it is important to understand how error analysis and statistical power interact in ML as probes to characterize identified candidates for pathogenic SNPs or other features. Therefore, we present a ML pipeline, called RubricOE, namely, a rubric for multi-omics association studies and genetic epidemiology. It includes main stages for feature selection, and for cross-validation applied to the selected feature set from the first stage. We consider the effect of error analysis to feature selection and cross validation. Lastly, it also considers the possibility of identifying SNPs that are purely epistatic. We find that RubricOE can correctly identify true associations in simulated data. It observes a trade-off with genetic effect in simulated datasets leading to minimal spurious associations for increased genetic effect and vice versa.

RubricOE is a robust ML pipeline that ranks disease associated SNPs and predicts the outcome of different disorder or traits. It provides further interpretation of the selected associations with their ranks and scores for studying their relative importance. The ranks and scores of SNPs closely imitates Polygenic Risk Scores (PRS) and thus can be a very useful tool for studying heritability of different traits. RubricOE is less sensitive to noise than standard GWAS and provides different optimizations in selecting hyper-parameters further elucidating the importance of each

selected SNP. It also provides a framework to integrate clinical Electronic Health Record (EHR) information along with genomic data, providing further information on interactions between genetic and non-genetic factors associated with a disease or trait.

Results

Simulated Data

We applied RubricOE on six different simulation data sets with 1,000 individuals across 10,000 SNPs, related to two different phenotype distribution cases for each of the three population structure simulation scenarios related to genotypes. We observe that in the most challenging scenario with very low genetic effect (10%) consisting of only ten causal SNPs, RubricOE detects 30% of causal SNPs in the real-world TGP scenario. The Youden's J statistic on the validation data in this case is understandably very low due to the overwhelming amount of noise in the dataset. Alternatively, when the genetic effect (70%) is substantially more than environmental effect and noise, RubricOE is able to discover accurately around 90% of the true causal SNPs in PSD and 70% in TGP, respectively. The Youden's J statistic are also relatively higher with around 0.45 in PSD model, even though there are only ten true causal SNPs contributing to the binary phenotype. The score curves of the above experiments show that the peak scores of each

	BN		PSD		TGP	
	Causal SNPs	J statistic	Causal SNPs	J statistic	Causal SNPs	J statistic
10-20-70	0	0.038	10%	0.056	30%	0.061
70-20-10	70%	0.505	90%	0.446	70%	0.261

Table 1: RubricOE performance on validation data in simulation studies evaluated by the percentage of causal SNPs detected and the mean Youden's J statistic observed.

feature in RubricOE is reached by using a very small set of features. As RubricOE computes the rank of the features by the descending order of their score and then computes Youden's J statistic iteratively on the dataset, it gives an opportunity to understand how much a subset of feature contributes to the total classification accuracy of the learning algorithm. Thus, we observe, in case of PSD which accurately identifies 90% of causal associations, it needs only the first few hundred features to reach its peak cumulative score (Figures 4-6 in Appendix), deeming the rest of the SNPs non-informative. We see similar patterns in other simulation scenarios such as TGP

and BN as well. We can leverage this information in making RubricOE faster by asking it to compute curves only with a few thousand features instead of the large number of SNPs, genes and clinical features, sometimes amounting to a few millions. This provides a significant scale-up in computational time of the pipeline. As SVM has quadratic complexity, this feature of RubricOE is particularly useful.

Discussion

RubricOE provides a ML framework with which a stable set of features discriminating between the healthy controls and diseased patients, can be obtained. This differs from a single marker analysis as performed in GWAS. ML algorithms allow multi-SNP analysis along with interpretation of the subset of SNPs classifying different diseases. It ranks the SNPs along with their relative scores as calculated by Youden's J statistic. One of the advantages of ML algorithms is in applying criteria for accepting features that focus on aggregated predictive power. For regression, the fraction of variation predicted by a feature is closely related, and for standard regression equal, to the square of the z-score for the coefficient. The downside is that large models may be underdetermined, and feature selection may identify false features due to overfitting. Regulation (Lasso and Ridge) mitigates that, but complicates the relationship between feature significance and how much variation is accounted for by the feature. PRS is comprised by most of the features of ML algorithms explored here, including susceptibility to overfitting. So the diffusion of functional associations may be spurious.

Associations can be spurious due to environmental effects as well as genotyping errors, etc. We explored robustness of RubricOE to such effects in the simulation studies and observed that the framework was able to correctly detect about 70% of true causal associations in the simulation scenarios closely emulating real-world population structure when the genetic effect is significantly more than non-genetic factors. RubricOE is also able to detect about 20-30% of causal associations when the genetic effect is only 10%, among which only 0.1% is causal. This demonstrates the prowess of RubricOE in detecting rare associations in the presence of noise.

The estimate of an individual's genetic liability to a trait or disease of interest is captured by the PRS computation. It returns a score which is parallel to the Youden's J statistic described here. Another factor that RubricOE can provide a solution to is the infamous "missing heritabil-

ity" issue. Single marker analysis as done in GWAS does not account for the heritabiity of a trait or disease across generations. PRS aims to solve that problem, but, RubricOE provides a straight-forward one-pass solution for doing multi-SNP tests. One of the advantages of ML algorithms is in applying criteria for accepting features that focus on aggregated predictive power. For regression, the fraction of variation predicted by a feature is closely related, and for standard regression equal, to the square of the z-score for the coefficient. The downside is that large models may be underdetermined, and feature selection may identify false features due to overfitting. Regularization (Lasso and Ridge) mitigates that, but complicates the relationship between feature significance and how much variation is accounted for by the feature. We showed that, for random sampling variation, cross-validation does not really boost power. However, cross-validation is an effective way to recognize biased variations (e.g. batch effects). PRS is comprised by most of the features of machine learning algorithms explored here, including susceptibility to overfitting. So the diffusion of functional associations may be spurious. A detailed study on the parallels between PRS and Youden's J statistic as well as demonstrate how it approaches the "missing heritability" issue across multi-ethnic cohorts would highlight the importance of studying other ML algorithms such as neural network or deep learning in disorder trait prediction and interpretable feature selection. RubricOE, or a rubric for omics and epidemiology can be useful in stable feature selection from integrated multi-omic and clinical data, elucidating their interactions.

Methods

Datasets

Simulation Study

We generated an extensive set of simulations with real-world challenging scenarios to demonstrate the robustness to different scenario and power to detect true disorder or trait associated SNPs. We simulated and analyzed six different data sets with 1,000 individuals and 10,0000 SNPs, pertaining to three population structure scenarios and for each, two phenotype simulation scenarios. The data was simulated based on a binary trait model using the Odds Ratio (OR) as the classifier for

disease status from a quantitative trait model described in previous work [6].

$$y_j = \alpha + \sum_{i=1}^{m} \beta_i \mathbf{X}_{i,j} + \lambda_j + \epsilon_j \tag{1}$$

where β_i is the genetic effect of SNP i on the trait, λ_j is the random non-genetic effect and ϵ_j is the random noise variation for individual j. $\mathbf{X}_{i,j}$ is the i^{th} marker for the j^{th} individual and $y \in \mathbb{R}^m$ is the trait response variable (binary or continuous).

For the genotype data, we simulated allele frequencies using (i) Balding-Nichols (BN) model [7] based on allele-frequency and F_{ST} estimates calculated on the HapMap data set, (ii) three different levels of admixture by varying the parameter α from {0.01,0.1,0.5} in Pritchard-Stephens-Donnelly model (PSD) [8] and (iii) structure estimated from 1000 Genomes Project (TGP) [9]. The three scenarios allow us to simulate genotypes with varying degree of population structure [6, 10]. BN provides three arbitrarily structured group of populations who are not mixing with each other. This provides an ideal case of structure to evaluate the model. PSD model accounts for admixture between these arbitrarily structured clusters of individuals (such as individuals between each group share alleles between them) and the degree of admixture can be altered by the parameter α. Lastly, TGP model provides the most real-world like scenario where the individuals are sampled from allele frequencies corresponding to different population across the world as represented in the 1000 Genomes data set [9].

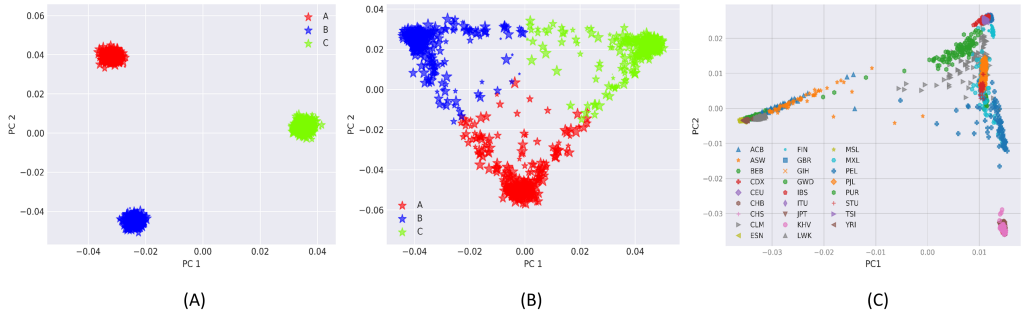


Figure 1: Projection of the samples from three populations simulated from (i) BN (ii) PSD ($\alpha = \{0.1, 0.1, 0.1\}$) and (iii) TGP model on the top two axes of variation.

We used different variance ratio for the phenotype simulation scenarios. For the first scenario, we allowed 70% genetic effect, 20% environmental effect and 10% noise effect. This allowed us

to study whether the model can accurately detect the true associations when the genetic effect is substantial. In comparison, we reversed the variances of the genetic effect (10%) and noise (70%) to simulate a more challenging scenario. The causal genetic effect is simulated by independently simulating the first 0.001 of the total number of SNPs, n (10,000 SNPs). β_i $Normal(0, 0.5)$ for $i = 1, 2, \cdots, 0.001 * n$. For all other $i > 0.001 * n$ we set $\beta_i = 0$.

RubricOE

The ML pipeline,"RubricOE" outlined in Figure 2, has 3 major components: Quality Control (QC) of input omics data, Iterative Feature Selection (IFS), and Stable Set Construction (SSC). Optional selection of machine learning components with hyper-parameters may be employed in each to optimize performance. The pipeline employs a nested test-training set configuration. The outer test-training set split reserves the test set ("validation") for final SNP evaluation and denotes the training set as "working" data. Within the latter, further train-test splits are performed to rank the features.

The working data is used within the IFS stage. In this stage, replications of data test-training splits are each subject to iterative feature ranking, using machine learning algorithms (Ridge Regression in this case), and scored against the replicated test sets using a scoring metric (e.g. Youden or h^2) using another machine learning algorithm (Support Vector Machines or SVM with linear kernel in this case) to identify an subset of SNPs. If the ML is non-linear in character, feature selection is based on recursive feature elimination (RFE) [11] is an option we have used. These rankings are combined to form a final candidate set. Models constructed from the final set are applied to the "unseen" (test) set for final scoring. Iterative replications are performed subsequently, and the set of SNPs that persistently survive in the intersection are called the "Stable Set".

Iterative Feature Selection

Robust SNP ranking Some of the ranking algorithms that were applied here, besides application of simple ridge regression, are described here. We attempt to make our SNP ranking model robust by employing multiple linear SVMs (LSVMs) in a concerted way. Each LSVM runs on slightly different data sets randomly sampled from original dataset D. In the experiment, we

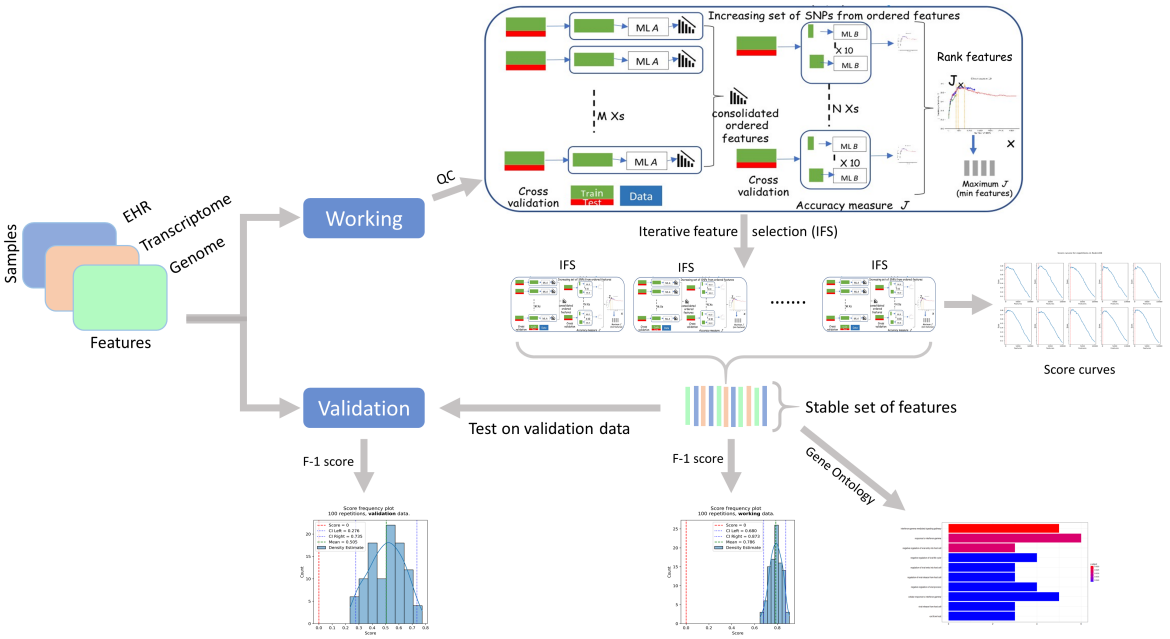


Figure 2: Components of RubricOE, the ensemble ML pipeline producing a stable set of features from multi-omics data.

randomly picked 90% unique samples from D for each LSVM to build a robust learning model. Due to this small change, the weight associated with each SNP is not identical for different runs. By averaging the weights across a set of weight vectors from LSVMs of the same configuration, we can make the weight vector robust. Since weights are directly associated with the importance of the SNPs, ranking should also be robust.

Let the number of LSVMs to rank a set S of SNPs of interest be L. Each SVM $l \in L$ is trained on a slightly smaller dataset D' to build an inductive model and we get a corresponding weight vector w_l $(1 \leq l \leq |L|)$. According to [11], the significance of the i^{th} SNP $I_i = (w_l^i)^2$ where w_l^i represents the i^{th} weight component of w_l^{th} weight vector. At the end of this step, we get a set of $|L|$ weight vectors. Since, in each run, we introduce slightly different dataset D' by randomly sampling the original dataset D, the weight vectors will be different from each other. To make it stable we normalize each weight vector and average each component. Each weight vector w_l will have n components and is normalized as follows: $w'_l = \frac{w_l}{\sum_{i=1}^{n} |w_l^i|} \ldots (2)$. The i^{th} component of the final weight vector W is formed as follows: $W^i = \sum_{l=1}^{|L|} |(w_l'^i)| \ldots (3)$. Now, the importance of the i^{th} SNP is defined as $I_i = \frac{W^i}{n}$. We sort the SNPs based on their significance in non-decreasing order. So, the highest significant SNP will have lowest rank, the 2^{nd} highest significant SNP will

have 2^{nd} lowest rank, and so forth.

Robust SNP subset selection Finding optimal set of features from n features will take $\mathcal{O}(2^n)$ time which is exponential in the number of features. To reduce the search space, we follow two step procedure. At first, we employ robust SNP ranking algorithm (as described above) to rank SNPs based on their significance. In the second step, we linearly search through the ranked SNP space to get the subset of SNPs maximizing the classification accuracy (such as Youden index). Next, we describe the procedure in detail.

Let $r_0, r_1, r_2, \cdots, r_{n-1}$ be the ranks of the given SNPs where r_0 is the smallest rank, r_1 is the second smallest rank, etc. Next, we take top x SNPs (i.e. $r_0, r_1, \cdots, r_x$ where $x \ll n$) and compute Youden index using 10-*fold* cross validation based on those x SNPs. Subsequently, we compute Youden indices based on top $2x$ SNPs, top $3x$ SNPs, etc. As soon as there is no improvement over the maximum Youden index seen so far in succeeding iterations, we stop the linear search and return back those SNPs having maximum Youden index. These SNPs constitute our robust subset of SNPs.

Stable Set Construction

Finally, the whole procedure (i.e. SNP ranking and subset selection) is repeated multiple times to obtain stable set of features. For an illustrative example, let's assume we run the entire procedure p times. Consequently, we get p rankings of SNPs and p subsets of robust SNPs (through ranking and subset selection procedure). Finally, we extract common SNPs among the p subsets. Suppose, t_i represent a robust subset of SNPs from i^{th} repetition where $1 \leq i \leq p$. Then the stable subset of SNPs will be $T = t_1 \cap t_2 \cap t_3 \cap \cdots \cap t_p$.

Cross Validation

Cross validation splits data into a training, or discovery, set D used to determine a set of SNPs P_D that passes a threshold predicting a phenotype, and a test set V independent of the training set, used to test, or validate, the predictive power of the test P_V on the training set's best variants. This is used in the "Feature Set Selection" and "Stable Set Construction" stages of our nested training-test set configuration. Therefore, a major component of power analysis of these methods revolve around evaluating factors impacting cross validation.

The discovery set parameters are $\mathbb{P}(P_D|\bar{A}) = \alpha_D$ and $\mathbb{P}_{\mathbb{D}}(P_D|A) = 1 - \beta_D$, and the validation set parameters are $\mathbb{P}(P_V|\bar{A}) = \alpha_V$ and $\mathbb{P}_{\mathbb{V}}(P|A) = 1 - \beta_V$. $\mathbb{P}(A) = f$ in both groups. P_D and P_V are assumed to be independent. That is $\mathbb{P}(P_V|P_D) = \mathbb{P}(P_V)$, etc, and $\mathbb{P}(P_V \cap A|P_D \cap A) = \mathbb{P}(P_V| \cap A)$, Following arguments above,

$$\begin{aligned}
\mathbb{P}(P_D \cap P_V) &= (1-\beta_D)(1-\beta_V)f + \alpha_D\alpha_B(1-f) \\
\mathbb{P}(P_D \cap P_{\bar{V}}) &= (1-\beta_D)\beta_V f + \alpha_D(1-\alpha_B)(1-f) \\
\mathbb{P}(P_{\bar{D}} \cap P_V) &= \beta_D(1-\beta_V)f + (1-\alpha_D)\alpha_B(1-f) \\
\mathbb{P}(P_{\bar{D}} \cap P_{\bar{V}}) &= \beta_D\beta_V f + (1-\alpha_D)(1-\alpha_B)(1-f)
\end{aligned}$$

These relations satisfy $\mathbb{P}(P_D \cap P_V) + \mathbb{P}(P_D \cap P_{\bar{V}}) + \mathbb{P}(P_{\bar{D}} \cap P_V) + \mathbb{P}(P_{\bar{D}} \cap P_{\bar{V}}) = 1$.

We apply the above to estimate numbers of SNPs expected to be shared between training and test sets in some of the steps of our ML algorithm.

An odds ratio may be defined

$$r = \frac{\mathbb{P}(P_D \cap P_V)\mathbb{P}(\bar{P}_D \cap \bar{P}_V)}{\mathbb{P}(P_D \cap \bar{P}_V)\mathbb{P}(\bar{P}_D \cap P_V)}$$

If there were no biologically active SNPs, then $f = 0$, and $r = 1$. So this can test whether the number of SNPs in both test and training sets is more than expected by chance.

Model selection

We use two different ML algorithms for the two steps outlined in the RubricOE pipeline (Figure 2), IFS and SSC. We use Ridge Regression for ranking the features in IFS and LSVM to score the Youden curves for optimization.

One standard option for selecting the initial feature ranking has been to use regression coefficients. In our approach, one method for ranking in the 'IFS stage is ridge regression ranking by magnitude of the coefficients. Another alternative was to consider the variation in the phenotype covered by the coefficient.

In standard regression, given $\mathrm{cov}(y, y^T) = \Sigma$, a χ^2 distributed statistic measuring goodness of fit of a model $y = Xb$ is

$$\mathcal{E}^2 = (y - Xb)^T\Sigma^{-1}(y - Xb).$$

This may be rewritten

$$\mathcal{E}^2 = \mathcal{E}^2_{res} + (b-\beta)^T(X^T\Sigma^{-1}X)(b-\beta)$$

where

$$\mathcal{E}^2_{res} = y^T\Sigma^{-1}y - (X\beta)^T\Sigma^{-1}(X\beta)$$

with

$$\beta = (X^T\Sigma^{-1}X)^{-1}X^T\Sigma^{-1}y,$$

and

$$\mathrm{cov}(\beta,\beta^T) = (X^T\Sigma^{-1}X)^{-1}.$$

Given this, it is notable that the residual may be written

$$\mathcal{E}^2_{res} = y^T\Sigma^{-1}y - \beta^T(\mathrm{cov}(\beta,\beta^T))^{-1}\beta,$$

noting that X and b were augmented by a feature identical to unity in X, and equal to an offset in b. This gives a measure of variability to the variabiity in the mean for y. Therefore, the multivariate equivalent of sums of squares of the z-scores, including effects of correlations, represents the contribution of the predicted values to the residual. We identify the unexplained variability to be $h^2 = \mathcal{E}^2_{res}$, with the proportion explained by individual features h^2_i for β_i as $h^2_i = \beta^T_i(\mathrm{cov}(\beta,\beta^T))^{-1}\beta_i$ for each component by itself. We note that $y^T\Sigma^{-1}y$ is not centered. The contribution from offsets to the variability are included in the offset variable included in the augmented β's offset estimation. Centering y shifts the offset, but does not change the coefficients. The residual error $\mathcal{E}^2_{res}$ represents the proportion of phenotype not predicted by the regression model, including genetics and adjustment.

Feature selection regressions tend to be underdetermined, so a regularization that preserves the form of $\mathcal{E}^2_{res}$, and that provides error bars and covariances for the coefficients is desirable. This argues for an $L2$ regularization. In this case, it is assumed that the coefficients b in the regression are themselves distributed, and have terms with variance C^{-1} and mean 0. Then for $\mathrm{cov}(y,y^T) = \Sigma$, and $\mathrm{cov}(b,b^T) = C^{-1}I$ for a number C, a χ^2 distributed statistic measuring

goodness of fit of a model $y = Xb$ is

$$\mathcal{E}^2 = (y - Xb)^T \Sigma^{-1} (y - Xb) + Cb^T b.$$

This may be rewritten

$$\mathcal{E}^2 = \mathcal{E}^2_{res} + (b - \beta)^T (X^T \Sigma^{-1} X + C \cdot I)(b - \beta),$$

where

$$\mathcal{E}^2_{res} = y^T \Sigma^{-1} y - \beta^T (X^T \Sigma^{-1} X + C \cdot I)\beta,$$

with

$$\beta = (X^T \Sigma^{-1} X + C \cdot I)^{-1} X^T \Sigma^{-1} y,$$

and

$$\mathrm{cov}(\beta, \beta^T) = (X^T \Sigma^{-1} X + C \cdot I)^{-1} X^T \Sigma^{-1} X (X^T \Sigma^{-1} X + C \cdot I)^{-1}.$$

In the above, the

$$(X^T \Sigma^{-1} X + C \cdot I)^{-1}$$

term induced by regularization tempers the response of β to variations in y. But the contribution to $\mathcal{E}^2_{res}$ that reflects the $\mathrm{cov}(\beta, \beta^T)$ contribution is the $(X^T \Sigma^{-1} X)$, while the $C \cdot I$ does not reflect predictive information. Expanding, this becomes

$$\mathcal{E}^2_{res} = y^T \Sigma^{-1} y - (X\beta)^T \Sigma^{-1} (X\beta) - C\beta^T \beta.$$

Given this, note that

$$(X\beta)^T \Sigma^{-1} (X\beta) \neq \beta^T (\mathrm{cov}(\beta, \beta^T))^{-1} \beta$$

So the amount of variation due to the y variation accounted for in the residual is not equivalent to the z-score squared of the coefficients in the way that it is in non-regularized linear regression. Note that the part of the contribution to $\mathcal{E}^2_{res}$ that responds to y is $(X\beta)^T \Sigma^{-1} (X\beta)$. However, while the direct relationship between variability is lost in ridge regression, the expression capturing the dependency on y variation should be applied for ranking, using the $h_i^2 = (X\beta_i)^T \Sigma^{-1} (X\beta_i)$

as in standard regression described above, but with the ridge-regression estimated coefficients. Note also that $X\beta$ is still the estimator for y, so $(X\beta)^T\Sigma^{-1}(X\beta)$ is still the component that describes contributions to the proportion of variability in $y^T\Sigma^{-1}y$. We apply ridge regression to feature selection as described above, computing the regression in the singular-value decomposition (SVD) basis. The value of the regularization parameter C is chosen to be the median singular value obtained from the SVD.

Characterizing Stable Features

Logistic regression is evaluated for the stable set applied to the training sets and to the final test set to understand how cross-validated selection of final features relates to their estimated uncertainties, and to understand more clearly the power analysis situation in this problem.

Code Availability. RubricOE executable is available at `https://github.com/ComputationalGenomics/RubricOE`.

Data Availability. All simulated data was used in this manuscript.

Acknowledgements. This work was conducted by SS as part of his Postdoctoral Research at IBM T.J Watson Research Center.

References

[1] Zhou, W., Nielsen, J. B., Fritsche, L. G., Dey, R., Gabrielsen, M. E., Wolford, B. N., LeFaive, J., VandeHaar, P., Gagliano, S. A., Gifford, A., et al. (2018). Efficiently controlling for case-control imbalance and sample relatedness in large-scale genetic association studies. *Nature genetics* **50**(9), 1335–1341.

[2] Szymczak, S., Biernacka, J. M., Cordell, H. J., González-Recio, O., König, I. R., Zhang, H., and Sun, Y. V. (2009). Machine learning in genome-wide association studies. *Genetic epidemiology* **33**(S1), S51–S57.

[3] Libbrecht, M. W. and Noble, W. S. (2015). Machine learning applications in genetics and genomics. *Nature Reviews Genetics* **16**(6), 321–332.

[4] González-Camacho, J. M., Ornella, L., Pérez-Rodríguez, P., Gianola, D., Dreisigacker, S., and Crossa, J. (2018). Applications of machine learning methods to genomic selection in breeding wheat for rust resistance. *The plant genome* **11**(2), 170104.

[5] Dasgupta, A., Sun, Y. V., König, I. R., Bailey-Wilson, J. E., and Malley, J. D. (2011). Brief review of regression-based and machine learning methods in genetic epidemiology: the Genetic Analysis Workshop 17 experience. *Genetic epidemiology* **35**(S1), S5–S11.

[6] Song, M., Hao, W., and Storey, J. D. (2015). Testing for genetic associations in arbitrarily structured populations. *Nature genetics* **47**(5), 550.

[7] Balding, D. J. and Nichols, R. A. (1995). A method for quantifying differentiation between populations at multi-allelic loci and its implications for investigating identity and paternity. *Genetica* **96**(1-2), 3–12.

[8] Pritchard, J. K., Stephens, M., and Donnelly, P. Jun (2000). Inference of population structure using multilocus genotype data. *Genetics* **155**(2), 945–959. 10835412[pmid].

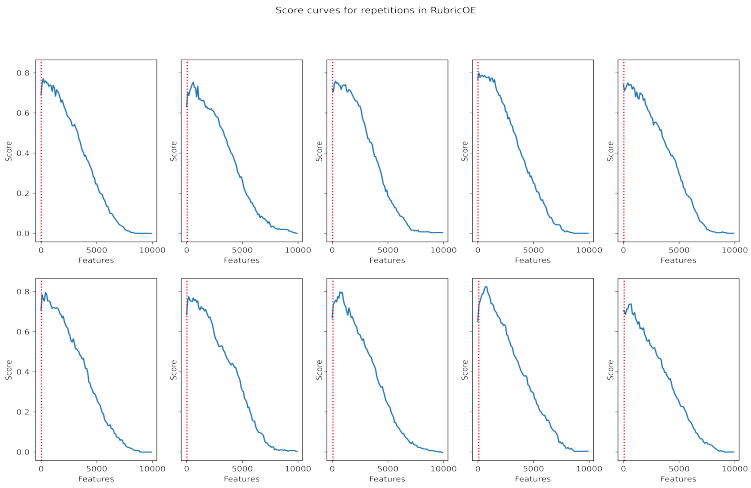
[9] Auton, A., Abecasis, G. R., Altshuler, D. M., Durbin, R. M., Abecasis, G. R., Bentley, D. R., Chakravarti, A., Clark, A. G., Donnelly, P., Eichler, E. E., et al. (2015). A global reference for human genetic variation. *Nature* **526**(7571), 68–74.

[10] Hao, W., Song, M., and Storey, J. D. (2015). Probabilistic models of genetic variation in structured populations applied to global human studies. *Bioinformatics* **32**(5), 713–721.

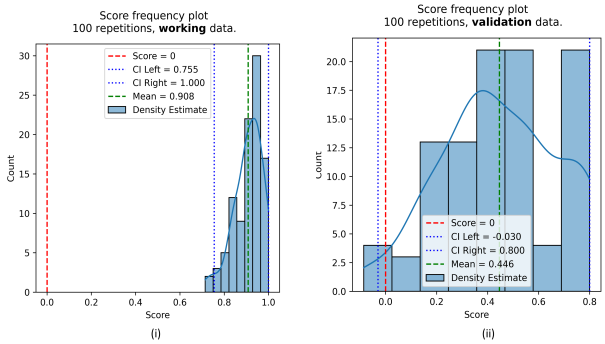
[11] Guyon, I., Weston, J., Barnhill, S., and Vapnik, V. January (2002). Gene Selection for Cancer Classification using Support Vector Machines. *Machine Learning* **46**(1), 389–422.

Appendix

Simulated Data

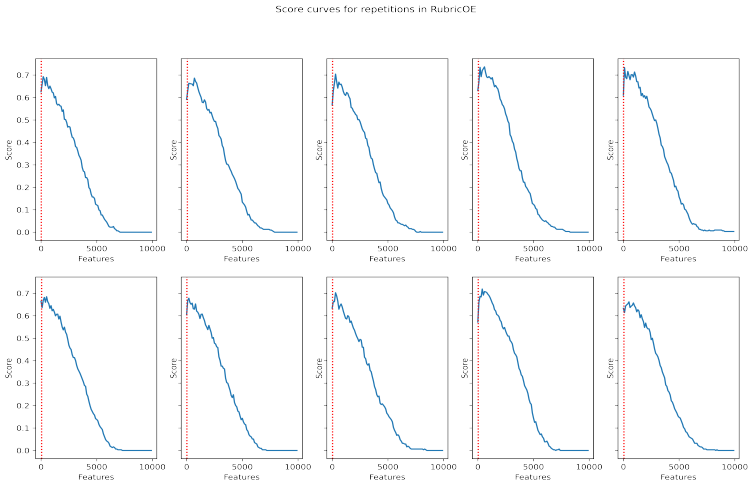


(a) Youden's J statistic score curves if IFS stage of RubricOE. The optimal number of SNPs to reach the highest peak cumulative score is marked by the red dotted line.

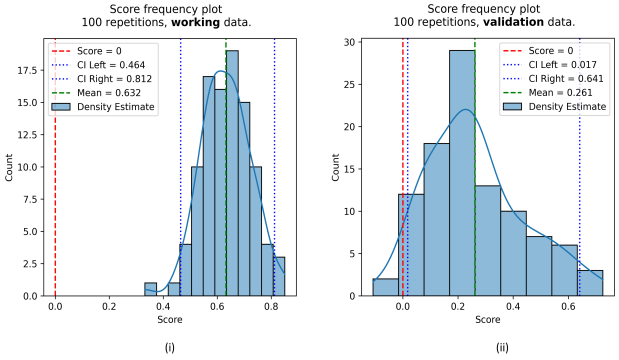


(b) Histograms of F-1 score on working and validation data.

Figure 3: Performance of RubricOE on PSD scenario with 70% genetic effect.

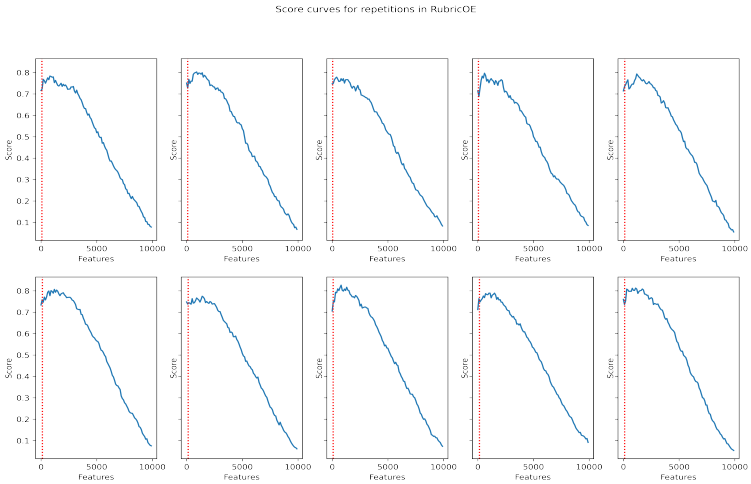


(a) Youden's J statistic score curves if IFS stage of RubricOE. The optimal number of SNPs to reach the highest peak cumulative score is marked by the red dotted line.

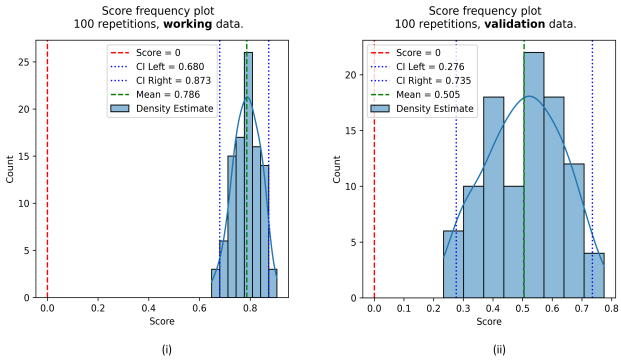


(b) Histograms of F-1 score on working and validation data.

Figure 4: Performance of RubricOE on TGP scenario with 70% genetic effect.



(a) Youden's J statistic score curves if IFS stage of RubricOE. The optimal number of SNPs to reach the highest peak cumulative score is marked by the red dotted line.



(b) Histograms of F-1 score on working and validation data.

Figure 5: Performance of RubricOE on BN scenario with 70% genetic effect.