Statistics in Biosciences manuscript No.
(will be inserted by the editor)

Prediction-oriented marker selection (PROMISE) with application to high-dimensional regression

Soyeon Kim · Veerabhadran Baladandayuthapani · J. Jack Lee

Received: date / Accepted: date

Abstract In personalized medicine, biomarkers are used to select therapies with the highest likelihood of success based on an individual patient's biomarker/genomic profile. Two goals are to choose important biomarkers that accurately predict treatment outcomes and to cull unimportant biomarkers to reduce the cost of biological and clinical verifications. These goals are challenging due to the high dimensionality of genomic data. Variable selection methods based on penalized regression (e.g., the lasso and elastic net) have yielded promising results. However, selecting the right amount of penalization is critical to simultaneously achieving these two goals. Standard approaches based on cross-validation (CV) typically provide high prediction accuracy with high true positive rates but at the cost of too many false positives. Alternatively, stability selection (SS) controls the number of false positives, but at the cost of yielding too few true positives. To circumvent these issues, we propose prediction-oriented marker selection (PROMISE), which combines SS with CV to conflate the advantages of both methods. Our application of PROMISE with the lasso and elastic net in data analysis shows that, compared to CV, PROMISE produces sparse solutions, few false positives, and small type I + type II error, and maintains good prediction accuracy, with a marginal decrease in the true positive rates. Compared to SS, PROMISE offers better prediction accuracy and true positive rates. In summary, PROMISE can be applied in many fields to select regularization parameters when the goals are to minimize false positives and maximize prediction accuracy.

S.Kim
Department of Statistics, Rice University, Houston, TX, USA
E-mail: soyeon.sophia.kim@gmail.com

S. Kim · V. Baladandayuthapani · J.J. Lee
Department of Biostatistics, The University of Texas MD Anderson Cancer Center, Houston, TX, USA

V. Baladandayuthapani E-mail: Veera@mdanderson.org
J.J. Lee E-mail: jjlee@mdanderson.org

Keywords Predictive marker · Personalized medicine · Cross-validation · Stability Selection · Variable Selection · Lasso

1 Introduction

Recent advances in genomic technologies have enabled us to obtain large amounts of biological information from patients. This has led to challenges in identifying important biological information (biomarkers) associated with relevant clinical/disease outcomes from high-throughput data sets. Biomarkers can be broadly classified into two groups: *prognostic* and *predictive* markers [**?**]. A prognostic marker is used to predict the patient's disease outcome regardless of the choice of treatment. For example, a high level of prostate-specific antigen is an indicator of prostate cancer development [**?**]. A predictive marker is used to predict the likelihood of benefit from a specific treatment. For example, in lung cancer, patients with mutations of the epidermal growth factor receptor (EGFR) are predicted to have higher response rates to erlotinib treatment than patients without EGFR mutations [**? ?**]; therefore, EGFR status is a predictive marker for lung cancer. The search for these two types of biomarkers comprises the holy grail of *personalized medicine* – one of the most important problems in cancer research today [**?**]. Personalized medicine aims to provide specific treatments to patients who have certain biomarkers, giving them therapies that are more effective and less toxic than standard treatments.

Statistically, the task of selecting biomarker profiles can be framed as a variable selection problem in standard regression settings. In this setting, prognostic markers are identified through the main effects of the biomarkers on responses (e.g., disease status). Predictive markers are detected through the marker-by-treatment interaction effects on response such as treatment response or survival times. Identifying both prognostic and predictive markers is, however, challenging. First, the search for the optimal inclusion of the main effect and interaction terms substantially increases the search space of the variables. Second, identifying both prognostic and predictive markers requires the use of clinical trial data along with the patients' genomic data, which were not routinely collected until recent years [**?**]. Thus, little published work is available on the identification of data-driven predictive markers in the statistical literature. As a result, most predictive markers have been developed on the basis of purely biological findings and have not been identified by principled statistical algorithms. Furthermore, purely biologically driven candidate markers may not accurately predict treatment outcomes [**?**]. Consequently, few predictive markers have been approved by the FDA for guiding the selection of treatments (see [**?**]).

Our methods are motivated by a clinical trial, the *Biomarker-integrated Approaches of Targeted Therapy for Lung Cancer* (BATTLE) trial, which was one of the first biomarker-based clinical trials [**?**]. The goals of that trial were to test the effects of treatments and prognostic and predictive markers on lung cancer development and progression. Four biomarker groups were chosen before the trial on the basis of biological and clinical information. The biomarker status of each patient was then used to stratify patients into groups and then randomize the patients to one of four treatments. Although some pre-selected markers were found to be useful in the final data analyses of the trial, several of the pre-selected markers could not predict the outcome of their companion treatments: patients with and without the putative predictive markers had similar treatment outcomes [**?**]. Since only biological and clinical information was used to select the candidate predictive markers, this raised the awareness of the importance of incorporating statistical processes to identify biomarkers. As a result, in the BATTLE II trial, which is ongoing at MD Anderson Cancer Center, both biological and statistical knowledge are combined in identifying prognostic and predictive markers. In the first stage, KRAS mutation, a (putatively) biologically relevant marker, is used to adaptively randomize patients to one of four treatments. Based on statistical analyses of the data from the first stage, biomarkers are selected for treatment assignment in the second stage, and the statistically chosen biomarkers guide the assignment of treatments for the next cohort of patients [**?**]. The core statistical question is how to develop a variable selection method for choosing the important markers to use in assigning the best treatment to each patient according to the patient's biomarker/genomic profile.

The following three characteristics are used to develop a statistical method for accurately selecting biomarkers in this setting.

(a) Few false positives Selecting a small number of false positives is critical for biomarker discovery. To obtain FDA approval, a biomarker must be validated clinically and biologically. If a method selects numerous false positives, biological/clinical validation is prohibitively expensive.

(b) High prediction accuracy for future data - Even when a method selects only a few false positives, it is not desirable for marker selection if it misses important variables and therefore cannot accurately predict treatment outcome. This is a critical property because the predicted response guides the choice of treatment: the treatment that is expected to yield the higher response rate has a higher probability of being assigned to a patient. If the prediction is not accurate, it would be unlikely that the trial would correctly assign the treatment that would most benefit the individual patient.

(c) Ability to handle correlated data - Biomarkers (or genes) can be correlated on the basis of their biological functions and complex underlying mechanisms. For example, genes that are part of a common biological pathway share some functional similarity and the correlation between such genes can be high [**?**]. This correlation should be handled by appropriate statistical methods.

In the statistics literature, there is limited research in identifying predictive markers in high-dimensional settings. Werft et al.[**?**] used p-values for the permutation of the regressor residual test to determine the significance of individual predictive markers. The proposed test can handle correlated variables; however, the authors focused only on controlling the false discovery rate but not on the type II error rate. In addition, the test was designed for selecting variables and not for predicting future outcomes.

Gu et al.[?] proposed a method that uses a group lasso penalty followed by an adaptive lasso for marker selection. Each group consists of a marker and the marker-by-treatment interaction effects. The group lasso selects (or not) the entire group members together. As highly correlated genes in the same pathway are in different groups in this setting, the correlation between groups can be high; however, the group lasso penalty cannot deal with strong correlations between groups [?]. Furthermore, this paper focuses on variable selection only and not on the prediction of treatment outcomes.

To deal with high-dimensional variable selection for correlated data, one of the most commonly used penalization methods is the elastic net [?]. However, finding the right amount of penalty is critical and challenging. A penalty that is too small results in keeping all the variables, and one that is too large results in having no variables detected in the model. To find the regularization parameter, cross-validation (CV) [?] is commonly used: it selects the parameter based on prediction accuracy. CV often gives good prediction accuracy, but selects a model with too many variables, resulting in too many false positives [?]. An alternative method, stability selection (SS), was proposed to control the number of false positives [?]. However, two critical cutoffs in SS, which are theoretically determined for controlling false discoveries, often yield results that are too conservative to identify important variables [? ?]. Thus, a model based on the variables selected by SS often has low prediction accuracy due to too few true positives being selected.

To conflate the advantages of both methods, i.e., good prediction accuracy along with a small number of false positives, we propose prediction-oriented marker selection (PROMISE). In PROMISE, the sub-sampling method of SS acts to reduce false discoveries and produce a sparse solution. To select the two cutoffs of SS, the CV method acts to maximize the prediction accuracy and true positive rate. Our PROMISE algorithm is general and can be applied to any penalization method; we illustrate out methods using both lasso and elastic net. A companion R code is available for public use.

This paper is organized as follows. In Section 2, we provide an overview of a logistic regression model for both prognostic and predictive marker selection, penalization methods, and regularization parameter selection methods. In Section 3, we propose PROMISE and explain the algorithm of PROMISE. In Section 5, we use an example to show how PROMISE selects variables compared to CV and SS. In Section 6 and Section 7, we compare PROMISE, CV, and SS using simulated data and the BATTLE trial data. We present prognostic and predictive markers selected by PROMISE with the elastic net, as well as their biological relationships with cancer. We conclude the paper with a discussion in Section 8.

2 Marker selection model and overview of existing methods

2.1 Logistic regression for biomarker selection

For prognostic and predictive marker selection, logistic regression is a standard model for a binary treatment outcome and when the predictors are individual markers, treatment indicator variables, and their interaction effects [**?**]. For the ith patient, we let p_i denote the probability of the treatment response, T_{iw} denote the wth treatment indicator, considering treatment 1 is the reference group, and M_{il} denote the lth marker value. When the number of patients is n, the number of treatments is W, and the number of markers is L, the basic model can be written as

$$logit(p_i) = \alpha_1 + \sum_{l=1}^{L} \eta_l M_{il} + \sum_{w=2}^{W} \alpha_w T_{iw} + \sum_{l=1}^{L} \sum_{w=2}^{W} \gamma_{lw} M_{il} T_{iw}. \tag{1}$$

The wth treatment is considered to be significant compared to the reference treatment if $\alpha_w \neq 0$ for $w = 2, 3, ..., W$. The lth marker is considered to be a prognostic marker if $\eta_l \neq 0$. Also, the lth marker is considered to be a predictive marker for the wth treatment if $\gamma_{lw} \neq 0$.

To make a general form of the logistic regression, the right side of equation (1) can be written as $\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}$, where $p = (W-1)L + W - 1$. In our microarray data, L is about 30000, the number of probe sets, and W is 4, so p is more than 120000. In this setting, the biomarker selection problem is a variable selection problem in ultra-high dimensions. A common way of dealing with a variable selection problem in ultra-high dimensions is pre-screening variables first to achieve a smaller dimension and then applying a penalized regression to select a final set of variables [**?**]. Popular penalized regression methods are the lasso and the elastic net.

2.2 Penalized methods

When the response is binary, we maximize the penalized log likelihood,

$$\max_{(\beta_0,\beta)\in\mathbb{R}^{(p+1)}} \left[\sum_{i=1}^{n} \{y_i \log\, p_i + (1-y_i)\log(1-p_i)\} - P_\lambda(\boldsymbol{\beta}) \right],$$

where

$$p_i = \frac{\exp(\beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j)}{1 + \exp(\beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j)}$$

is the probability of having a response of 1, and $P_\lambda(\beta)$ is a penalty function.

The lasso penalty [**?**], one of the most popular penalties for variable selection, is

$$P_\lambda(\boldsymbol{\beta}) = \lambda ||\boldsymbol{\beta}||_{l_1} = \lambda \sum_{j=1}^{p} |\beta_j|. \tag{2}$$

This L1 penalty shrinks some of the coefficients to exactly zero. Therefore, it automatically selects variables without considering multiple testing issues. However, it tends to select only one variable among highly correlated variables [**?**].

The ridge penalty [**?**] is another well-known penalty method.

$$P_{\lambda}(\boldsymbol{\beta}) = \lambda ||\boldsymbol{\beta}||_{l_2}^2 = \lambda \sum_{j=1}^{p} \beta_j^2$$

This ridge regression handles the multicollinearity problem well, but the L2 penalty does not shrink any of the coefficients to exactly zero [**?**]. Therefore, this method does not provide automatic variable selection.

The elastic net penalty [**?**] is a combination of the lasso ($\alpha = 1$) and ridge penalty ($\alpha = 0$) methods.

$$\begin{aligned} P_{\alpha,\lambda}(\boldsymbol{\beta}) &= \lambda \left\{ (1-\alpha)\tfrac{1}{2} ||\boldsymbol{\beta}||_{l_2}^2 + \alpha ||\boldsymbol{\beta}||_{l_1} \right\} \\ &= \lambda \left\{ \textstyle\sum_{j=1}^{p} \left[\tfrac{1}{2}(1-\alpha)\beta_j^2 + \alpha|\beta_j| \right] \right\} \end{aligned} \tag{3}$$

Thus, it has the advantages of both methods: it simultaneously selects a group of variables that are correlated with each other, while providing automatic variable selection [**?**]. Because genes can be correlated with each other when they share the same biological pathway [**?**] or have similar functionality, we apply the elastic net to choose the markers.

2.3 Calibration methods: choice of the regularization parameter

The challenge of using the penalized models is choosing the right amount of the regularization parameter(s): λ for the lasso in equation (2) and (α ,λ) for the elastic net in equation (3). This is a critical issue because variable selection largely depends on the regularization parameter. When the parameter is too large, no variable is selected, and when the parameter is too small, all variables are included. There are two major strategies for choosing the regularization parameters, CV and SS; these strategies are summarized below.

2.3.1 Cross-validation (CV)

CV is one of the most widely used methods for selecting the regularization parameter(s). CV uses part of the data to fit the model and the other part to test the model [**?**]. We use an example to explain the CV procedure in the lasso.

To perform K-fold CV, we split the data into K roughly equal sizes. For a candidate set of the regularization parameters, $\lambda_r = \lambda_1, ..., \lambda_R$, we fit the lasso, selecting variables and estimating regression coefficients, using all the data except for k-fold of the data. The estimated coefficient of the lasso using λ_r, computed with the data except for the k part is

$$\hat{\boldsymbol{\beta}}^{\text{lasso}(-k)}(\lambda_r) = \arg\max_{\boldsymbol{\beta}} \left[\sum_{i \notin k} y_i \boldsymbol{\beta}^T x_i - \log(1+\exp(\boldsymbol{\beta}^T x_i)) - \lambda_r ||\boldsymbol{\beta}||_{l_1} \right],$$

where $\boldsymbol{\beta} = (\beta_0, \beta_1, ..., \beta_p)^T$ and x_i includes the constant term 1 to include the intercept.

Then, we predict the probability of response in k-fold of the data using the predicted coefficients for each regularization parameter. For $i \in k$,

$$\hat{p}_i(\lambda_r) = \frac{\exp((\hat{\boldsymbol{\beta}}^{\text{lasso}(-k)}(\lambda_r))^T x_i)}{1+\exp((\hat{\boldsymbol{\beta}}^{\text{lasso}(-k)}(\lambda_r))^T x_i)}.$$

Then, we calculate the prediction accuracy, measured by the area under the receiver operating characteristic curve (AUC), $\text{AUC}(y^{(k)}, \hat{p}^{(k)}(\lambda_r))$ where $y^{(k)}$ is the response in the k-fold of data, and $\hat{p}^{(k)}(\lambda_r)$ is the predicted probability of response in the k-fold of data using λ_r.

We perform this procedure for k=1,...,K and then calculate the average prediction accuracy for each regularization parameter. The K-fold cross-validated AUC estimator to select λ in the lasso is

$$\text{CV}_{\text{AUC}} = \frac{1}{K} \sum_{k=1}^{K} \text{AUC}(y^{(k)}, \hat{p}^{(k)}(\lambda_r)).$$

We choose the regularization parameter that yields a model that maximizes CV_{AUC} (the maximum rule) or which gives the most parsimonious model whose CV_{AUC} is not more than one standard error (1SE) difference from the model established by the maximum rule (1SE rule) [**?**]. For variable selection purposes, using the 1SE rule is preferred since noise variables can be effectively screened out[**?**]. After selecting the regularization parameter, the entire data set is used to fit the lasso with the parameter for the final model.

CV flexibly selects the parameter based on the prediction accuracy; therefore, it provides a satisfactory result for prediction accuracy. Also, to maximize prediction accuracy, the model selected by CV tends to include most of the important variables. However, the model size is often too large and includes too many false positives [**?**].

2.3.2 Stability Selection (SS)

Whereas the goal of CV is to maximize the prediction accuracy for future data sets, the goal of SS is to control false discoveries under a desired level [**?**].

Let I denote a random sample of size $\lfloor n/2 \rfloor$ from the observed data without replacement. We are interested in finding the set of nonzero regression coefficients, $S = \{j : \beta_j \neq 0, j = 1, ..., p\}$ where p is the number of variables.

For each variable $j \in 1, ..., p$ and for each λ, we estimate the selection probability using sub-samples [**?**]:

$$\hat{\Pi}_j^{\lambda} = \frac{1}{B}\sum_{m=1}^{B} \mathbb{1}\{j \in \hat{\mathrm{S}}^{\lambda}(I_m)\},$$

where B is the number of sub-samplings and $\hat{\mathrm{S}}^{\lambda}(I_m)$ is the selected variables using λ, with the mth random sub-sample I_m of size $\lfloor n/2 \rfloor$, and $\mathbb{1}$ is an indicator function.

For a cutoff $0 < \pi_{thr} < 1$ and a set of regularization parameters Λ, the set of selected variables is

$$\hat{S}^{stable} = \left\{ j : \max_{\lambda \in \Lambda}(\hat{\Pi_j^{\lambda}}) \geq \pi_{\mathrm{thr}} \right\}. \tag{4}$$

The next question is how to select π_{thr} and the choice of the range of λ, Λ, in equation (4). Meinshausen and Buhlmann [**?**] claimed that the choice of π_{thr} in range $(0.6, 0.9)$ does not affect the result of the variable selection. They set 0.9 as a default cutoff. Once a cutoff is chosen, Λ is determined by the desired number of false positives [**?**]. For $\pi_{thr} \in (0.5, 1)$,

$$E(\mathrm{V}) \leq \frac{1}{2\pi_{\mathrm{thr}} - 1}\frac{q_{\Lambda}^2}{p}, \tag{5}$$

where V is the number of false discoveries and q_{Λ} is the average number of selected variables, $E(|\hat{\mathrm{S}}^{\Lambda}(\mathrm{I})|)$ where $\hat{\mathrm{S}^{\Lambda}} = \bigcup_{\lambda \in \Lambda} \hat{\mathrm{S}}^{\lambda}$ [**?**].

Let q denote q_{Λ} when it achieves equality, then $q = \sqrt{vp(2\pi_{\mathrm{thr}} - 1)}$ is the maximum average number of selected variables when the desired error control is $E(V) \leq v$. Then, given v and π_{thr}, q determines the range of lambda, Λ.

For example, for the default cutoff value $\pi_{\mathrm{thr}} = 0.9$ and desired error control $E(V) \leq 1$, we choose Λ such that $q = \sqrt{0.8p}$. Meinshausen and Buhlmann [**?**] suggested selecting λ_{max} and λ_{min} such that $\left|\bigcup_{\lambda_{max} \geq \lambda \geq \lambda_{min}} \hat{\mathrm{S}}^{\lambda}\right| \leq q$. However, obtaining the left side of the inequality can be very computationally demanding [**?**]. We simplify it as $\left|\bigcup_{\lambda_{max} \geq \lambda \geq \lambda_{min}} \hat{\mathrm{S}}^{\lambda}\right| \doteq |\hat{\mathrm{S}}^{\lambda_{min}}|$, because for methods that select smaller variables as the regularization parameter increases, for $\lambda \geq \lambda^{'}$, $\hat{S}^{\lambda} \subseteq \hat{S}^{\lambda^{'}}$ [**?**]. Hence, in the range of λ, we consider only λ_{min} to control the false discoveries. In this way, we select λ_{min} to control $E(V)$ under v in the following way:

$$q_{\Lambda} = E(|\hat{\mathrm{S}}^{\Lambda}(\mathrm{I})|) \doteq E(|\hat{\mathrm{S}}^{\lambda_{min}}(\mathrm{I})|) = \frac{1}{B}\sum_{m=1}^{B}|\hat{\mathrm{S}}^{\lambda_{min}}(I_m)| \leq q = \sqrt{vp(2\pi_{thr} - 1)} \tag{6}$$

Therefore, in SS, only two parameters, π_{thr} and λ_{min}, play a critical role in determining the final set of variables once desired error control is determined.

The advantage of SS is that the sub-sampling procedure helps to control the false discoveries. However, the selection of two cutoffs (π_{thr} and λ_{min}) based on inequality (5) or inequality (6) yields results that are too conservative [**?**

?]. Therefore, it often misses many important variables. SS was developed to select variables, and the goal was not to build a predictive model from the variables. Therefore, Meinshausen and Buhlmann [**?**] did not report prediction accuracy using the selected variables when using SS. However, it is obvious that a model that does not include a considerable number of the important variables cannot predict outcome well. We also discuss this issue in Section 6 and Section 7.

3 Prediction-Oriented Marker Selection (PROMISE)

3.1 Joint optimization based on CV and SS

We propose PROMISE to increase prediction accuracy and the true positive rate by selecting the parameters (π_{thr} and λ_{min}) in SS, based on the prediction accuracy of the individual data sets rather than inequality (5).

As we have discussed, the sub-sampling method in SS plays an important role in screening out false positives. However, the selection of two cutoffs (π_{thr} and λ_{min}) based on inequality (5) often yields results with too few variables and too few true positives [**? ?**]. Also, it is impossible to find a fixed combination of cutoffs that work well for all kinds of data in order to have good prediction accuracy and variable selection accuracy. To resolve these issues, we propose PROMISE, which flexibly selects the cutoffs (π_{thr}, λ_{min}) of SS based on the prediction accuracy of the individual data sets. We apply CV to select the cutoffs of SS. Since the goal of CV is to maximize prediction accuracy, important variables are automatically included in the model selected by CV. Therefore, PROMISE, with the application of CV to SS, also ensures that important variables are included in the model in order to have good prediction accuracy. In addition, since the sub-sampling method in SS greatly reduces the false discoveries and produces a sparse solution, PROMISE also reduces the false discoveries and produces a sparse solution compared to CV. Therefore, PROMISE, which combines SS and CV, is developed to reduce the false positives and the number of selected variables using SS, and to increase prediction accuracy and the number of true positives using CV.

PROMISE uses the K-1 part of the data to sub-sample, select variables using various combinations of π_{thr} and λ_{min}, and fit logistic regression models using the selected sets of variables. The remaining K part of the data is used to estimate prediction accuracy. Then, it selects the combination of π_{thr} and λ_{min}, which shows the best prediction accuracy using the 1SE rule.

To illustrate, we apply PROMISE to the lasso in the following way. While conventional CV is used to select the regularization parameter, λ, PROMISE uses CV to determine λ_{min} and π_{thr}, the cutoffs in SS. Using K-1 folds of data, instead of fitting the lasso just one time for a set of regularization parameters as in conventional CV, we fit the lasso B times using B sub-samplings for a set of regularization parameters. Then, we calculate the selection probability for each variable $j \in 1, ..., p$ for each λ, excluding the kth part of the data,

$$\hat{\Pi}_j^{(-k)\lambda} = \frac{1}{B}\sum_{m=1}^{B} \mathbb{1}\{j \in \hat{\mathrm{S}}^{\lambda}(I_m^{(-k)})\},$$

where B is the number of sub-samplings and $\hat{\mathrm{S}}^{\lambda}(I_m^{(-k)})$ is the selected variables using λ when it is applied to a random mth sub-sample $I_m^{(-k)}$, excluding the kth part of the data with size $\lfloor (n - n_k)/2 \rfloor$, where n_k is the number of observations in k-fold of the data, and $\mathbb{1}$ is an indicator function.

Instead of selecting variables using each λ, as in conventional CV, we select variables using each combination of (λ_{min}, π_{thr}). For a candidate set of the cutoffs, $(\lambda_r, \pi_r) = ((\lambda_1, \pi_1), ..., (\lambda_R, \pi_R))$, the set of selected variables using (λ_r, π_r) without the kth part of the data is

$$\mathrm{SV}^{(-k)}(\lambda_r, \pi_r) = \left\{ j : \max_{\lambda_r \le \lambda \le \lambda_{\mathrm{max}}} (\hat{\Pi}_j^{(-k)\lambda}) \ge \pi_r \right\}.$$

Subsequently, we estimate the regression coefficients to predict outcomes, which is missing in SS. The unselected variables have 0 coefficients,

$$\hat{\beta}_j^{\mathrm{SV}(-k)}(\lambda_r, \pi_r) = 0 \text{ for } j \notin \mathrm{SV}^{(-k)}(\lambda_r, \pi_r).$$

For the set of selected variables, we fit a logistic regression model. The estimated coefficients using (λ_r, π_r), computed with the data except for the k part, are

$$\begin{aligned}\hat{\boldsymbol{\beta}}^{\mathrm{SV}(-k)}(\lambda_r, \pi_r) = \arg\max_{\boldsymbol{\beta}} \Big[&\sum_{i \notin k} y_i (\boldsymbol{\beta}^T x_i^{\mathrm{SV}}(\lambda_r, \pi_r)) \\ &- \log(1 + \exp(\boldsymbol{\beta}^T x_i^{\mathrm{SV}}(\lambda_r, \pi_r))) - \lambda ||\boldsymbol{\beta}||_{l_2}^2 \Big],\end{aligned}$$

where $\boldsymbol{\beta}$ is a vector of β_j for $j \in \mathrm{SV}^{(-k)}(\lambda_r, \pi_r)$ and $x_i^{SV}(\lambda_r, \pi_r)$ is a vector x_{ij} for $j \in \mathrm{SV}^{(-k)}(\lambda_r, \pi_r)$, including the constant term 1. We use a simple logistic regression, $\lambda = 0$, or a ridge logistic regression, $\lambda = 0.01$ only when the logistic regression model does not converge due to high dimensionality.

The remaining procedures of PROMISE are similar to those of conventional CV: we predict the probability of response in the kth part of the data using the predicted coefficients for each combination of the cutoffs. The predicted probability of response is, for $i \in k$,

$$\hat{p}_i(\lambda_r, \pi_r) = \frac{\exp((\hat{\boldsymbol{\beta}}^{\mathrm{SV}(-k)}(\lambda_r, \pi_r))^T x_i)}{1 + \exp((\hat{\boldsymbol{\beta}}^{\mathrm{SV}(-k)}(\lambda_r, \pi_r))^T x_i)}.$$

Then, we calculate the prediction accuracy, the AUC, $\mathrm{AUC}(y^{(k)}, \hat{p}^{(k)}(\lambda_r, \pi_r))$ where $y^{(k)}$ is the response in the k-fold of data, and $\hat{p}^{(k)}(\lambda_r, \pi_r)$ is the predicted probability of response in the k-fold of data using (λ_r, π_r). We perform this procedure for k=1,...,K and then calculate the average prediction accuracy for each combination of the cutoff.

Fig. 1: Algorithm of PROMISE

```
1. Set candidate $\pi_{thr}$, $(\pi_1, ..., \pi_R)$, and candidate $\lambda_{min}$, $(\lambda_1, ..., \lambda_R)$.
2. Divide a data set into K parts
3. For each part, k=(1,...,K),
   (a) Using the samples except those in part k,
       i. Randomly select $n_{\text{sub}}$ sub-samplings and fit a variable selection
          method $B$ times to obtain the selection probability, $\hat{\Pi}_j^\lambda$ for $j \in \{1, ..., p\}$ and a set of $\lambda$.
       ii. Using each combination of ($\pi_{thr}$, $\lambda_{min}$), select the variables and
           then fit a logistic regression model to estimate the regression coefficients.
   (b) Using the samples in part k, with the estimated coefficients, predict the
       outcomes and calculate the prediction accuracy.
4. For each combination of ($\pi_{thr}$, $\lambda_{min}$), calculate the CV errors.
5. Select the cutoffs ($\lambda_{min}$, $\pi_{thr}$) based on the 1SE rule.
6. Obtain the final model.
   (a) Perform $B$ sub-samplings using the entire training data set.
   (b) Using the selected $\pi_{thr}$ and $\lambda_{min}$, select the final set of variables.
   (c) Fit a logistic regression to obtain the final model.
```

The K-fold cross-validated AUC estimator to select λ_{min} and π_{thr} is

$$\text{CV}_{\text{AUC}} = \frac{1}{K}\sum_{k=1}^{K} \text{AUC}(y^{(k)}, \hat{p}^{(k)}(\lambda_r, \pi_r)).$$

We choose the combination of λ_r and π_r that provide the most parsimonious model, for which CV_{AUC} is not more than 1SE difference from the model that maximizes CV_{AUC} (1SE rule).

After choosing the cutoffs, similar to the conventional way, we use the entire data set to finalize the model. However, instead of fitting the lasso using the selected regularized parameter just one time, we fit the lasso B times using B sub-samplings in a range of λ. Then, we calculate the selection probabilities of each variable. Instead of determining the final set of variables using a regularization parameter, selected by conventional CV, we determine the final set of variables using the values of λ_{min} and π_{thr}, selected by the CV procedure in PROMISE: we select the variables whose selection probabilities are higher than π_{thr} in the lambda range $[\lambda_{\text{min}}, \lambda_{\text{max}}]$. To obtain the final model, we fit a logistic regression with the final set of variables. The PROMISE algorithm is summarized in Figure 1.

4 Application for selecting more than one regularization parameter

When we have more than one parameter to be selected, we want to determine the parameter using CV along with the cutoffs. For example, for the elastic net, we want to simultaneously select the two parameters, α and λ. The PROMISE procedure for the elastic net is as follows. We set a candidate α, $(\alpha_1, ..., \alpha_R)$. For each α, we perform the CV procedures the same as for the lasso, except that we fit the elastic net with α instead of the lasso in the sub-sampling procedure. Then, we have CV errors for each combination of α, π_{thr}, and $\lambda_{\min}$. Then, we choose the combination based on the 1SE rule. To decide the set of final variables, using the α selected by CV, we perform sub-samplings and select the variables using the cutoffs selected by CV. We fit a regression model for the final step.

Other Details

- To improve the sub-sampling procedure, we use a stratified random sampling method, which selects the same portion of response 1 or 0 as the entire data set.
- A range of λ for the sub-sampling procedure is set by the glmnet [**?**]; the maximum λ is the smallest λ that chooses no variables, and the minimum λ is the maximum $\lambda \times 0.001$. The range of λ is a sequence from maximum λ to minimum λ, with 20 values on a log scale.
- $B = 100$, the default value from Meinshausen and Buhlmann [**?**], and $\pi_r = (0.3, ..., 0.8)$.
- $\alpha_r = (0.1, ..., 1)$ for the elastic net.

5 Illustrative examples

We want to illustrate how PROMISE works using data from the BATTLE trial and generating treatment responses from the data.

BATTLE trial specifics The BATTLE trial, a phase II trial, was to evaluate 4 treatments for 255 patients with advanced non-small cell lung cancer. In the trial, the primary endpoint was the 8-week disease control, which was defined as the percentage of patients who achieved complete response, partial response or stable disease following the therapeutic intervention in the clinical trial [**?**]. Progressive disease or death is considered as not reaching disease control. Note that the disease control rate is a binary outcome. The trial was an adaptive randomized trial, which assigned patients to a treatment with a high probability of disease control rate based on their biomarker-guided profiles [**?**]. In the trial, 11 candidate predictive markers were pre-selected on the basis of biological background information such as mutations, copy numbers, and protein expression. However, after the trial, several of the markers did not turn out to be predictive markers: they did not predict treatment outcomes when they were used to select patients to receive the corresponding treatment [**?**].

In this study, we use a microarray gene expression data set, which was not used for treatment assignment in the trial, to identify predictive markers so that patients can be assigned to the best treatment according to their marker status.

Pre-screening variables and generating the treatment response We benchmarked our method with the BATTLE data. The data matrix $\mathbf{X}$ is a subset of the BATTLE microarray data set. We set the regression coefficients to generate a "pseudo treatment outcome" based on the final models from the real data analysis. The real data analysis is described in Section 7.

The microarray data (platform:Affymetrix HG1.0ST) were collected from 101 patients among 255 evaluable patients in the BATTLE trial. In this paper, we use probe-set-level data as candidate markers in our model, which consist of 33297 probe sets. For predictive marker detection, we included the marker-by-treatment interaction effects in addition to the marginal effects as variables. Since there were 4 treatments in the trial, we used 3 treatment indicator variables, defined in Table 1. Therefore, the number of variables is $33297 \times 4 + 3 = 133191$.

Treatments	t2	t3	t4
Erlotinib	-1	-1	-1
Vandetanib	1	-1	-1
Erlotinib+bexarotene	-1	1	-1
Sorafenib	-1	-1	1

Table 1: Treatment coding

Since this dimension was too high for either the lasso or the elastic net to work properly, we pre-screened the variables, which consisted of probe sets and their interaction effects with the treatments. Prior to that step, we standardized each variable with mean 0 and variance 1. Using a univariate t-test with threshold p-value of 0.0013, we selected 98 probe sets for which either the marginal effects or the interaction effects with treatments are significant. Each interaction effect was treated as one variable. (See more details in the Online Resource.) Thus, after pre-screening, the number of variables, including the interaction effects and treatment indicator variables, is $98 \times 4 + 3 = 395$. This forms our data matrix $\mathbf{X}$ in this section.

To mimic the treatment response of the real data, we set β as the estimated coefficients from the real data analysis in Section 7. We set the 13 most frequently selected probe sets as nonzero variables, and ordered the data matrix such that the first 13 markers are significant. The coefficients are from the mean values of the estimated coefficients using the elastic net with PROMISE among the results with high prediction accuracy (AUC > 0.9).

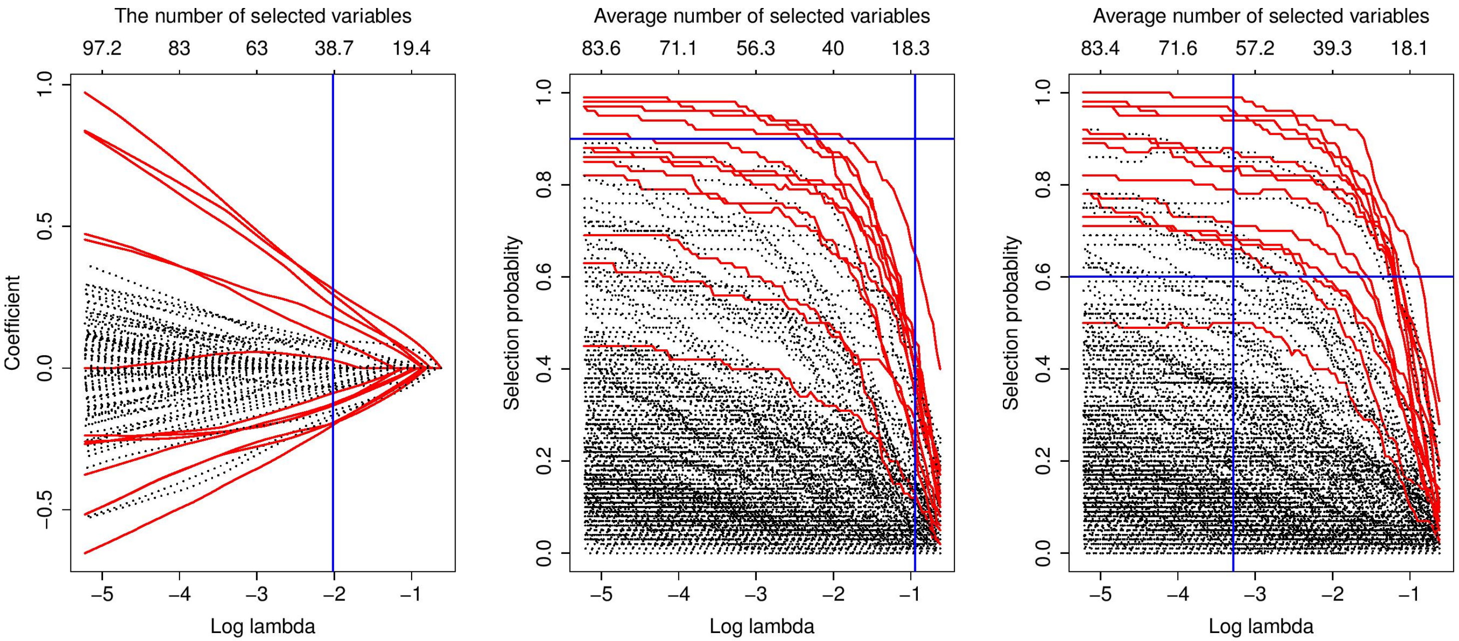
Our simulation model is

$$\begin{aligned} logit(p) = {} & (5.51M_1 + 4.51M_2 - 0.78M_3 + 4.57M_4 + 4.51M_5)T_2 \\ & + (-5.83M_6 + 1.92M_7)T_3 \\ & + (-3.72M_8 - 2.63M_9 - 3.15M_{10} - 2.27M_{11} - 2.05M_{12} + 3.44M_{13})T_4. \end{aligned}$$

Then, we simulated the treatment response from the Bernoulli distribution in the trial with the probability of response.

Results Figure 2 shows the variable selection paths of CV, SS, and PROMISE with the elastic net ($\alpha = 0.4$). (Note that we usually do not fix the value of α for data analysis. A fixed value of α is only used here to present an example.) In the regularization path, (Figure 2a), some of the important variables stand out, but many are mixed with irrelevant variables, and it is hard to identify and select important variables without selecting many noise variables. As a result, CV selected twice as many noise variables as important variables. Important variables are better identified in the stability path (Figure 2b) than in the regularization path because few variables are mixed with noise variables in the stability path. However, using the default cutoffs, $\pi_{thr} = 0.9$ and $q = 17.78$, no variable was selected by SS. The stability path using PROMISE (Figure 2c) is similar to the stability path using SS. However, a difference arises from the cutoffs, which are automatically selected by PROMISE, and which identify most of the important variables while selecting far fewer noise variables than CV. CV selected almost three times more noise variables than PROMISE.

These paths are drawn using the entire data set of observations. The analysis in Section 6 uses separate training and test data sets to evaluate prediction accuracy.



(a) Regularization path

(b) Stability path using SS

(c) Stability path using PROMISE

Fig. 2: Comparison of variable selection paths using the elastic net ($\alpha = 0.4$) on the simulated data

Solid (red) lines are the paths of important variables; dotted (black) lines are the paths of noise variables. (a) A regularization path: x-axis is log λ and the corresponding number of selected variables is shown along the top of the plot. The y-axis shows $\hat{\beta}^{\lambda}$. The vertical line is at the logλ that is selected by CV. It selects 39 variables, including all 13 important variables and 26 noise variables. (b) A stability path using the default cutoffs in SS: The x-axis is logλ and the average number of selected variables (q_{Λ}) using 100 sub-samplings is shown along the top of the plot. The y-axis shows selection probability ($\hat{\Pi}_k^{\lambda}$). The horizontal and vertical lines are the default cutoffs: $\pi_{thr} = 0.9$ and $q = 17.78$ (log$\lambda_{min} = -0.94$), respectively. The selected variables are those with selection probabilities higher than π_{thr} in the range of log$\lambda \geq$ logλ_{min} (the upper right corner formed by the intersection of the vertical and horizontal lines). No variable is selected. (c) A stability path using PROMISE: the cutoffs are selected by CV - $\pi_{thr} = 0.6$ and log$\lambda_{min} = -3.28$. PROMISE selects 21 variables, among which 12 are important and 9 are noise; it selects about 3 times fewer false positives and a similar number of true positives compared to CV.

6 Simulation studies

6.1 Simulation studies based on real data

In this section, we present a simulation example (described in Section 5) to show the performance of PROMISE, CV, and SS in terms of prediction and variable selection accuracy.

In Section 5, we used the entire data set to select variables. In this section, we split the data set into a training data set and a test data set to measure prediction accuracy along with variable selection accuracy: we randomly select three quarters of the samples as the training data set, while preserving the proportion of response 1, and use the remaining one quarter of the samples as the test data.

For CV, the training data set is used two times. First, 5-fold cross-validation is used to find the optimal regularization parameter(s), λ and α for the elastic net and only λ for the lasso. Since three quarters of the data are the training set, 3/20 of the samples are in each fold. The CV error is calculated by the AUC since the outcome is binary. Second, to estimate $\boldsymbol{\beta}$, we fit the elastic net/the lasso with the entire training set using the selected regularization parameters(s) as chosen by CV.

For SS, the training data set is used in three parts. First, to choose α, 5-fold CV is used (this step is skipped for the lasso). CV selects α and λ simultaneously, but we use only α for SS. Second, given α, to select variables, we perform SS using B sub-samplings and select variables using the default cutoffs ($\pi_{\text{thr}} = 0.9$ and $q = \sqrt{0.8p}$). Third, to estimate β, we fit a logistic regression model using the variables selected by SS.

For PROMISE, the training data set is used in three parts. First, 5-fold CV is used to choose α, $\lambda_{\min}$, and π_{thr} for the elastic net and $\lambda_{\min}$, and π_{thr} for the lasso. Second, to select the variables, we perform SS using B sub-samplings with the selected α ($\alpha = 1$ for the lasso) and select variables using the selected $\lambda_{\min}$ and π_{thr}. Third, to estimate β, we fit a logistic regression model using the variables selected by PROMISE.

We use the test data set to estimate the prediction accuracy. Prediction accuracy is measured by the AUC. A reasonable model should predict outcomes with an AUC between 0.5 and 1: AUC = 1 shows perfect prediction accuracy; and AUC = 0.5 is the same as random guessing [**?**].

We generated the simulation data set 100 times and applied PROMISE, CV, SS to the lasso and the elastic net, respectively.

Result Table 2 summarizes the prediction and variable selection results using the 100 simulated data sets. First, the stability selection does not properly function as a variable selector using either the elastic net or the lasso; it selects almost no variables and results in a random prediction (AUC=0.5). Second, although the prediction accuracy of PROMISE (around 90%) is similar to that of CV, PROMISE produces a much more parsimonious solution than CV: PROMISE selects about half of the variables using the lasso and about

Table 2: Comparison of PROMISE, CV, and SS using the BATTLE simulation data for (a) the lasso and (b) the elastic net

(a) Lasso

Method	PROMISE	CV	SS
AUC	0.87(0.01)	0.9(0.01)	0.5(0)
# Sel	12.98(0.48)	23.21(0.65)	0.01(0.01)
# TP	7.62(0.23)	9.67(0.18)	0.01(0.01)
# FP	5.36(0.32)	13.54(0.55)	0(0)
# FN	5.38(0.23)	3.33(0.18)	12.99(0.01)
# Errors	10.74(0.28)	16.87(0.51)	12.99(0.01)
TPR	0.59(0.02)	0.74(0.01)	0(0)
PPV	0.62(0.01)	0.43(0.01)	0.01(0.01)

(b) Elastic net

Method	PROMISE	CV	SS
AUC	0.9(0.01)	0.9(0.01)	0.5(0)
# Sel	23.69(1.32)	32.66(1.55)	0(0)
# TP	9.58(0.19)	10.5(0.25)	0(0)
# FP	14.11(1.22)	22.16(1.41)	0(0)
# FN	3.42(0.19)	2.5(0.25)	13(0)
# Errors	17.53(1.14)	24.66(1.29)	13(0)
TPR	0.74(0.01)	0.81(0.02)	0(0)
PPV	0.46(0.01)	0.35(0.01)	0(0)

The numbers are the mean and standard error (in parentheses) of AUC, # of selected variables (# Sel), # of true positives (# TP), # of false positives (# FP), # of false negatives (# FN), , # of false positive + false negatives (# Errors), true positive rate (TPR), and positive predictive value (PPV)

three quarters of the variables using the elastic net compared to CV. Third, PROMISE selects variables more accurately than CV, yielding smaller type I errors (false positive) plus type II errors (false negative) than CV. This is because PROMISE selects far fewer noise variables than CV even though it misses more nonzero variables than CV. Using the lasso, PROMISE selects 8 fewer noise variables than CV but misses only two more nonzero variables than CV on average. Also, using the elastic net, PROMISE selects 7 fewer noise variables than CV but misses only one more important variable than CV on average. As a result, PROMISE shows higher positive predictive value (PPV) than CV even though PROMISE shows a lower true positive rate (TPR) than CV. In biomarker discovery, PPV is a more important measure than TPR because, in reality, the denominator of TPR, which is the number of true biomarkers, is not known. Also, investigators focus more on PPV, the proportion of successes among the total findings, because it directly relates to decision making when assigning patients to treatments or performing additional tests to confirm the initial findings.

Overall, PROMISE achieves a sparser solution and more accurate variable selection than CV while giving similar prediction accuracy. In contrast, the stability selection method does not function at all for this data set.

6.2 Simulation studies with various settings

In this section, we present various simulation examples to show how many PROMISE can reduce false positives compared to that obtained by CV, while better or similarly predicting outcomes compared to CV.

So far, we used the pre-screened BATTLE data set described in Section 5 to compare the performance of the three methods. However, having only 101 samples limits the size of the test data set to measure prediction accuracy. To increase the sample size, in addition to the pre-screened BATTLE data set, we generated more samples from a multivariate normal distribution with mean $\mathbf{0}$ and covariance matrix Σ, which is the same as the covariance matrix from the pre-screened BATTLE data set.

We respectively generated 299 and 699 more observations from the multivariate normal distribution, which leads to n=400 and n=800 respectively, including the pre-screened BATTLE data set. Then, we randomly selected 1/4 of the data as a training data set and 3/4 of the data as a test data set (n_{train} = 100, 200).

To see the effect of the number of variables on the performance of PROMISE, we varied the number of variables, p. In one setting, we used the entire 98 probe sets from the pre-screened BATTLE data set (p=395). In the other setting, we selected 50 probe sets from the data set (p=203). These probe sets were randomly selected, except for nonzero variables. We also varied the number of important variables, s, as 6, 12, and 20, respectively.

Most of the coefficients are set on the basis of the real data analysis described in Section 5, and some of the coefficients are artificially set. Since the number of coefficients from the real data set is large, we also used a quarter of the coefficients to represent smaller signals in the other simulation settings. For each scenario, we generated 100 simulation data sets to test the performance of the three methods. Here are the details of the scenarios.

1. s=6 (3 prognostic markers and 3 predictive markers)
   (a) Larger signal

$$\begin{aligned} logit(p) &= 5.51M_1T_2 - 5.83M_5T_3 - 3.72M_7T_4 \\ &+ 3.98M_{11} + 1.92M_{12} + 3.44M_{15} \end{aligned}$$

   (b) Smaller signal

$$\begin{aligned} logit(p) &= 1.38M_1T_2 - 1.46M_5T_3 - 0.93M_7T_4 \\ &+ 0.99M_{11} + 0.48M_{12} + 0.86M_{15} \end{aligned}$$

M_1, M_5, M_7: predictive markers only
M_{11}, M_{12}, M_{15}: prognostic markers only

2. s=12 (6 prognostic markers and 6 predictive markers)
   (a) Larger signal

$$\begin{aligned} logit(p) = &(5.51M_1 + 4.57M_2 + 4.51M_3)T_2 \\ &- 5.83M_5T_3 + (-3.72M_7 - 2.63M_9)T_4 \\ &+ 2M_1 - 2M_5 - 2M_7 + 3.98M_{11} + 1.92M_{12} + 3.44M_{15} \end{aligned}$$

   (b) Smaller signal

$$\begin{aligned} logit(p) = &(1.38M_1 + 1.14M_2 + 1.13M_3)T_2 \\ &- 1.46M_5T_3 + (-0.93M_7 - 0.66M_9)T_4 \\ &+ 0.5M_1 - 0.5M_5 - 0.5M_7 + 0.99M_{11} + 0.48M_{12} + 0.86M_{15} \end{aligned}$$

M_1, M_5, M_7: both predictive and prognostic markers
M_2, M_3, M_9: predictive markers only
M_{11}, M_{12}, M_{15}: prognostic markers only

3. s=20 (10 prognostic markers and 10 predictive markers)
   (a) Larger signal

$$\begin{aligned} logit(p) = &(5.51M_1 + 4.57M_2 + 4.51M_3 + 4.51M_4)T_2 \\ &+ (-5.83M_5 + 1.92M_6)T_3 \\ &+ (-3.72M_7 - 3.15M_8 - 2.63M_9 - 2.27M_{10})T_4 \\ &+ 2M_1 + 2M_2 - 2M_5 - 2M_7 - 2M_8 \\ &+ 3.98M_{11} + 1.92M_{12} - 1.67M_{13} - 2.05M_{14} + 3.44M_{15} \end{aligned}$$

   (b) Smaller signal

$$\begin{aligned} logit(p) = &(1.38M_1 + 1.14M_2 + 1.13M_3 + 1.13M_4)T_2 \\ &+ (-1.46M_5 + 0.48M_6)T_3 \\ &+ (-0.93M_7 - 0.79M_8 - 0.66M_9 - 0.57M_{10})T_4 \\ &+ 0.5M_1 + 0.5M_2 - 0.5M_5 - 0.5M_7 - 0.5M_8 \\ &+ 0.99M_{11} + 0.48M_{12} - 0.42M_{13} - 0.51M_{14} + 0.86M_{15} \end{aligned}$$

M_1, M_2, M_5, M_7, M_8: both predictive and prognostic markers
$M_3, M_4, M_6, M_9, M_{10}$: predictive markers only
$M_{11}, M_{12}, M_{13}, M_{14}, M_{15}$: prognostic markers only

The results are shown in Table 3 for the lasso and Table 4 for the elastic net. For the lasso, while the prediction accuracy is similar for PROMISE and CV (Table 3a), PROMISE selects far fewer false positives than CV (Table 3b): PROMISE selects 1.53 to 5.43 times fewer false positives than CV

(Table 3c). When the sample size is larger and the number of important variables is smaller, PROMISE is more advantageous than CV. Note that when s=6 and n_{train}=200, PROMISE always has better prediction accuracy than CV and selects far fewer false positives than CV (2.60 to 5.43 times fewer) regardless of the number of variables and the strength of the signals. Prediction accuracies using SS vary according to the scenario but are never better than those achieved by PROMISE, and some of them are similar to random guessing (e.g.,smaller signals, s=20, n=100).

For the elastic net, PROMISE has better prediction accuracy than CV in most of the scenarios (Table 4a), and PROMISE still selects fewer false positives than CV (Table 4b): PROMISE selects 1.24 to 4.09 times fewer false positives than CV (Table 4c). When the sample size is larger, PROMISE has an even bigger advantage over CV. PROMISE always has better prediction accuracy than CV, and PROMISE selects 2.13 to 4.09 times fewer false positives than CV. Also, when the number of important variables is smaller, PROMISE has a bigger advantage over CV. Note that when s=6, PROMISE always has better prediction accuracy than CV and selects far fewer false positives than CV (1.38 to 4.09 times fewer) regardless of the number of variables, the number of samples, and the strength of the signals. Again, the prediction accuracies using SS vary according to the scenario, but are never better than those achieved by PROMISE, and some are similar to random guessing (e.g., smaller signals, s=20, n=100).

In summary, even with better or similar prediction accuracy compared to CV, PROMISE selects far fewer false positives than CV. Especially when combined with the elastic net, in most cases, PROMISE has better prediction accuracy than CV and produces far fewer false positives than CV. PROMISE has a bigger advantage over CV when 1) the number of samples is larger and 2) the number of important variables is smaller relative to the sample size. Additional results are in the Online Resource.

Table 3: Simulation results using the lasso: mean and standard error (in parentheses) of (a) AUC and (b) # of false positives(FP), and (c) the ratio of mean FP using PROMISE to mean FP using CV

		(a) Prediction Accuracy (AUC)			(b) The number of false positives			(c) FP Ratio
p	n_{train}	PROMISE	CV	SS	PROMISE	CV	SS	CV/PROMISE
Larger signals								
s=6								
203	100	0.943(0.004)	0.944(0.003)	0.869(0.007)	3.12*(0.38)	5.56(0.56)	0.04(0.02)	1.78
203	200	0.985*(0.001)	0.977(0.001)	0.973(0.002)	0.61*(0.11)	3.31(0.32)	0.06(0.02)	5.43
395	100	0.937(0.004)	0.934(0.004)	0.852(0.008)	3.16*(0.39)	5.35(0.45)	0.09(0.03)	1.69
395	200	0.982*(0.001)	0.975(0.001)	0.976(0.002)	0.75*(0.14)	3.85(0.37)	0.08(0.03)	5.13
s=12								
203	100	0.884(0.005)	0.898*(0.005)	0.7(0.012)	4.31*(0.4)	9.16(0.67)	0.04(0.02)	2.13
203	200	0.952(0.002)	0.95(0.002)	0.883(0.005)	2.17*(0.28)	7.99(0.58)	0.07(0.03)	3.68
395	100	0.851(0.007)	0.868(0.005)	0.709(0.009)	3.77*(0.41)	9.36(0.75)	0.08(0.03)	2.48
395	200	0.955*(0.002)	0.948(0.002)	0.886(0.004)	3.09*(0.48)	9.95(0.65)	0.06(0.02)	3.22
s=20								
203	100	0.85(0.006)	0.865(0.005)	0.61(0.01)	5.4*(0.48)	9.97(0.64)	0.08(0.03)	1.85
203	200	0.93(0.002)	0.933(0.002)	0.76(0.008)	4.5*(0.63)	11.6(0.75)	0.09(0.03)	2.58
395	100	0.809(0.007)	0.83*(0.008)	0.615(0.01)	4.38*(0.39)	11.86(0.72)	0.05(0.02)	2.71
395	200	0.915(0.003)	0.919(0.002)	0.772(0.007)	5.23*(0.6)	14.24(0.72)	0.14(0.03)	2.72
Smaller signals								
s=6								
203	100	0.82(0.006)	0.824(0.005)	0.713(0.012)	2.11*(0.29)	3.23(0.45)	0.02(0.01)	1.53
203	200	0.878*(0.003)	0.868(0.003)	0.855(0.004)	0.81*(0.23)	2.63(0.35)	0.08(0.03)	3.25
395	100	0.801(0.006)	0.804(0.006)	0.658(0.013)	2.38*(0.3)	5.35(0.71)	0.08(0.03)	2.25
395	200	0.882*(0.002)	0.865(0.003)	0.855(0.004)	1.03*(0.15)	2.68(0.3)	0.13(0.04)	2.60
s=12								
203	100	0.797(0.005)	0.798(0.007)	0.611(0.01)	3.02*(0.31)	7.54(0.77)	0.05(0.02)	2.50
203	200	0.864(0.004)	0.856(0.004)	0.794(0.005)	1.75*(0.21)	5.14(0.52)	0.14(0.04)	2.94
395	100	0.763(0.006)	0.757(0.007)	0.615(0.01)	2.79*(0.3)	6.94(0.86)	0.07(0.03)	2.49
395	200	0.858(0.004)	0.854(0.003)	0.783(0.006)	2.6*(0.3)	6.79(0.53)	0.09(0.03)	2.61
s=20								
203	100	0.78(0.006)	0.785(0.007)	0.567(0.008)	4.46*(0.36)	8.96(0.72)	0.07(0.03)	2.01
203	200	0.859(0.004)	0.858(0.004)	0.705(0.008)	3.64*(0.46)	9.08(0.66)	0.08(0.03)	2.49
395	100	0.754(0.006)	0.77*(0.006)	0.572(0.009)	4.26*(0.4)	10.02(0.83)	0.08(0.03)	2.35
395	200	0.842(0.003)	0.843(0.004)	0.695(0.008)	4.76*(0.44)	12.16(0.82)	0.11(0.03)	2.55

*Significantly better (larger for AUC and smaller for FP) with p-value<0.05 using one-sided two-sample t-test to compare PROMISE and CV

Table 4: Simulation results using the elastic net: mean and standard error (in parentheses) of (a) AUC and (b) # of false positives(FP), and (c) the ratio of mean FP using PROMISE to mean FP using CV

		(a) Prediction Accuracy (AUC)			(b) The number of false positives			(c) FP Ratio
p	n_{train}	PROMISE	CV	SS	PROMISE	CV	SS	CV/PROMISE
Larger signals								
s=6								
203	100	0.946*(0.003)	0.907(0.013)	0.859(0.008)	3.53*(0.39)	6.11(0.57)	0.06(0.02)	1.73
203	200	0.985*(0.001)	0.89(0.018)	0.97(0.002)	0.92*(0.15)	3.28(0.38)	0.08(0.03)	3.57
395	100	0.943*(0.003)	0.89(0.013)	0.847(0.009)	3.95*(0.39)	7.42(0.73)	0.14(0.04)	1.88
395	200	0.982*(0.001)	0.879(0.019)	0.976(0.001)	1.03*(0.19)	4.21(0.54)	0.18(0.05)	4.09
s=12								
203	100	0.885*(0.005)	0.864(0.012)	0.71(0.01)	8.76*(1.33)	12.56(1.08)	0.08(0.03)	1.43
203	200	0.955*(0.002)	0.903(0.014)	0.872(0.005)	2.56*(0.3)	7.9(0.52)	0.11(0.03)	3.09
395	100	0.856(0.006)	0.843(0.011)	0.713(0.009)	7.94*(0.88)	15.45(1.7)	0.16(0.04)	1.95
395	200	0.954*(0.002)	0.915(0.012)	0.887(0.004)	2.93*(0.34)	10.41(0.76)	0.06(0.02)	3.55
s=20								
203	100	0.858(0.005)	0.839(0.012)	0.608(0.01)	11.17(1.43)	13.82(1.06)	0.14(0.03)	1.24
203	200	0.931*(0.002)	0.892(0.013)	0.75(0.008)	5.52*(0.86)	13.99(1.08)	0.11(0.04)	2.53
395	100	0.817(0.006)	0.827(0.01)	0.608(0.01)	9.05*(0.94)	17.46(1.41)	0.1(0.03)	1.93
395	200	0.915*(0.003)	0.873(0.013)	0.762(0.006)	5.69*(0.7)	18.05(1.84)	0.24(0.05)	3.17
Smaller signals								
s=6								
203	100	0.826*(0.005)	0.771(0.013)	0.721(0.012)	4.61(1.11)	6.38(1.21)	0.1(0.03)	1.38
203	200	0.877*(0.003)	0.753(0.018)	0.846(0.005)	0.91*(0.2)	2.98(0.65)	0.22(0.05)	3.27
395	100	0.809*(0.006)	0.767(0.012)	0.676(0.012)	6.23(2.55)	10.18(1.86)	0.19(0.04)	1.63
395	200	0.881*(0.003)	0.78(0.016)	0.856(0.004)	1.24*(0.19)	3.92(0.74)	0.19(0.04)	3.16
s=12								
203	100	0.799*(0.005)	0.768(0.01)	0.615(0.01)	7.11(1.91)	9.7(1.15)	0.1(0.03)	1.36
203	200	0.864*(0.003)	0.797(0.014)	0.788(0.006)	1.96*(0.24)	5.37(0.6)	0.16(0.04)	2.74
395	100	0.768(0.006)	0.756(0.009)	0.627(0.01)	7.16*(1.12)	13.91(2.05)	0.09(0.04)	1.94
395	200	0.858*(0.003)	0.802(0.013)	0.779(0.006)	3.67*(0.59)	7.82(0.8)	0.1(0.03)	2.13
s=20								
203	100	0.782(0.006)	0.791(0.008)	0.574(0.008)	8.23*(0.97)	15.74(1.42)	0.09(0.03)	1.91
203	200	0.86*(0.003)	0.815(0.013)	0.695(0.008)	3.76*(0.39)	10.88(0.87)	0.13(0.03)	2.89
395	100	0.756(0.005)	0.755(0.01)	0.582(0.009)	11.76*(1.95)	17.75(1.72)	0.15(0.04)	1.51
395	200	0.845*(0.004)	0.818(0.01)	0.682(0.008)	5.47*(0.62)	15.86(1.7)	0.16(0.04)	2.90

*Significantly better (larger for AUC and smaller for FP) with p-value<0.05 using one-sided two-sample t-test to compare PROMISE and CV

7 BATTLE data analysis

Data set and pre-processing . In this section, we analyze real data from the BATTLE clinical trial. This analysis is similar to the simulated data analysis in Section 6.1, except that the treatment response is obtained from the real clinical trial. The goals of this section are to identify the important biomarkers and their biological meaning, and to predict the treatment outcome based on the selected markers.

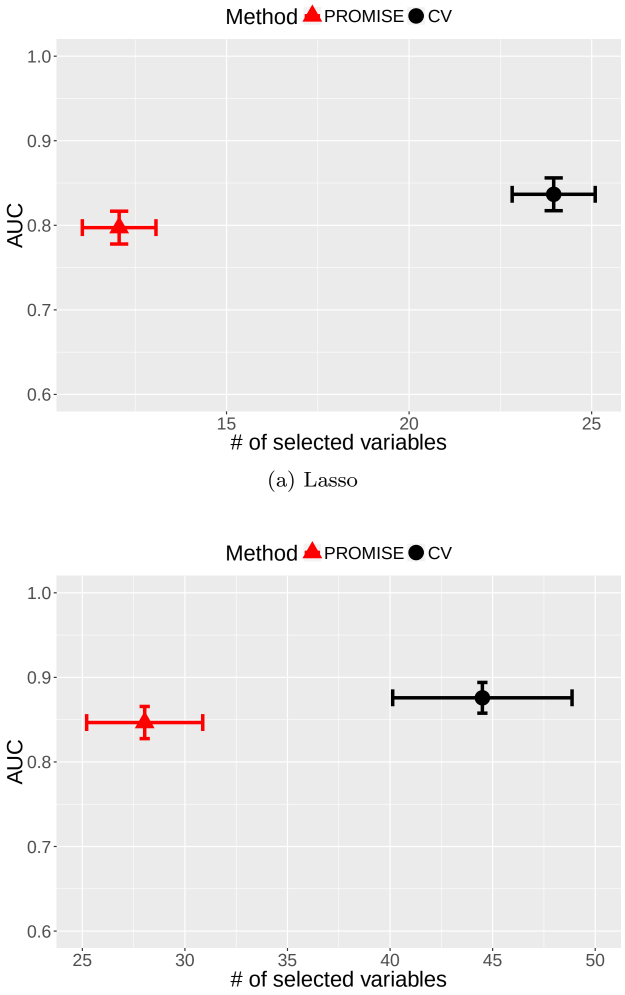
We use the same subset of microarray data as in Section 5. We add the clinical covariates and candidate predictive markers that were used to assign patients to a treatment in the trial. There were 13 clinical covariates and 14 markers, and we pre-screened those variables. We find that 2 clinical covariates and 6 markers were statistically significant in predicting the disease control status at level $\alpha = 0.05$, either by Kim et al.[**?**] or by a z-test in a simple logistic regression model, which consists of a single variable and its interaction effects with the treatments. These pre-selected markers are EGFR mutation, EGFR amplification, Kras/Braf marker group, VEGFR-2 expression, RXR beta expression, and cyclin D1 expression. The pre-selected clinical covariate is histology, which consists of adenocarcinoma, squamous cell carcinoma, and others. Since it has 3 categories, 2 indicator variables are created for this covariate.

As a result, we pre-selected 98 probe sets and 8 clinical + biomarker variables. Therefore, the total number of variables is $(98 + 8) \times 4 + 3 = 427$. We apply CV, SS, and PROMISE with the lasso and the elastic net to this data set to select significant markers and predict the 8-week disease control rate. We split the data into a training set and a test set as described in Section 6.1. We perform random division 100 times to evaluate the performance of each method.

Results Figure 3 summarizes the prediction and variable selection results. Since SS with the default cutoff essentially does not select any variable (on average 0.04 variable selected), we remove it from the graph. The figure shows that although PROMISE predicts as well as CV, PROMISE produces a much more parsimonious solution than CV: PROMISE selects almost half (12) the number of variables selected by CV (24) using the lasso and 60% less variables (28) than CV (45) using the elastic net.

Table 5 shows the gene symbols/probe set identifications (IDs) that are selected by PROMISE with the elastic net at least 20 times, which are sorted by the number of selections among 100 runs. The original microarray data included only probe set IDs for the probe sets; however, when a gene can be matched with a probe set ID, a gene symbol is listed in the table.

Biological Interpretation We checked whether any association exists between the type of cancer and individual genes that are selected with high probability by PROMISE using the elastic net. We used Ingenuity® Systems (www.ingenuity.com) to conduct the gene search. Even though the resource



(a) Lasso

(b) Elastic net

Fig. 3: Comparison of PROMISE (triangle/red) with CV (circle/black) for (a) lasso and (b) elastic net using the BATTLE data. Each point represents the mean value; error bar represents 2 standard errors.

does not distinguish whether the genes are predictive or prognostic markers, we explored the biological implications of our findings.

The most frequently selected gene, TXNDC12 (thioredoxin domain containing 12), is closely related to cancer. It is one of two members of protein-disulfide reductase (glutathione). Since high glutathione levels protect tumor cells, a targeted therapy that lowers glutathione levels can protect normal cells while allowing tumor cells to be sensitive to chemotherapy. This anti-

Table 5: List of gene/probe sets selected from the BATTLE trial analysis.

Gene/Probe Symbol	Type	Probability
TXNDC12	Vandetanib	0.95
ZNF334	Sorafenib	0.92
SLITRK6	Vandetanib	0.85
GAGE12B/GAGE88	Erlotinib+bexarotene/Marginal	0.82/0.23
7893300	Sorafenib	0.81
8003846	Vandetanib	0.79
AGPAT4	Sorafenib	0.76
7896225	Sorafenib/Vandetanib	0.71/0.39
8088915	Vandetanib	0.70
7894855	Erlotinib+bexarotene	0.70
RFT1	Sorafenib	0.70
SLC16A12	Sorafenib	0.66
7896557	Erlotinib+bexarotene/Vandetanib	0.65/0.28
SLC1A4	Vandetanib	0.61
UNC80	Vandetanib	0.61
ZNF674	Vandetanib/Sorafenib	0.60/0.37
8113784	Vandetanib	0.57
PRH1	Sorafenib	0.57
HCP5	Marginal	0.52
7895065	Sorafenib	0.52
Olfr1288	Marginal/Erlotinib+bexarotene/Vandetanib	0.49/0.32/0.30
7893401	Marginal/Vandetanib	0.49/0.42
RPL38	Vandetanib	0.46
NPY5R	Vandetanib	0.45
CRABP2	Erlotinib+bexarotene/Marginal	0.34/0.34
ACTR6	Vandetanib	0.29
GABRB1	Vandetanib	0.28
7892819	Vandetanib	0.24
RRP7A	Sorafenib	0.24
TMX3	Vandetanib	0.23
DPYSL5	Marginal	0.22
KDM3A	Vandetanib	0.21
SNX32	Vandetanib	0.21
RXRB	Vandetanib	0.21
CTRC	Vandetanib	0.20

Genes/probe sets that are selected more than 20 times among 100 runs by PROMISE with the elastic net. This list is sorted by the probability of selection among 100 runs. A gene symbol begins with a letter and a probe set ID begins with a number. The type indicates whether each gene/probe set has a marginal effect or an interaction effect with a treatment indicator variable: if it has an interaction effect with a treatment indicator variable, the treatment name is shown. Selection probability indicates how many times each variable is selected among the 100 runs.

neoplastic therapeutic approach has been tested in adenocarcinoma and in ovarian and breast cancer [**?**]. Also, glutathione disulfide reductase, which catalyzes the increase of glutathione [**?**], has been approved as a predictive marker for carmustine treatment of multiple myeloma.

Table 6 shows that the 9 genes most frequently selected by PROMISE with the elastic net are related to cancer. In particular, 5 of the 9 genes are related to adenocarcinoma, which is the most common cancer type in non-small cell

Table 6: Top 9 most frequently selected genes from the BATTLE data analysis and their association with cancer.

Gene Symbol	Associated Cancer
TXNDC12	adenocarcinoma, breast cancer, ovarian cancer [?]
ZNF334	adenocarcinoma, endometrioid carcinoma, melanoma
SLITRK6	adenocarcinoma, endometrioid carcinoma, melanoma
GAGE12B/GAGE8	bone marrow cancer [?]
AGPAT4	melanoma, endometrioid carcinoma
RFT1	melanoma
SLC16A12	melanoma, endometrioid carcinoma
SLC1A4	adenocarcinoma [?], bone cancer cell lines, colon cancer cell lines
UNC80	adenocarcinoma, melanoma, endometrioid carcinoma

lung cancer. (Note that the BATTLE trial was for patients with non-small cell lung cancer).

In addition, we conducted Ingenuity pathway analysis (Ingenuity® Systems: www.ingenuity.com) to gain biological insight into the functional roles of the selected genes in signaling pathways. Figure 4 shows the significant pathways to which the selected genes belong.

Many of these pathways are related to lung cancer. Non-small cell lung cancer signaling and small cell lung cancer signaling are directly related to lung cancer. Granulocyte-macrophage colony-stimulating factor (GM-CSF) signaling reduces tumor proliferation and invades lung cancer cells [?]. Activation of the vitamin D receptor (VDR) and retinoid X receptor (RXR) pathway mediates calcitriol, which inhibits tumor growth in lung cancer [?]. The aryl hydrocarbon receptor pathway has been suggested as a biomarker for lung cancer [?]. Down-regulation of Gadd45 expression is associated with tumor differentiation in non-small cell lung cancer [?]. Inhibition of Wnt reduces the proliferation of non-small cell lung cancer cell lines [?]. Many of the other significant pathways are related to cancer, such as thyroid cancer signaling, glioma signaling, and molecular mechanisms of cancer.

These findings indicate that the genes frequently selected by PROMISE with the elastic net have biological plausibility, especially in their association with lung cancer or adenocarcinoma. These findings suggest that the genes shown in Table 6 in particular are possible predictive markers for non-small cell lung cancer.

8 Discussion

We have proposed PROMISE as a regularization parameter selection method to select both prognostic and predictive markers for personalized medicine. We compared PROMISE, CV, and SS using simulated and real data. Our results show that PROMISE outperforms CV and SS in the following ways: (1) PROMISE predicts the treatment outcome better than or similar to CV; (2) PROMISE selects far fewer noise variables than CV with a marginal decrease in true positives; (3) PROMISE selects variables more accurately than CV,

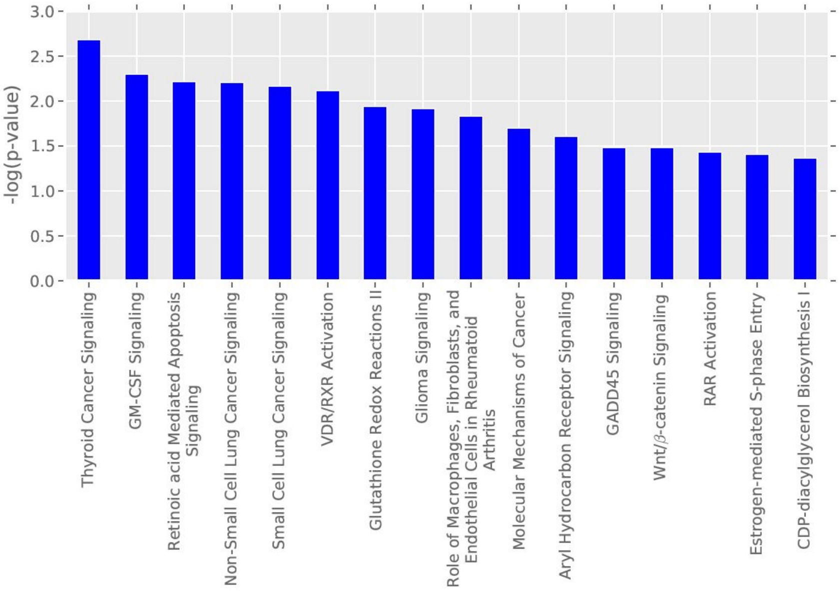


Fig. 4: Canonical pathways enriched for genes selected from the BATTLE data analysis

These pathways are significant (p-value < 0.05) using Fisher's exact test, which identifies whether the selected genes are in each pathway by chance. The pathways are sorted by significance.

producing fewer type I + type II errors than CV; and (4) PROMISE produces a much sparser solution than CV. The prediction accuracy of SS has some variability depending on the data set, but is consistently lower than that of PROMISE because it often selects too few variables to have good prediction accuracy.

CV tends to select too many variables to ensure good prediction accuracy. On the other hand, SS tends to select too few variables to control false discoveries. In essence, PROMISE strikes a balance between the two, having good prediction accuracy and yielding few false positives. This is attributed to the following characteristics of PROMISE: first, the sub-sampling method of PROMISE plays a role in reducing false discoveries. Second, using the CV method in PROMISE to flexibly select the cutoffs ($\lambda_{\min}$, π_{thr}) plays a role in maximizing the prediction accuracy and therefore includes important variables for that purpose.

We show PROMISE is an effective method for selecting the regularization parameters in high-dimensional variable selection settings, e.g., genomic data. For biomarker identification, it significantly reduces false positives so that the cost of biological experimental verification can be reduced, while maintaining good prediction accuracy so that each individual can receive an optimal personalized treatment. PROMISE can be applied to many parameter selection problems when the goals are both to minimize false discoveries/induce sparsity and to maximize the prediction accuracy/true discoveries.

In this article, we applied PROMISE to binary outcomes; however, the method can be generalized to survival outcomes or continuous outcomes. For example, one can use a penalized Cox proportional hazards model for survival outcomes or a penalized linear regression for continuous outcomes and apply PROMISE to select the penalty term(s). We are currently working on making an R package for binary, survival, and continuous outcomes. Moreover, we applied PROMISE along with the lasso and elastic net, but it can be applied with other statistical learning methods to select regularization parameter(s) in more complex settings such as grouped variable selection. We leave these tasks for future exploration.

Acknowledgements This work was supported by grants P30 CA016672 from the National Cancer Institute (JJL). We thank Ms. Lee Ann Chastain for helping to edit this manuscript.