A general kernel boosting framework to integrate pathways for cancer genomic data analysis

Li Zeng[1], **Zhaolong Yu**[2], **Yiliang Zhang**[1] **and Hongyu Zhao**[1,2,3,*]

[1] Department of Biostatistics, Yale University, New Haven, CT 06511, USA
[2] Interdepartmental Program in Computational Biology and Bioinformatics, Yale University, New Haven, CT 06511, USA
[3] Department of Genetics, Yale School of Medicine, New Haven, CT 06510, USA
[*] Correspondence: hongyu.zhao@yale.edu

Abstract

Keywords:

1 Introduction

In last chapter, we have proposed a pathway-based boosting (PKB) algorithm that can utilize cancer patients' gene expression data and known pathway information to classify patients into different clinical groups. It has been applied to several cancer study datasets in The Cancer Genome Atlas (TCGA) to predict various clinical outcomes, such as tumor stage, grade, and metastasis status. However, two drawbacks of the model may limit the usefulness of it in cancer data analysis. First of all, typical cancer genomic study datasets provide both clinical features and genomic features. In PKB, only gene expressions are considered as predictors, which is a waste of the clinical features that are potentially predictive of the outcome variable. Second, the current PKB is only able to predict categorical outcome variables, which excludes the analysis for interesting clinical variables, such as drug response, disease free survival, and overall survival.

In this chapter, we propose a modified PKB framework. It enables the inclusion of clinical features as predictors by adding a linear model part to the base learner spaces. It also unifies classification, regression, and survival analysis under the same Boosting procedure, which can handle different types of outcome variables by specifying different loss functions. We focus on the applications of the new PKB model to regression and survival analysis, and examine its performance in two databases: the Cancer Cell Line Encyclopedia (CCLE) for cell line drug response prediction, and TCGA for patient survival prediction.

2 Materials and methods

Suppose our observed data is collected from N subjects. For subject i, we use a p dimensional vector $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{ip})$ to denote the normalized gene expression profile. Similarly, the gene expression levels of a given pathway m with p_m genes can be represented by $\mathbf{x}_i^{(m)} = (x_{i1}^{(m)}, x_{i2}^{(m)}, \dots, x_{ip_m}^{(m)})$, which is a sub-vector of $\mathbf{x}_i$. We use $\mathbf{z}_i = (z_{i1}, z_{i2}, \dots, z_{iq})$ to represent available clinical features for the patient, such as gender, age, and cancer type.

For classification, each subject has an observed class label $y_i \in \{1, -1\}$. The probability of being class 1 is modeled as

$$p(y = 1|\mathbf{x}, \mathbf{z}) = \frac{\exp[F(\mathbf{x}, \mathbf{z})]}{1 + \exp[F(\mathbf{x}, \mathbf{z})]},$$

where $F(\mathbf{x}, \mathbf{z})$ is the log odds function of classification model. Maximizing the likelihood is equivalent to minimizing the following log loss function:[1]

$$l(y, F(\mathbf{x}, \mathbf{z})) = \log(1 + \exp[-yF(\mathbf{x}, \mathbf{z})]).$$

In the regression model, we observe continuous outcome y for each subject. The loss function for regression is the widely used squared error:

$$l(y, F(\mathbf{x}, \mathbf{z})) = (y - F(\mathbf{x}, \mathbf{z}))^2$$

In the survival model, outcome for each subject is a bivariate tuple $y = (t, \delta)$, where t is the survival time or censoring time, and δ is an indicator of endpoint event, such as disease relapse or death in most cancer studies. Therefore, t is the actual survival time when $\delta = 1$, and censoring time if $\delta = 0$. We build the survival model following the Cox regression's assumption on the hazard function, but replacing the linear part with a nonlinear risk score function $F(\mathbf{x}, \mathbf{z})$:[2]

$$h(t, F(\mathbf{x}, \mathbf{z})) = h_0(t) \exp[F(\mathbf{x}, \mathbf{z})] \tag{1}$$

where $h_0(t)$ is an unknown baseline hazard function. Using the partial likelihood function enables us to circumvent the inference of $h_0(x)$, and directly estimate $F(\mathbf{x}, \mathbf{z})$. We use the negative log partial likelihood[2] as the loss function in the survival model:

$$l(y, F(\mathbf{x}, \mathbf{z})) = -\delta \left\{ F(\mathbf{x}, \mathbf{z}) - \log \left(\sum_{j=1}^{N} 1_{\{t_j \geq t\}} \exp[F(\mathbf{x}, \mathbf{z})] \right) \right\}.$$

The goal of PKB is to estimate $F(\mathbf{x}, \mathbf{z})$ non-parametrically through a Boosting procedure. $F(\mathbf{x}, \mathbf{z})$ has different interpretations in different loss functions. We will refer to it as the score function in what follows.

2.1 Base learner space

Results from theories of Reproducing Kernel Hilbert Space (RKHS)[3] have shown that kernel functions can capture complex interactions among features. It has been used in Zeng et al[1] to capture predictive genomic information for classification purpose. We extend it by incorporating clinical features in the base learner space. For pathway m, the base learner space takes the following form:

$$\mathcal{G}_m = \{f(\mathbf{x}, \mathbf{z}) = \sum_{i=1}^{N} K_m(\mathbf{x}_i^{(m)}, \mathbf{x}^{(m)})\beta_i + \sum_{j=1}^{q} z_j\gamma_j : \beta \in R^N, \gamma \in R^q\}, \tag{2}$$

where $K_m(\cdot, \cdot)$ is a kernel function that calculates similarity between two subjects using only genes in the mth pathway. Each element in the space is composed of two parts: a nonlinear component to model gene expression effect and a linear component to model clinical features' effect. The overall base learner space $\mathcal{G}$ is simply the union of individual learner spaces from each pathway: $\mathcal{G} = \bigcup_{m=1}^{M} \mathcal{G}_m$.

Assume there are M pathways considered in our model. Since genes within the same pathway likely have much stronger interactions than genes in different pathways, in our pathway-based model setting, we assume additive effects across pathways and focus on capturing gene interactions within pathways:

$$F(\mathbf{x}, \mathbf{z}) = \sum_{m=1}^{M} H_m(\mathbf{x}^{(m)}, \mathbf{z}),$$

where each H_m is a function that only depends on the expression level of genes in the mth pathway and the clinical features. H_m belongs to the RKHS of the mth pathway $\mathcal{G}_m$. Due to the additive nature of this model, it only captures gene interactions within each pathway but not across pathways.

Given a problem and the corresponding loss function, the empirical loss is defined as the mean of losses evaluated at each sample point:

$$L(\mathbf{y}, \mathbf{F}) = \frac{1}{N}\sum_{i=1}^{N} l(y_i, F(\mathbf{x}_i, \mathbf{z}_i)),$$

where $\mathbf{F} = (F(\mathbf{x}_1, \mathbf{z}_1), F(\mathbf{x}_2, \mathbf{z}_2), \ldots, F(\mathbf{x}_N, \mathbf{z}_N))$. In the rest of this article, we will also use boldface font of a function to represent the vector of the function evaluated at each observed sample point.

2.2 Identify the optimal increment function

Boosting is an iterative functional descent procedure to minimize the empirical loss function. Assume that at iteration t, the estimated target function is $F_t(\mathbf{x}, \mathbf{z})$. The most crucial step in the Boosting procedure is the identification of an "optimal" increment function $f \in \mathcal{G}$, which shrinks the empirical loss as much as possible, given the current value of $F_t(\mathbf{x}, \mathbf{z})$ and add it to $F_t(\mathbf{x}, \mathbf{z})$. We approximate $L(\mathbf{y}, \mathbf{F}_t)$ by its second order Taylor expansion around $\mathbf{F}_t$:

$$\begin{aligned} L(\mathbf{y}, \mathbf{F}_t + \mathbf{f}) &\approx L(\mathbf{y}, \mathbf{F}_t) + (\nabla_{\mathbf{F_t}} L)^T \mathbf{f} + \frac{1}{2}\mathbf{f}^T \mathbf{H}_{\mathbf{F_t}} \mathbf{f} \\ &= \frac{1}{2}(\mathbf{f} + \mathbf{H}_{\mathbf{F_t}}^{-1} \nabla_{\mathbf{F_t}} L)^T \mathbf{H}_{\mathbf{F_t}} (\mathbf{f} + \mathbf{H}_{\mathbf{F_t}}^{-1} \nabla_{\mathbf{F_t}} L) + \text{const}, \end{aligned}$$

where f is the increment function. $\nabla_{\mathbf{F_t}} L$ and $\mathbf{H}_{\mathbf{F_t}}$ are the gradient and Hessian matrix of $L(\mathbf{y}, \mathbf{F_t})$ with respect to $\mathbf{F_t}$, respectively, and const includes all the terms that do not involve $\mathbf{f}$. Note that this approximation is accurate in the case of regression, since the regression loss is quadratic itself. Calculations of $\nabla_{\mathbf{F_t}} L$ and $\mathbf{H_{F_t}}$ for different problems are available in Appendix A in the Supplementary Materials. Direct minimization for this loss approximation in $\mathcal{G}$, however, may lead to overfitting, due to the high flexibility of the base learner space. It is necessary to apply penalties in the selection of f. Therefore we propose the following regularized loss as the working loss function in PKB:

$$L_R(f) = \frac{1}{2}(\mathbf{f} + \mathbf{H}_{\mathbf{F_t}}^{-1}\nabla_{\mathbf{F_t}} L)^T \mathbf{H}_{\mathbf{F_t}}(\mathbf{f} + \mathbf{H}_{\mathbf{F_t}}^{-1}\nabla_{\mathbf{F_t}} L) + \lambda\Omega(f),$$

where $\Omega(f)$ is the penalty term. Since f takes the functional form as presented in (2), it is natural to consider the L_1 and L_2 norm of β as choices of penalty: $\Omega(f) = \|\beta\|_1$ or $\|\beta\|_2^2$. Such a penalized boosting step has been employed in sevaral methods (e.g., Johnson et al).[4] Intuitively, the regularized loss function would prefer simple solutions that also fit the observed data well, which usually leads to better generalization capability to unseen data.

Optimizing $L_R(f)$ in $\mathcal{G}_m$ is equivalent to

$$\min_{\beta,\gamma} \frac{1}{2}(K_m\beta + Z\gamma + \mathbf{H}_{\mathbf{F}}^{-1}\nabla_{\mathbf{F}} L)^T \mathbf{H}_{\mathbf{F}}(K_m\beta + Z\gamma + \mathbf{H}_{\mathbf{F}}^{-1}\nabla_{\mathbf{F}} L) + \lambda\Omega(f), \tag{3}$$

where K_m is the $N \times N$ kernel matrix calculated using gene expressions in the mth pathway, and Z is the $N \times q$ clinical feature matrix. It can be proved that, by applying the following transformation:

$$\begin{aligned} \tilde{\eta} &= \frac{1}{\sqrt{2}}\mathbf{H}_{\mathbf{F}}^{\frac{1}{2}}(I_N - Z(Z^T\mathbf{H}_{\mathbf{F}}Z)^{-1}Z^T\mathbf{H}_{\mathbf{F}})\mathbf{H}_{\mathbf{F}}^{-1}\nabla_{\mathbf{F}} L \\ \tilde{K}_m &= \frac{1}{\sqrt{2}}\mathbf{H}_{\mathbf{F}}^{\frac{1}{2}}(I_N - Z(Z^T\mathbf{H}_{\mathbf{F}}Z)^{-1}Z^T\mathbf{H}_{\mathbf{F}})K_m, \end{aligned}$$

solving β in Equation (3) is reduced to:

$$\min_{\beta} \|\tilde{\eta} + \tilde{K}_m\beta\|_2^2 + \lambda\Omega(f). \tag{4}$$

Proof is provided in Appendix B in the Supplementary Materials. Equation (4) becomes the LASSO problem when we use the L_1 penalty, and the Ridge Regression when we use L_2 penalty. Both can be efficiently solved with existing solvers. After solving β, we can subsequently obtain the solution of γ as:

$$\gamma = -(Z^T\mathbf{H}_{\mathbf{F}}Z)^{-1}Z^T\mathbf{H}_{\mathbf{F}}(K_m\beta + \mathbf{H}_{\mathbf{F}}^{-1}\nabla_{\mathbf{F}} L).$$

2.3 The PKB algorithm

In this section we propose the PKB algorithm that solves classification, regression, and survival analysis in a unified framework.

1. Initialization
   Let $F_t(\mathbf{x}, \mathbf{z})$ be the estimated score function at time t. Initialize $F_0(\mathbf{x}, \mathbf{z})$ as a constant that minimizes the empirical loss:

   $$F_0(\mathbf{x}, \mathbf{z}) = \arg\min_c \frac{1}{N}\sum_{i=1}^{N} l(y_i, c).$$

   In the case of survival model, we set $F_0(\mathbf{x}, \mathbf{z}) = 0$, because the partial likelihood of Cox model is not affected by constants.

2. Identify the optimal increment function f
   At iteration t, we calculate the gradient $\nabla_{\mathbf{F_t}} L$ and Hessian matrix $\mathbf{H}_{\mathbf{F_t}}$. For each pathway m, we solve the optimal $\hat{f}_m \in \mathcal{G}_m$ and corresponding β, γ following the steps in Section 2.2. The optimal increment function $\hat{f}$ is the $\hat{f}_m$ which yields the smallest $L_R(f)$.

3. Line search and update F_t
   While $\hat{f}$ gives the direction in which the regularized loss has the steepest descent, we still need to decide the step length. We first perform a line search

   $$\hat{d} = \arg\min_{d \in R^+} L(\mathbf{y}, \mathbf{F_t} + d\hat{\mathbf{f}}).$$

   The solution $\hat{d}$ represents the step size to reach the bottom of the empirical loss following the direction of $\hat{f}$. We then apply a shrinkage on the step size using a learning rate parameter ν, and update our estimate of the score function:

   $$F_{t+1}(\mathbf{x}, \mathbf{z}) = F_t(\mathbf{x}, \mathbf{z}) + \nu \hat{d} \hat{f}(\mathbf{x}, \mathbf{z}).$$

   The learning rate parameter ν takes value in $(0, 1)$, usually smaller than 0.1. Our experience shows that the combination of the line search technique and the learning rate parameter makes the model fitting procedure more stable, and less prone to overfitting.

4. Repeat step 2 and 3 until T iterations. The final $F_T(\mathbf{x}, \mathbf{z})$ is the estimated score function.

The choice of T is critical to prediction accuracy on test data. T being to small or too large can lead to underfitting and overfitting, respectively. We employ a cross-validation procedure to determine the number of iterations T. We split the dataset into three folds, and simultaneously initiate three runs of the Boosting algorithm, each using two folds as training data and the other fold as testing data. After each iteration, a cross-validated loss is calculated by averaging the three testing loss values. We keep track of the minimum cross-validated loss. If its value does not change in 50 iterations, we end the cross-validation process, and choose the iteration with the minimum loss as the number of iterations T.

After fitting the PKB model, the final score function estimate takes the form

$$F_T(\mathbf{x}, \mathbf{z}) = \sum_{m=1}^{M} \sum_{i=1}^{N} K_m(\mathbf{x}_i^{(m)}, \mathbf{x}^{(m)}) \beta_i^{(m)} + \mathbf{z}^T \gamma,$$

where $\beta^{(m)}$ is the coefficient vector for pathway m. The values of $\beta^{(m)}$s can be used to evaluate the significance of pathways in the score function. We propose to use the L_2 norm, $w_m = \|\beta^{(m)}\|_2$, as weights for pathways. Note that w_m is non-zero only if the pathway is selected at least once during model fitting. From our experience, in applications with large number of input pathways, many pathways will end up with zero weights.

3 Simulation Study

In this section, we apply PKB to a variety of simulation datasets, and demonstrate that it can yield better prediction accuracy compared to competing methods, as well as correctly identify informative pathways.

We design the following three models for the underlying truth score functions $F(\mathbf{x}, \mathbf{z})$:

- Model 1:

$$F(\mathbf{x}, \mathbf{z}) = 3z_1 - 4z_2 + 3z_3 + 2x_1^{(1)} + 3x_2^{(1)} + 3\exp(0.5x_1^{(2)} + 0.5x_2^{(2)}) + 4x_1^{(3)}x_2^{(3)}$$

- Model 2:

$$F(\mathbf{x}, \mathbf{z}) = z_1 - 3z_2 + 3z_3 - z_4 + 6\sin(0.5x_1^{(1)} + 0.5x_2^{(1)}) + 2\log(|x_1^{(2)^3} - x_2^{(2)^3}|) + 2(x_1^{(3)^2} - x_2^{(3)^2})$$

- Model 3:

$$F(\mathbf{x}, \mathbf{z}) = z_1 + z_3 + 2\sum_{m=1}^{8} \|\mathbf{x}^{(m)}\|_2$$

In the above equations, $x_i^{(m)}$ represents the expression level for the ith gene in pathway m. The three models involve a wide variety of functional forms of pathway effects, including linear, polynomial, exponential, logarithm, and sine effects. We assume the effects from clinical variables are linear.

Under each model, we simulate two datasets,with 20 and 50 pathways, respectively. Note that in model 1 and 2, only the first 3 pathways are informative, and in model 3, the first 8 pathways are informative. The predictive signals in the datasets with 50 pathways are much sparser than the 20-pathway datasets. For each pathway, we simulate expression for 5 genes using Gaussian distribution. In each dataset, we also generate 5 clinical features: 2 binary features generated from Bernoulli distribution, and 3 continuous features generated from Normal distribution. The number of predictive clinical variables are 3, 4, and 2 for the three models, respectively. The sample sizes for all datasets are 300.

The outcome values $\mathbf{y}$ for regression and survival models are generated from different mechanisms. For regression model, we add a Gaussian noise to the $F(\mathbf{x}, \mathbf{z})$ values to generate $\mathbf{y}$. The variance of the Gaussian noise is set to the 1/5 of the $F(\mathbf{x}, \mathbf{z})$ values' variance.

For simulation of survival outcomes, we assume a Weibull baseline hazard $h_0(t) = \kappa\rho t^{\rho-1}$, and cumulative hazard function $H_0(t) = \kappa t^{\rho}$, where κ and ρ are the scale and shape parameters, respectively. Suppose the score $F(\mathbf{x}, \mathbf{z})$ is calculated for one sample. A corresponding survival time can be generated from

$$t = \left(-\frac{\log(U)}{\kappa \exp[F(\mathbf{x}, \mathbf{z})]}\right)^{\frac{1}{\rho}},$$

where U is randomly drawn from Uniform$(0, 1)$ distribution.[5] The values of κ and ρ are chosen such that the median survival time is 20 (months), which is on the same scale of median survival times of many cancer types. We then randomly draw 20% samples for censoring, and the censoring times are drawn from a uniform distribution between zero and the generated survival times.

When evaluating prediction performance, we use mean square error (MSE) for regression, and C-index for survival model.[6] C-index is commonly used in assessing survival prediction accuracy. In general, it looks at all possible pairs of samples, and calculate the ratio of the pairs where the predicted risk scores are concordant with the observed survival times. If the predicted risk score is not informative at all, the C-index would be about 0.5. More details regarding the calculation of C-index can be found in Appendix C in the Supplementary Materials. On each dataset, we perform 10 runs of the PKB algorithm. In each run, we use 2/3 of the samples as training data, and assess prediction performance on the remaining samples.

3.1 Regression model results

We compared the performance of the PKB regression model to several existing methods, including LASSO, Ridge Regression, and ElasticNet,[7] which are linear models, and RandomForest,[8] Gradient Boosting Regression (GBR),[9] and Support Vector Regression (SVR),[10] which are nonlinear models. We extensively tuned the parameters for all methods, and the details are provided in Appendix D in the Supplementary Materials. Table 1 presents the average MSE on test data over 10 runs for all methods. Standard deviations of the MSEs can be found in Appendix H in the Supplementary Materials.

In all simulation scenarios, the two PKB algorithms, PKB-L_1 and PKB-L_2, yielded significantly better prediction accuracy than competing methods. Among the competing methods, the sparse linear models had better accuracy in Model 1 and 2, where only three pathways are informative. Since there are eight pathways relevant to the outcome in Model 3, the nonlinear genomic signal becomes stronger, thus the nonlinear methods produced equal or better accuracy than the linear methods. We also assessed the ability of PKB to properly weigh the informative pathways. For each pathway, we calculated its weights in the final score function over the 10 runs. The distributions of the weights are presented in Figure 1. In all simulation scenarios, the PKB algorithm gave relevant pathways significantly higher weights than other pathways. Note that PKB shrank the weights of some, but not all the noise pathways to zero. This is expected, because as the Boosting procedure continues, the predictive effect from true informative pathways becomes weaker, and the noise pathways may accidentally result in the smallest loss and get selected in certain iterations.

3.2 Survival model results

In the survival analysis simulations, we compared our method with Glmnet,[11] RandomSurvivalForest,[12] and CoxBoost.[13] Glmnet is an extension of Cox regression model with penalties, and has been popular in analysis of survival data with high dimensional predictors. RandomSurvivalForest and CoxBoost are also extensions of Random Forest and CoxBoost, respectively, to perform survival analysis. Predictive performances were evaluated using C-index. The average C-index for each method over the 10 runs are presented in Table 2, and the standard deviations are available in Appendix H in the Supplementary Materials.

Both PKB methods significantly outperformed the competing methods in all simulation scenarios. We also examined the pathway weights, similar to what we did in the regression simulations. The results also indicated that the PKB survival model could effectively identify informative pathways and weigh them heavily in the score function.

Since the weights distribution figure has similar pattern to Figure 1 from regression simulations, we leave it in Appendix F in the Supplementary Materials.

4 Applications

In order to examine the performances of our proposed model on real datasets, we applied PKB, along with all the competing methods, to two cancer-related databases: the Cancer Cell Line Encyclopedia (CCLE)[14] for regression analysis, and The Cancer Genome Atlas (TCGA) for survival analysis.

We followed the same procedure as in simulation studies to assess the performances of the methods. In the applications of PKB, we considered the Kyoto Encyclopedia of Genes and Genomes (KEGG),[15] Biocarta,[16] and Gene Ontology Biological Process (GO-BP)[17,18] as sources of pathway information. Details about the choices of model parameters can be found in Appendix D in the Supplementary Materials.

4.1 CCLE drug response prediction

CCLE is a rich database containing cancer cell line responses to anti-cancer compounds. It involves cell lines from more than 20 cancer types, and more than 20 compounds with various targets. For about 1000 of the cell lines, RNAseq gene expression data is available, which allows the analysis of the association between genes and drug responses.

Drug response is measured by the IC50 value, which is defined as the concentration needed for the compound to kill 50% of the tumor cells in the cell line. The log-transformed IC50 value was used as the outcome variable for regression. For clinical predictors, we considered cell line cancer types and gender of the cell line provider. We applied our method to predict responses for 6 of the compounds (named after their corresponding targets) that have sufficient sample sizes: EGFR, HDAC, MEK, RAF, TOP, and TUBB1.

Table 3 demonstrates the cross-validated prediction MSEs from PKB and all competing methods, with top two methods marked in bold. In five out of the six datasets, at least one of the PKB methods appeared in the top two methods. In the MEK dataset, both PKB-L_1 and PKB-L_2 were ranked top two, and the difference in MSE, compared to competing methods, was significant (please see Appendix H in the Supplementary Materials for standard deviations of MSEs).

4.2 TCGA cancer patient survival prediction

In order to assess the predictive performance of PKB on real survival data, we applied the PKB survival model to 7 cancer study datasets, including (by primary tumor sites): brain, head/neck, skin, lung, kidney, stomach, and bladder. The datasets were selected using criteria such as large sample size with gene expression data and low censoring ratio. The TCGA datasets contain richer clinical information compared to the cell line datasets in CCLE. The clinical features we used generally included patient gender, age, tumor subtype, site/laterality, and stage, which were available in almost all datasets and had low missing rates.

The cross-validated prediction C-indices from all methods are presented in Table 4. The PKB methods achieved top two accuracy in five out of the seven datasets, and the C-index differences to competing methods were significant (please see Appendix H in the Supplementary Materials for standard deviations of the C-indices). In the remaining two cases, PKB also yielded comparable performances (C-index difference < 0.01, not significant) to the top two. In the Brain and Kidney datasets, PKB was most successful, beating the third best method by 0.26 and 0.15 in C-index, respectively.

We further performed pathway enrichment analysis, and examined the p-values of the pathways considered significant by PKB. (If a pathway took positive weights in at least four out of ten runs of PKB, it was considered significant.) The enrichment analysis was conducted on each dataset following the steps below:

1. For each gene, we performed a Cox regression using the clinical features and the gene's expression as predictors. P-values for each gene were calculated from the Cox models.

2. We chose 20% genes with the smallest p-values as relevant genes to patients' survival times.

3. The relevant genes were used to perform Fisher's exact test-based enrichment analysis[19] on the pathway database (KEGG, GO-BP, or Biocarta) used in the PKB model that achieved the highest accuracy. An enrichment p-value was calculated for each pathway.

The enrichment analysis results for the brain, kidney, and lung cancer datasets are presented in Figure 2. Results for other datasets are available in Appendix E in the Supplementary Materials. In the figure, we highlight the significant pathways from PKB using red bars and stars on the x-axes. We observe different enrichment patterns from different datasets. In the lung dataset (bottom panel of Figure 2), the PKB significant pathways are highly concentrated to the left, which are also significant pathways in the enrichment analysis. In the brain and kidney cancer, the PKB pathways are more spread out: some of them appear at the top, but many others are not considered significant by the enrichment analysis. This is not surprising, because enrichment analysis looks at marginal associations between pathways and outcomes, while PKB focus on additive effects. It is possible for pathways without strong marginal signals to be picked out by PKB at the existence of other pathways in the model.

4.3 Comparison with models using only clinical features

The PKB score function contains a linear model of clinical features and a nonlinear model of genomic features. It is natural to ask how much improvement the genomic part brings to the prediction accuracy in addition to the clinical part. Considering that genomic data is much more expensive to acquire compared to clinical data, it makes sense to acquire genomic data only when the prediction accuracy can indeed be improved. We compared the performances of the following three methods in both the CCLE and TCGA datasets: the PKB methods which utilize clinical, genomic, and pathway information; linear models with both clinical and genomic features; and linear models with only clinical features. The results are presented in Figure 3.

The upper panel of Figure 3 shows the MSEs of the three methods on CCLE drug response datasets. The "LM" method represents the best results from LASSO, Ridge Regression, and ElasticNet. In four datasets (EGFR, HDAC, MEK, and TOP1), using genomic information significantly improved the prediction MSE. In the MEK dataset, the MSE difference between the linear models with and without genomic features is not significant. However, using PKB can improve the accuracy significantly.

Comparison of the three models in the TCGA datasets is presented in the lower panel of Figure 3. Glmnet was used to fit the linear models. Only in the skin cancer dataset, using genomic information failed to offer predictive signals in addition to the clinical features. In all other six dataset, PKB achieved significantly higher C-indices than the clinical-only Glmnet. However, when the genomic features were modeled using Glmnet, the gain in C-indices became moderate, and even failed to outperform the clinical-only version on the stomach and bladder cancer datasets. Therefore, PKB seemed to be able to capture genomic signals more efficiently than Glmnet.

5 Discussion

In this article, we have extended the PKB framework proposed by Zeng et al[1] to perform regression and survival analysis. It has also enabled the algorithm to incorporate clinical features as predictors by adding a linear part to the base learner spaces. We have applied PKB to the CCLE database to predict cancer drug responses on cell lines, and to the TCGA database to predict cancer patients' survival. In both applications, PKB has achieved equal or superior prediction accuracy than competing methods in most datasets. Especially in several TCGA datasets, PKB has significantly improved the prediction accuracy by a large amount. We have further compared PKB with linear models that only use clinical predictors. Results have demonstrated that PKB can effectively capture genomic predictive signals in addition to clinical signals, and significantly improve prediction performance.

In the PKB regression model, the final score $F(\mathbf{x}, \mathbf{z})$ function is an estimate of the regression function, which can be used directly to predict the outcome variable. However, in the survival model, $F(\mathbf{x}, \mathbf{z})$ is not an estimate of the patient's survival time, but rather an estimate to the risk score in the hazard function (see Equation (1)). Since the baseline hazard $h_0(t)$ is still unknown, we cannot provide direct estimates of patients' survival times. Nonetheless, after an estimate for $F(\mathbf{x}, \mathbf{z})$ is acquired, it is easy to get a nonparametric estimate of the baseline hazard function, and subsequently estimate the survival times.

We have also assumed that the clinical effects and pathway effects are additive, therefore no interactions between clinical features and pathways are modeled. In the existence of categorical clinical features, informative pathways are supposed to be informative in all categories. This assumption, however, may be violated in real datasets. For example, each CCLE dataset is a mixture of cell lines from several cancer types. A pathway may be significant in one or two cancers, but it is hard to be predictive in all the cancer types. This is a probable reason why PKB has not made significant improvement in certain CCLE datasets. We have tried to fit PKB for each cancer type separately, but the small sample sizes make it difficult to pick out true pathway signals.

In the calculation of kernel functions, all the genes in the same pathways have been treated equally. It is possible that, by giving larger weights to important genes, the PKB models can achieve better prediction accuracy. Suppose gene i takes weight $w_i > 0$. We can modify the calculation of the kernel functions to focus more on the highly weighted genes, we propose the following modifications to the radial basis and polynomial kernel (degree d) functions, respectively:

$$K_{\mathrm{rbf}}(\mathbf{u},\mathbf{v}) = \exp\left[-\frac{\sum_j w_j(u_j - v_j)^2}{\sum_j w_j}\right], \quad K_{\mathrm{poly}}(\mathbf{u},\mathbf{v}) = \left(1 + \frac{\sum_j w_j u_j v_j}{\sum_j w_j}\right)^d.$$

We have explored this idea using gene weights calculated from GeneMANIA,[20] where we can acquire physical interaction network between genes. The edges between genes are annotated with interaction strength, and the total degrees of the genes are used as the weights. We have utilized the above weighted kernel function to fit PKB models. However, no significant difference in prediction performance has been observed compared to the unweighted version. Detailed results are available in Appendix G in the Supplementary Materials. We leave gene weights as an optional parameter when using PKB, so that users are able to explore different ways to calculate weights for improved prediction accuracy.

The current PKB algorithm can also be improved for better computational efficiency. The 3-fold cross-validation step is the most time and space-consuming part of the model, since it involves running three Boosting processes at the same time. It is possible to adopt the notion of out-of-bag (OOB) samples from RandomForest and GBR to calculate testing loss in just one Boosting process. In each iteration, instead of using all the samples, we draw a bootstrap sample to train the increment function. The samples not selected in the training set are treated as OOB samples, on which testing loss can be computed. It has been reported that OOB often underestimates the optimal number of iterations,[21] but brings the advantage of efficient model training, especially when the dataset is large.

A Python software implementing the PKB algorithms is available from Github repository: https://github.com/zengliX/PKB.

References

1. Li Zeng, Zhaolong Yu, and Hongyu Zhao. A pathway-based kernel boosting method for sample classification using genomic data. *Genes*, 10(9):670, 2019.

2. Hongzhe Li and Yihui Luan. Boosting proportional hazards models using smoothing splines, with applications to high-dimensional microarray data. *Bioinformatics*, 21(10):2403–2409, 2005.

3. Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

4. Rie Johnson and Tong Zhang. Learning nonlinear functions using regularized greedy forest. *IEEE transactions on pattern analysis and machine intelligence*, 36(5):942–954, 2014.

5. Ralf Bender, Thomas Augustin, and Maria Blettner. Generating survival times to simulate cox proportional hazards models. *Statistics in medicine*, 24(11):1713–1723, 2005.

6. Frank E Harrell Jr, Robert M Califf, David B Pryor, Kerry L Lee, Robert A Rosati, et al. Evaluating the yield of medical tests. *Jama*, 247(18):2543–2546, 1982.

7. Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

8. Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

9. Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

10. Alex J Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222, 2004.

11. Noah Simon, Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for coxs proportional hazards model via coordinate descent. *Journal of statistical software*, 39(5):1, 2011.

12. Hemant Ishwaran, Udaya B Kogalur, Eugene H Blackstone, and Michael S Lauer. Random survival forests. *The annals of applied statistics*, pages 841–860, 2008.

13. Harald Binder. Coxboost: Cox models by likelihood based boosting for a single survival endpoint or competing risks. *R package version*, 1, 2013.

14. Jordi Barretina, Giordano Caponigro, Nicolas Stransky, Kavitha Venkatesan, Adam A Margolin, Sungjoon Kim, Christopher J Wilson, Joseph Lehár, Gregory V Kryukov, Dmitriy Sonkin, et al. The cancer cell line encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature*, 483(7391):603, 2012.

15. Minoru Kanehisa and Susumu Goto. Kegg: kyoto encyclopedia of genes and genomes. *Nucleic acids research*, 28(1):27–30, 2000.

16. Darryl Nishimura. Biocarta. *Biotech Software & Internet Report: The Computer Software Journal for Scient*, 2(3):117–120, 2001.

17. Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25, 2000.

18. Gene Ontology Consortium. Expansion of the gene ontology knowledgebase and resources. *Nucleic acids research*, 45(D1):D331–D338, 2016.

19. Da Wei Huang, Brad T Sherman, and Richard A Lempicki. Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. *Nucleic acids research*, 37(1):1–13, 2008.

20. David Warde-Farley, Sylva L Donaldson, Ovi Comes, Khalid Zuberi, Rashad Badrawi, Pauline Chao, Max Franz, Chris Grouios, Farzana Kazi, Christian Tannus Lopes, et al. The genemania prediction server: biological network integration for gene prioritization and predicting gene function. *Nucleic acids research*, 38(suppl_2):W214–W220, 2010.

21. Greg Ridgeway et al. gbm: Generalized boosted regression models. *R package version*, 1(3):55, 2006.

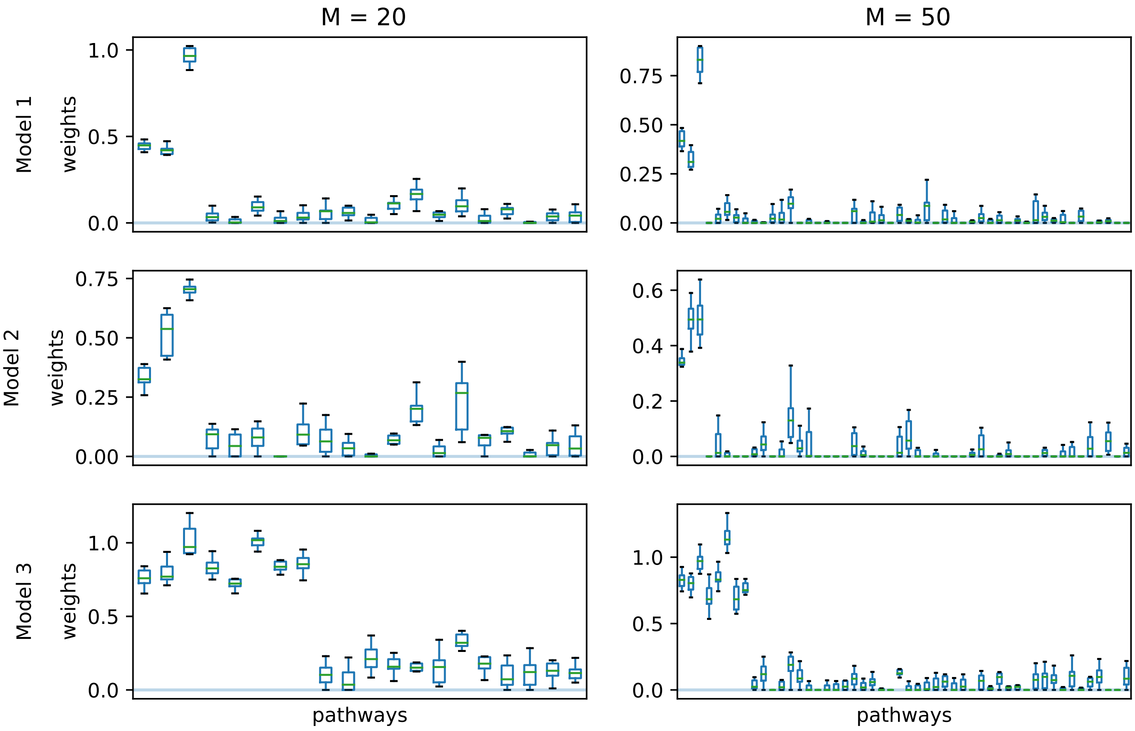

Figure 1: Boxplots for pathway weights in the regression simulations. Each box represents the weights distribution for one pathway over ten PKB runs.

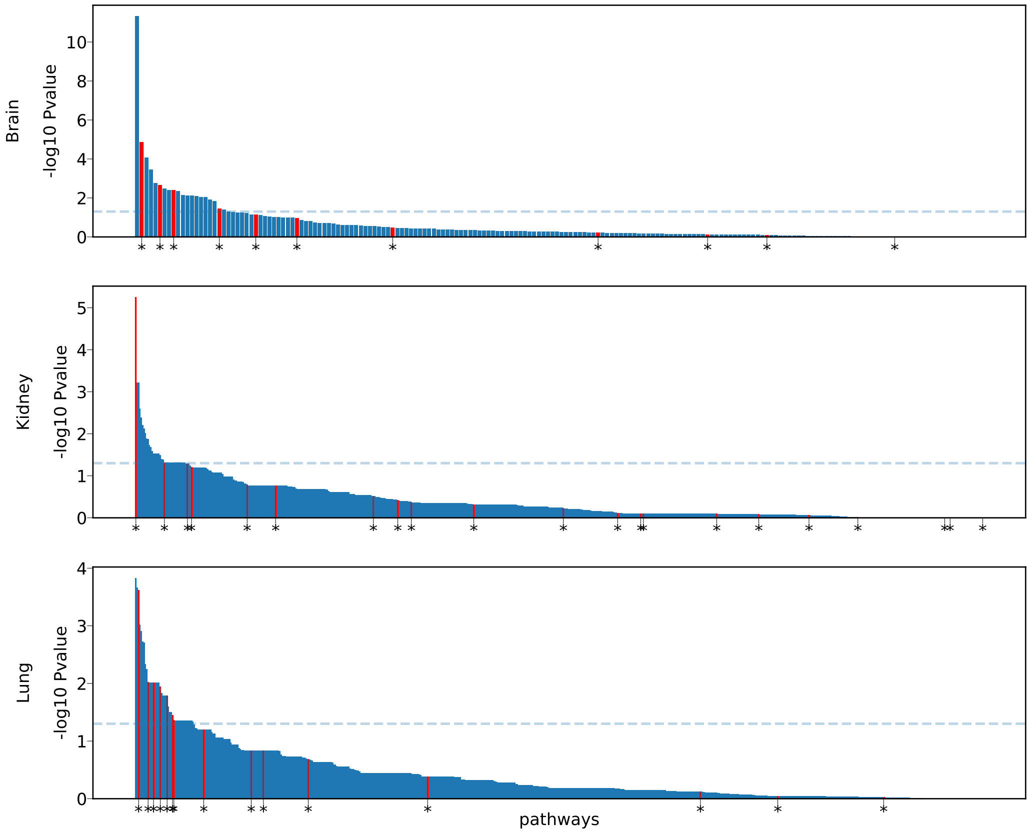

Figure 2: Enrichment analysis on brain, kidney, and lung cancer datasets. X-axis represents pathways sorted by their p-values in the enrichment analysis. The blue dashed line corresponds to p-value 0.05. The pathways marked with red bars and stars are pathways with significant weights in PKB.

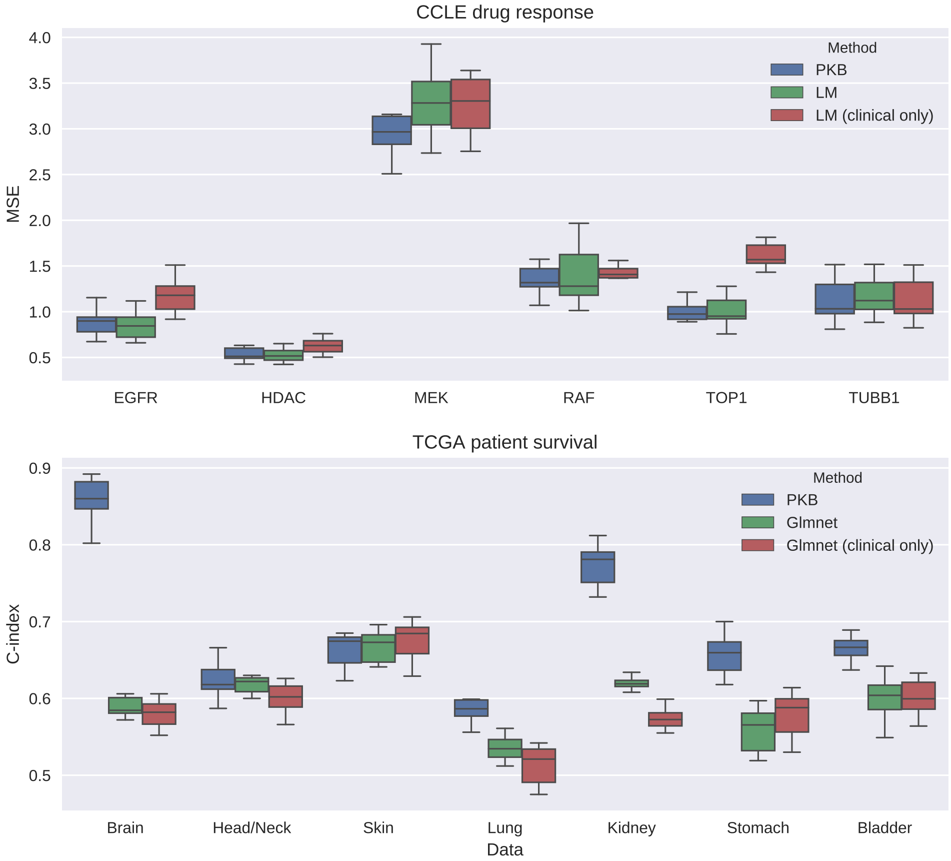

Figure 3: Prediction performances from models with and without genomic features. The figure presents the prediction accuracy from three types of models on each dataset. The models include: PKB, linear model, and linear model with only clinical features as predictors. LM in the upper panel represents linear regression model, which reports the best results from LASSO, Ridge Regression, and ElasticNet. The methods with label "clinical only" are trained without genomic features. The PKB boxes represent the best results from PKB-L_1 and PKB-L_2.

Method	Model 1		Model 2		Model 3	
	M = 20	M = 50	M = 20	M = 50	M = 20	M = 50
PKB-L_1	**17.11**	**22.82**	**32.92**	**33.94**	**7.22**	**8.37**
PKB-L_2	**16.84**	**22.41**	**31.25**	**33.65**	**5.4**	**5.77**
LASSO	34.85	40.58	49.74	50.94	21.15	22.88
Ridge	50.71	53.1	57.32	60.52	23.25	25.99
ElasticNet	34.87	42.21	49.27	50.93	21.25	23.51
RandomForest	46.88	50.94	55.34	57.34	21.09	22.67
GBR	48.54	51.75	50.9	52.82	21.46	23.7
SVR	50.02	53.58	56.12	59.58	19.06	24.31

Table 1: Cross-validated MSE of all methods on simulated regression datasets. M represents the number of simulated pathways. For each dataset, MSE values from the top two methods are in boldface.

Method	Model 1		Model 2		Model 3	
	M = 20	M = 50	M = 20	M = 50	M = 20	M = 50
PKB-L_1	**0.9**	**0.86**	**0.77**	**0.77**	**0.88**	**0.87**
PKB-L_2	**0.9**	**0.88**	**0.72**	**0.76**	**0.9**	**0.89**
Glmnet	0.78	0.79	0.69	0.71	0.65	0.66
RandomSurvivalForest	0.67	0.67	0.65	0.67	0.63	0.64
CoxBoost	0.78	0.78	0.7	0.7	0.66	0.66

Table 2: Cross-validated C-indices of all methods on simulated survival datasets. M represents the number of simulated pathways. For each dataset, C-indices from the top two methods are highlighted in boldface.

Method	EGFR	HDAC	MEK	RAF	TOP1	TUBB1
PKB-L_1	0.88	**0.54**	**2.92**	**1.37**	**0.98**	**1.12**
PKB-L_2	0.88	**0.54**	**2.9**	**1.35**	1.01	**1.12**
LASSO	**0.87**	0.62	3.63	1.47	1.14	1.18
Ridge	**0.84**	**0.52**	3.28	1.39	1.01	1.19
ElasticNet	0.89	0.59	3.37	1.44	1.14	1.17
RandomForest	0.88	0.56	3.17	**1.37**	**0.99**	**1.12**
GBR	0.91	**0.54**	3.3	1.43	1.0	1.14
SVR	0.88	0.55	3.17	1.38	**0.99**	**1.11**

Table 3: Cross-validated MSEs from all methods on CCLE drug response data. For each dataset, the two methods with the smallest MSEs are marked in boldface.

Method	Brain	Head/Neck	Skin	Lung	Kidney	Stomach	Bladder
PKB-L_1	**0.86**	**0.62**	0.66	**0.59**	**0.78**	**0.66**	**0.67**
PKB-L_2	**0.85**	0.61	0.64	**0.57**	**0.77**	**0.65**	**0.67**
Glmnet	0.59	**0.62**	**0.67**	0.54	0.62	0.56	0.6
RandomSurvivalForest	0.58	0.54	0.63	0.54	0.58	0.54	0.52
CoxBoost	0.59	**0.62**	**0.67**	0.53	0.61	0.53	0.6

Table 4: Cross-validated C-index from all methods on CCLE drug response data. For each dataset, the two methods with the highest C-index are marked in boldface.