BAYESIAN VARIABLE SELECTION FOR SURVIVAL DATA USING INVERSE MOMENT PRIORS

By Amir Nikooienejad\textsuperscript{\textdagger}, Wenyi Wang\textsuperscript{\textsection}, and Valen E. Johnson\textsuperscript{\textcopyright,\textquotesingle}

\textit{Eli Lilly and Company\textsuperscript{\textdagger}, MD Anderson Cancer Center\textsuperscript{\textsection} and Texas A\&M University\textsuperscript{\textcopyright,\textquotesingle}}

Efficient variable selection in high dimensional cancer genomic studies is critical for discovering genes associated with specific cancer types and for predicting response to treatment. Censored survival data is prevalent in such studies. In this article we introduce a Bayesian variable selection procedure that uses a mixture prior composed of a point mass at zero and an inverse moment prior in conjunction with the partial likelihood defined by the Cox proportional hazard model. The procedure is implemented in the R package BVSNLP, which supports parallel computing and uses a stochastic search method to explore the model space. Bayesian model averaging is used for prediction. The proposed algorithm provides better performance than other variable selection procedures in simulation studies, and appears to provide more consistent variable selection when applied to actual genomic datasets.

1. Introduction. Recent developments in sequencing technology have made it easier to collect massive genomic datasets that can be used to study cancer and other diseases. Given such data, there is great interest in linking genomic data to patient outcomes, and in many cases such outcomes are censored survival times.

Survival times for patients generally represent either the time to death or disease progression, the time to study termination, or the time until the subject is lost to follow up. In the latter cases, the subject’s survival time is censored. The relation between survival times and covariates is modeled through the hazard function, which is the limiting rate of death in the interval \((t, t + \Delta t)\) as \(\Delta t\) becomes small, given patient covariates. More precisely, the hazard function \(h\) for patient \(i\) may be defined as

\[
h(t \mid x_i) = \lim_{\Delta t \to 0} \frac{1}{\Delta t} P(t \leq T \leq t + \Delta t \mid T \geq t, x_i),
\]

Supported by NIH grant R01CA158113 prior to joining Eli Lilly and Company.
\footnote{Supported by R01CA174206, R01CA183793, R01CA158113 and P30CA016672.}
\footnote{Supported by NIH grant R01CA158113.}

Keywords and phrases: Bayesian Variable Selection, Nonlocal Prior, High Dimensional Data, Survival Data Analysis, Cox Proportional Hazard Model, Cancer Genomics
where $\mathbf{x}_i$ is a $p$ vector of covariates thought to influence survival. We denote by $\mathbf{X}$ the $n \times p$ design matrix obtained by stacking $n$ patient covariate vectors. Proportional hazard models take the form

$$(1.2) \quad h(t \mid \mathbf{x}_i) = h_0(t)\Phi(\mathbf{x}_i),$$

with an identifiability constraint of $\Phi(0) = 1$. In this formula, $h_0(t)$ denotes the baseline hazard function. The Cox proportional hazards model (Cox, 1972) is defined by taking $\Phi(\mathbf{x}_i) = \exp\{\mathbf{x}_i^T \beta\}$, leading to

$$(1.3) \quad h(t \mid \mathbf{x}_i) = h_0(t)e^{\mathbf{x}_i^T \beta}.$$ 

Here, $\beta$ is a $p \times 1$ vector of coefficients.

An important feature of the proportional hazards model is that it yields a partial likelihood function that is independent of the baseline hazard function, $h_0$. For complete survival analyses, however, the baseline hazard function is necessary for predicting survival times and can be estimated non-parametrically. Further details regarding the Cox proportional hazard model may be found in Cox and Oakes (1984), Kalbfleisch and Prentice (1980) or Cox (1972).

Gene expression datasets usually contain measurements on thousands of genes collected for only hundreds of subjects. Biologically, it seems plausible that only a relatively small number of these genes contribute significantly to survival. This implies that most of the elements in the vector $\beta$ are small or close to zero. The challenge is to find covariates with non-zero coefficients or, equivalently, those genes that contribute the most in determining the survival outcome.

Many common penalized likelihood methods originally introduced for linear regression have been extended to survival data. These methods include LASSO (Tibshirani et al., 1997), in which an $L_1$ penalty is imposed on regression coefficients. Zhang and Lu (2007) utilized adaptive LASSO methodology for time to event data, while Antoniadis, Fryzlewicz and Letué (2010) adopted the Dantzig selector for survival outcomes. The extension of non-convex penalized likelihood approaches, in particular SCAD, to the Cox proportional hazard model is discussed in Fan and Li (2002). The Iterative Sure Independence Screening (ISIS) approach introduced by Fan and Lv (2008) is also extended for ultrahigh dimensional survival data in Fan et al. (2010), where it is used on Cox proportional hazard models and the SCAD penalty is employed for variable selection.

Some Bayesian approaches have also been introduced. Faraggi and Simon (1998) proposed a method based on approximating the posterior distribution
of the parameters in the proportional hazard model by defining a Gaussian prior on regression coefficients. A loss function was then imposed to select a parsimonious model. A semi-parametric Bayesian approach was utilized by Ibrahim, Chen and MacEachern (1999), who employed a discrete gamma process for the baseline hazard function and a multivariate Gaussian prior for the coefficient vector. Sha, Tadesse and Vannucci (2006) considered Accelerated Failure Time (AFT) models along with data augmentation to impute failure times. A mixture prior proposed by George and McCulloch (1997) was used to impose sparsity. In more recent work, Held, Gravestock and Sabanés Bové (2016) proposed the use of a $g$-prior model for the coefficient vector and employed test-based Bayes factors (Johnson, 2005) to the Cox proportional hazard models. However, this method is intended for use only when the number of covariates is less than the number of observations; that is, when $p < n$.

To our knowledge, all previous Bayesian procedures for variable selection in survival data have used local priors on model coefficients. In Bayesian hypothesis tests, local priors put a positive probability on the null value of the parameter, in this case zero, whereas non-local priors put zero probability on the null value. Johnson and Rossell (2010) can be consulted for more discussion on properties of local and non-local priors in the context of Bayesian testing. In this article, we propose a Bayesian method based on a mixture prior comprised of a point mass at zero and a non-local prior on the regression coefficients. To handle the computational burden of implementing the resulting procedure we employ a stochastic search method, S5 (Shin, Bhattacharya and Johnson, 2018), which we implement in an R package BVSNLP. We also discuss a general procedure for setting the tuning parameter of the nonlocal prior.

This article is structured as follows. In Section 2 we introduce notation and discuss the modeling of the problem in a Bayesian framework. Section 3 discusses the proposed method, with details of parameter selection, model search, and assessment of the accuracy of the proposed variable selection procedure. Sections 4 and 5 provide simulation and real data analyses with various predictive performance measures to demonstrate how the proposed method compares to several other competing methods. Section 6 concludes with discussion.

2. Problem Modeling.

2.1. Preliminaries. Let $T_i$ denote the survival time and $C_i$ denote the censoring time for individual $i$. Each element in the observed vector of survival times, $y$, is defined as $y_i = \min\{T_i, C_i\}$. The status for each indi-
individual is defined as $\delta_i = I(T_i \leq C_i)$. The status vector is represented by $\delta = (\delta_1, \delta_2, \ldots, \delta_n)^T$. We assume that the censoring mechanism is “at random”, meaning that $C_i$ and $T_i$ are conditionally independent given $x_i$, where $x_i \in \mathbb{R}^p$ are the covariates for individual $i$ and comprise the $i^{th}$ row of $X$. The observed data is of the form $\{(y_i, \delta_i, x_i); \ i = 1, 2, \ldots, n\}$.

Model $k$ is defined as $k = \{k_1, \ldots, k_j\}$ where $(1 \leq k_1 < \cdots < k_j \leq p)$ and it is assumed that $\beta_{k_1}, \ldots, \beta_{k_j} \neq 0$ and all other elements of $\beta$ are 0. The design matrix corresponding to model $k$ is denoted by $X_k$, and the regression vector by $\beta_k = (\beta_{k_1}, \beta_{k_2}, \ldots, \beta_{k_j})^T$.

Let $\mathcal{R}(t) = \{i : y_i \leq t\}$ represent the risk set at time $t$, the set of all individuals who are still present in the study at time $t$ and are neither dead nor censored. We assume throughout this article that the failure times are distinct. In other words, only one individual fails at a specific failure time. With this assumption and letting $\xi_{ki} = \exp\{X_k^T \beta_k\}$, the partial likelihood (Cox, 1972) for $\beta_k$ in model $k$ can be written as

$$
L_p(\beta_k) = \prod_{i=1}^{n} \left[ \frac{\xi_{ki} \sum_{j \in \mathcal{R}(y_i)} \xi_{kj}}{\sum_{j=i}^{n} \xi_{kj}} \right]^{\delta_i}.
$$

Our method uses this partial likelihood as the sampling distribution in our Bayesian model selection procedure. We acknowledge that there is some information loss in (2.1) with respect to $\beta_k$. For instance, Basu (Ghosh, 1988) argues that partial likelihoods cannot usually be interpreted as sampling distributions. On the other hand, Berger et al. (1999) encourage the use of partial likelihoods when the nuisance parameters are marginalized out.

Sorting the observed unique survival times in ascending order and consequently re-ordering the status vector $\delta$ as well as the design matrix $X$ with respect to the ordered $y$, the sampling distribution of $y$ for model $k$ can be written as

$$
\pi(y \mid \beta_k) = \prod_{i=1}^{n} \left[ \frac{e^{X_k^T \beta_k}}{\sum_{j=i}^{n} e^{X_k^T \beta_k}} \right]^{\delta_i}.
$$

A Bayesian hierarchical model can be defined in which $\pi(y \mid \beta_k)$ in (2.2) represents the sampling distribution, $\pi_k(\beta_k)$ is the prior of model coefficients $\beta_k$, and $P(k)$ is the prior for model $k$. Using Bayes rule, the posterior probability for model $j$ is written as
where $\mathcal{J}$ is the set of all possible models and the marginal probability of the data under model $k$ is defined by

$$
(2.4) 
\sum_{k \in \mathcal{J}} P(j \mid y) \pi(j) P(k|m_j(y)) = \int \pi(y \mid \beta_k) m_k(y) d\beta_k.
$$

The prior density for $\beta_k$ and the prior on the model space impact the overall performance of the selection procedure and the amount of sparsity imposed on candidate models. Note that the sampling distribution in (2.2) is continuous in $\beta_k$, and in Section 2.3 we define an inverse moment prior (Johnson and Rossell, 2010) on each of the coefficients in model $k$.

2.2. Prior on Model Space. Let $\gamma_k = \{\gamma_1, \cdots, \gamma_p\}$ denote a binary vector indicating which covariates are included in model $k$. Suppose the size of model $k$ is $k$. That is, there are $k$ non-zero indices in $\gamma_k$. The non-zero indices of $\gamma_k$ represent the indices of the non-zero elements in the coefficient vector, $\beta$, which a priori are modeled as independent Bernoulli random variables with success probability $P(\gamma_i = 1) = \theta$ for every $1 \leq i \leq p$. As discussed in Scott et al. (2010), no fixed value for $\theta$ adjusts for multiplicity. As a result, it is necessary to define a prior on $\theta$, say $\pi(\theta)$. The resulting marginal probability for model $k$ in a fully Bayesian approach may then be written as

$$
(2.5) 
p(k) \propto \int \theta^k (1 - \theta)^{p - k} \pi(\theta) d\theta.
$$

A common choice for $\pi(\theta)$ is the beta distribution, $\theta \sim \text{Beta}(a, b)$, where in the special case of $a = b = 1$, $\pi(\theta)$ is a uniform distribution. The marginal probability for model $k$ derived from (2.5) is then equal to

$$
(2.6) 
p(k) = \frac{B(a + k, b + p - k)}{B(a, b)},
$$

where $B(\cdot)$ is the Beta function. $A priori$, the model size, $k$, thus follows a Beta-binomial distribution. By choosing $b = p - a$, the mean and variance of the selected model size $k$ is

$$
(2.7) \quad \mathbb{E}(k) = a, \quad \text{Var}(k) = \frac{2p^2a(p - a)}{p^3} \approx 2a.
$$
The approximation in the variance formula follow from a large $p$ and a fairly small $a$ under the sparsity assumption on the true model size. To incorporate the belief that the optimal predictive models are sparse, we recommend setting $a = 1$ and $b = p - a$. The resulting prior assigns comparatively small prior probabilities to models that contain many covariates.

2.3. Product Inverse MOMent (piMOM) Prior. We impose non-local prior densities on the non-zero coefficients, $\beta_k$. Specifically, we assume the prior densities on the non-zero coefficients in model $k$ take the form of a product of independent iMOM priors, or piMOM densities (Johnson and Rossell, 2012), expressible as

$$
(2.8) \quad \pi(\beta_k|\tau, r) = \frac{\tau^{rk/2}}{\Gamma(r/2)^k} \prod_{i=1}^{k} |\beta_i|^{-(r+1)} \exp \left( -\frac{\tau}{\beta_i^2} \right), \quad r, \tau > 0.
$$

The hyperparameter $\tau$ represents a scale parameter that determines the dispersion of the prior around 0, while $r$ determines the tail behavior of the density. These priors have two symmetric modes with Cauchy-like tails when $r = 1$, and assign negligible probability to a region around zero. In comparison to local priors, this characteristic of non-local priors potentially leads to smaller false positive rates in selection procedures by discouraging the selection of variables with small coefficients. On the other hand, piMOM priors possess Cauchy-like tails, which introduce comparatively small penalties on large coefficients. Unlike many penalized likelihood methods, large values of regression coefficients are thus not heavily penalized by these priors. As a result, they do not necessarily impose significant penalties on non-sparse models provided that the estimated coefficients in those models are not small. For these reasons, piMOM priors work well as a default choice of priors on non-negligible coefficients in variable selection problems. An example of an iMOM prior is depicted in Figure 1 for $r = 1$ and $\tau = 0.5$.

Another non-local prior that might be considered as a potential candidate for the prior densities on the non-zero coefficients in model $k$ is the product of independent MOM priors, or the pMOM densities (Johnson and Rossell, 2012). A detailed discussion on the pMOM priors and their properties is provided in Section 3 of the Supplementary Material (Nikooienejad et al., 2020). A simulation analysis to compare the performance of piMOM and pMOM in the selection process is also provided. For the reasons discussed there, piMOM-based procedures are more effective for variable selection in $p \gg n$ settings and therefore is our choice for the analyses of the simulation and real data in this article.

3.1. Selection of Hyperparameters. We use the procedure described in Nikooienejad, Wang and Johnson (2016) to select hyperparameter values for the piMOM prior. In that method, the null distribution of the maximum likelihood estimator for $\beta_k$ (i.e., all components of $\beta_k$ are 0), obtained from randomly selected design matrices $X_k$, is compared to the prior density on $\beta_k$ for various values of $(r, \tau)$. Fixing $r = 1$ to achieve Cauchy-like tails, a value of $\tau$ is chosen so that the overlap between the two densities is less than a specified threshold, $1/\sqrt{p}$, and is denoted by $\tau_1$. It can be shown that the maximum of the iMOM prior occurs at $\pm \sqrt{\tau}$. We also allow users to input a prior parameter $\alpha$ that controls where the modes in the prior occur. This can be useful in constraining the prior density when covariates are highly correlated (resulting in an over-dispersed prior when the sampling distribution of the null MLE under the null model becomes overly broad). We then set the value of $\tau$ according to

\begin{equation}
(3.1) \quad \tau = \min(\tau_1, \alpha^2).
\end{equation}

To implement the procedure for computing $\tau_1$ for survival models, we generate response vectors under the null model using the procedure described by Bender, Augustin and Blettner (2005). Survival times are sampled from a standard exponential model.
Let $t^s$ and $c^s$ be the vector of sampled survival times and censoring times, respectively. The sampled survival time and status for each observation is then computed as

\begin{equation}
(3.2) \quad y^s_i = \min \{ t^s_i, c^s_i \} \quad \text{and} \quad \delta^s_i = I(t^s_i \leq c^s_i),
\end{equation}

which comprise $y^s$ and $\delta^s$ under the null model. Using the pair $(y^s, \delta^s)$, the MLE from a Cox model is computed. It should be noted that the asymptotic distribution of the MLE for the Cox model under the null hypothesis is $\hat{\beta} \sim \mathcal{N}(0, I(\hat{\beta}))$, where $I(\beta)$ is the information matrix of the partial likelihood function. Thus, it is appropriate to approximate the pooled estimated coefficients in that algorithm with a normal density function. When the sample size gets large, the variance of the MLE decreases and causes the overlap to become small and consequently small values of $\tau$ are selected.

In general, we find that $r = 1$ and $\tau = 0.25$ are good default values if one chooses not to run the hyperparameter selection algorithm. When $r = 1$, the peaks of the iMOM prior occur at $-\sqrt{\pi}$ and $\sqrt{\pi}$. By equating $\sqrt{\pi}$ to the absolute value of the expected effect size for a given application, insight can be gained on what value of $\tau$ is appropriate. Further details regarding this algorithm can be found in Nikooienejad, Wang and Johnson (2016).

3.2. Computing Posterior Probability of Models. Computing the posterior probability for each model requires the marginal probability of observed survival times under each model, as shown in (2.3) and (2.4). The marginal probability is approximated using the Laplace approximation, where the regression coefficients in $\beta_k$ are integrated out. This leads to

\begin{equation}
(3.3) \quad m_k(y_n) = \pi(y_n \mid \hat{\beta}_k) \pi(\hat{\beta}_k)(2\pi)^{k/2} |G_{\hat{\beta}_k}|^{-1/2}.
\end{equation}

Here, $\hat{\beta}_k$ is the maximum a posteriori (MAP) estimate of $\beta_k$, $G_{\hat{\beta}_k}$ is the Hessian of the negative of the log posterior function,

\begin{equation}
(3.4) \quad g(\beta_k) = -\log(\pi(y \mid \beta_k)) - \log(\pi(\beta_k)),
\end{equation}

computed at $\hat{\beta}_k$ and $k$ is the size of model $k$. Finding the MAP of $\beta_k$ is equivalent to finding the minimum of $g(\beta_k)$.

The details of computing the gradient and Hessian matrix of $g(\beta_k)$ are discussed in Section 1 of the Supplementary Material (Nikooienejad et al., 2020). The gradient and Hessian matrix, described by equations (3) to (7) there, are used to find the MAP and to compute the Laplace approximation.
of the marginal probability of \( y \). We use the limited memory version of the Broyden-Fletcher-Goldfarb-Shanno optimization algorithm (L-BFGS) (Liu and Nocedal, 1989) to find the MAP. The initial value for the algorithm is \( \hat{\beta} \), the MLE for the Cox proportional hazard model.

Having all the components of formula (2.3), it is possible to define a MCMC framework to sample from the posterior distribution on the model space. A birth-death scheme, similar to that used in Nikoiejejad, Wang and Johnson (2016), could be used for this purpose. However, for computational reasons we use another stochastic algorithm to search the model space; this algorithm is described in the next section.

The highest posterior probability model (HPPM) is defined as the model having the highest posterior probability among all visited models. In practice, many models may be assigned probabilities that are close to the probability achieved by the HPPM. For this reason and for predictive purposes, it is useful to obtain the Median Probability Model (MPM) (Barbieri and Berger, 2004), which is the model containing covariates that have posterior inclusion probabilities of at least 0.5. According to Barbieri and Berger (2004), the posterior inclusion probability for covariate \( i \) is defined as

\[
    p_i = \sum_{k: \gamma_{ki} = 1} P(M_k | y).
\]

That is, the sum of posterior probabilities of all models that have covariate \( i \) as one of their variables. In this expression, \( \gamma_{ki} \) is a binary value determining the inclusion of the \( i^{th} \) covariate in model \( k \).

3.2.1. Stochastic Search Algorithm. To increase the efficiency of exploring the model space, we use the S5 algorithm. S5 was proposed by Shin, Bhattacharya and Johnson (2018) for variable selection in linear regression problems, and we adapt it here for survival models. It is a stochastic search method that screens covariates at each step. The algorithm is scalable and its computational complexity is only linearly dependent on \( p \) (Shin, Bhattacharya and Johnson, 2018).

Screening is the essential part of the S5 algorithm. In linear regression, screening is based on the correlation between excluded covariates and the residuals of the regression using the current model (Fan and Lv, 2008). The concept of screening covariates for survival response data is proposed in Fan et al. (2010), and is defined based on the marginal utility for each covariate.

To illustrate the screening technique, suppose that the current model is \( k \). Let \( k^c \) denote the complement of set \( k \) containing columns of the design matrix that are not in the current model, \( k \). The conditional utility of covariate
\( m \in k^c \) represents the amount of information covariate \( m \) contributes to the survival outcome, given model \( k \), and is defined as

\[
u_{m|k} = \max_{\beta_m} \delta^T \left[ \left( \beta_m, X(m) + X_k \beta_k \right) \right]
- \log \left\{ \sum_{j=i}^{n} \exp(\beta_m x_{jm} + X_k \beta_k) \right\}.
\]

(3.6)

By comparing \( u_{m|k} \) to the Cox log-likelihood equation (Formula (1) in the Supplementary Material (Nikooienejad et al., 2020)), it follows heuristically that the conditional utility is the maximum likelihood for covariate \( m \) after accounting for the information provided by model \( k \). Finding \( u_{m|k} \) is a univariate optimization procedure that can be computed rapidly.

With this background, the S5 algorithm for survival data works as follows. At each step, the \( d \) covariates with highest conditional utility are candidates to be added to the current model \( k \) and comprise the addition set, \( \Gamma^+ \). The deletion set, \( \Gamma^- \) contains the current model, except that one variable is removed. From the current model, \( k \), we consider moves to each of its neighbors in \( \Gamma^+ \) and \( \Gamma^- \) with a probability proportional to the marginal probabilities of these neighboring models.

To avoid local maxima, the model probabilities used in S5 are raised to the power of \( 1/t_l \), where \( t_l \) is the \( l^{th} \) temperature in an annealing schedule in which “temperatures” decrease. To increase the number of visited models, a specified number of iterations are performed at each temperature. At the end of the procedure, the model with the highest posterior probability of visited models is identified as the HPPM.

In our version of the S5 algorithm, we used 10 equally spaced temperatures varying from 3 to 1 and 30 iterations within each temperature. Section 4 of the Supplementary Material (Nikooienejad et al., 2020) provides some discussion on how these values are chosen for this application. To increase the number of visited models, we parallelized the S5 procedure so that it could be distributed to multiple CPUs. Each CPU executes the S5 algorithm independently with a different starting model. All visited models are pooled together at the end and the HPPM and MPM are determined. Using posterior probabilities of the visited models, the posterior inclusion probability for each covariate can be computed using (3.5). In our simulations, we used 120 CPUs to explore the model space for design matrices with \( O(10^4) \) covariates.

3.3. Predictive Accuracy Assessment. In addition to looking at the selected genes and their pathways to determine their biological relevance in
analyzing the real data sets, we used the time dependent AUC, obtained from time dependent ROC curves as introduced by Heagerty, Lumley and Pepe (2000) for survival times to summarize and compare the predictive performance of the various algorithms. This measure has a relatively straightforward interpretation, and unlike other summary measures such as the c-index (Harrell Jr et al., 1982), can be computed without requiring specific conditions or additional assumptions to hold (Blanche, Kattan and Gerds, 2018). However, predictive performance measures including the c-index, Integrated Brier Score (IBS) (Gerds and Schumacher, 2006), and prediction error curves are investigated and reported in Sections 4 and 5 for both simulation and real data sets.

There are different methods to estimate time dependent sensitivity and specificity. In our algorithm, we adapted a method proposed by Uno et al. (2007), henceforth called Uno’s method. In that method, after splitting data into training and test sets, sensitivity is estimated by

$$\hat{SE}_k(t, c) = \frac{\sum_{i=1}^{n} \delta_i I(X_k^T \hat{\beta}_k > c, T_i \leq t)}{\sum_{i=1}^{n} \delta_i I(T_i \leq t) / \hat{G}(T_i)}$$

and specificity is estimated by

$$\hat{SP}_k(t, c) = \frac{\sum_{i=1}^{n} I(X_k^T \hat{\beta}_k \leq c, T_i > t)}{\sum_{i=1}^{n} I(T_i > t)}.$$

These values are estimated for the test set. Therefore, in the equations above, $n$ is the number of observations in the test set, $\delta_i$ is the status of observation $i$ and $T_i$ is the observed time for that observation in the test set. The variable $c$ is the discrimination threshold that is varied to obtain the ROC curve. The function $\hat{G}$ is the Kaplan-Meier estimate of the survival function obtained from the training set. For each observation $i$ in the test set with observed time $T_i$, $\hat{G}(T_i)$ is computed by a basic interpolation procedure. That is,

$$\hat{G}(T_i) = \hat{G}(T_{tr}^*), \text{ where } T_{tr}^* = \arg\min_{T \in T_{tr}} |T - T_i|.$$  

Here, $T_{tr}$ is the set of all observed survival times in the training set. In (3.7) and (3.8), $\hat{\beta}$ represents the estimated coefficient under a specific model.

3.3.1. **Bayesian Model Averaging (BMA).** BMA can be used to improve the predictive accuracy by accounting for the uncertainty in selected models.
From (3.7) and (3.8) the final sensitivity and specificity using BMA may be defined as

\[
\hat{SE}_{BMA}(t, c) = \sum_{j=1}^{N} \hat{SE}_{k_j}(t, c) P(M_{k_j} \mid y_n),
\]

and

\[
\hat{SP}_{BMA}(t, c) = \sum_{j=1}^{N} \hat{SP}_{k_j}(t, c) P(M_{k_j} \mid y_n),
\]

where, \( P(M_{k_j} \mid y_n) \) is the posterior probability of model \( M_{k_j} \). The value of \( N \) depends on what type of BMA is used. We use Occam’s window, which means only models that have posterior probability of at least \( w \times p(M_{HPPM} \mid y_n) \) are used in model averaging. We set \( w = 0.01 \) for our applications.

In the proposed method, individual survival curves are estimated using the highest posterior probability model. Section 2 of the Supplementary Material (Nikooienejad et al., 2020) provides the details of this procedure. Similar approaches were also adopted by Held, Gravestock and Sabanés Bové (2016) in estimating the survival curve for each individual in a study.

4. Simulation Results. To investigate the performance of the proposed model selection procedure, we applied our method to simulated datasets. We followed the guidance of Morris, White and Crowther (2019) as a basis for our simulation protocol. In particular, the simulation design was based on the ADEMP structure (Aims, Data generating mechanism, Estimands, Methods, and Performance measures) discussed in that article. We refer to each of those elements as we explain different parts of the simulation design below.

Regarding ‘Methods’, we compared the performance of our algorithm to ISIS-SCAD (Fan et al., 2010) and GLMNET (Friedman, Hastie and Tibshirani, 2010), two of the most highly used algorithms for high dimensional variable selection for survival data. We used the published R packages of those two methods to run the simulations. We also performed a comparison with a case when pMOM priors are used as the prior for non-zero coefficients instead of piMOM priors.

The ‘Aim’ of the simulation study was to compare the performance of our method with the other two methods with respect to the correlation structure between covariates in the design matrix. More specifically, we reported three
different simulation settings that consider different combinations of correlation structure, true model size, and the magnitude of true coefficients. This was the basis of our ‘Data generating mechanism’. The correlation structure used in those settings are similar to the simulations reported in Fan et al. (2010).

For Case 1, $X_1, \ldots, X_p$ were multivariate Gaussian random variables with mean 0 and marginal variance of 1. The correlation structure was $\text{corr}(X_i, X_5) = 0$ for all $i \neq 4, 5$, $\text{corr}(X_5, X_4) = 1/\sqrt{2}$, and $\text{corr}(X_i, X_j) = 0.5$ for $i, j \in \{1, \ldots, p\} \setminus \{4, 5\}$. The size of the true model was 5 with non-zero regression coefficients $\beta_1 = -1.5389, \beta_2 = 0.6839, \beta_3 = -0.8498, \beta_4 = -1.2716, \beta_5 = -1.1045$ and $\beta_i = 0$ for $i > 5$. The non-zero coefficients were chosen randomly, and were generated as $Z \cdot U$ with $Z \sim \text{Unif}(0, 2)$ and $U = 1$ or $U = -1$ with probability 0.5. The number of observations and covariates were $n = 400$ and $p = 1000$. The censoring rate for this case was approximately 27.6%. The survival and censoring times are both sampled from an exponential distribution. The rate parameter for the distribution of censoring times was set to 0.1.

For Case 2, $X_1, \ldots, X_p$ were multivariate Gaussian random variables with mean 0 and marginal variance of 1. The correlation structure between variables was $\text{corr}(X_i, X_j) = 0.5; i \neq j$. The size of the true model was 6 with non-zero regression coefficients $\beta_1 = 1.1201, \beta_2 = 0.8322, \beta_3 = -1.9620, \beta_4 = -1.7639, \beta_5 = 1.6782, \beta_6 = 1.8995$, and $\beta_i = 0$ for $i > 6$. As in Case 1, the non-zero coefficients were chosen randomly, and were generated as $Z \cdot U$ with $Z \sim \text{Unif}(0, 2)$ and $U = 1$ or $U = -1$ with probability 0.5. The number of observations and covariates were $n = 400$ and $p = 1000$. In this case, the survival times were sampled from a Weibull distribution with rate parameter $\lambda = 0.1$ and shape parameter $\kappa = 15$. The censoring times were sampled uniformly from $[0, 8]$, and the resulting censoring rate for this case was approximately 14.8%.

For Case 3, the design matrix and correlation structure between variables was the same as Case 2, where $\text{corr}(X_i, X_j) = 0.5; i \neq j$. The size of the true model was 20 with non-zero regression coefficients $(\beta_1, \ldots, \beta_{20})$ equal to $(-1.6802, -1.2483, 2.9430, -2.6458, -2.5173, -1.5931, 0.8800, -0.9387, 1.6599, -2.9288, -1.2495, -2.6298, -2.3434, 1.9075, -1.1044, -0.7873, 2.6722, -0.6340)$, and $\beta_i = 0$ for $i > 20$. Like the other two cases, the non-zero coefficients were chosen randomly but with larger upper limit on the uniform distribution. They were generated as $Z \cdot U$ with $Z \sim \text{Unif}(0, 3)$ and $U = 1$ or $U = -1$ with probability 0.5. The number of observations and covariates were $n = 400$ and $p = 1000$. The censoring rate for this case was approximately 34.1%. The survival and censoring times were both sampled
from an exponential distribution. The rate parameter for the distribution of censoring times was set to 0.1.

Each simulation case was then repeated 50 times, \( n_{\text{iter}} = 50 \), and each time with different random seed numbers in order to generate different datasets.

The primary targets of our simulation study or the ‘Estimands’, according to Morris, White and Crowther (2019), were identifying the true model as well as estimating the vector of coefficients of the true model. Accordingly, we reported four different quantities as ‘Performance measures’ for those estimands. The first two quantities are the mean \( l_1 \) norm of the error in estimating the vector of coefficients and the mean squared error (MSE). The mean \( l_1 \) norm was computed as \( \frac{1}{n_{\text{iter}}} \sum_{i=1}^{p} |\hat{\beta}_i - \beta_i| \), and the MSE was computed as \( \frac{1}{n_{\text{iter}}} \sum_{i=1}^{p} (\hat{\beta}_i - \beta_i)^2 \). The third quantity is the mean model size of the selected models and was denoted by MMS. MTP and MFP denote mean false positive and mean true positive values for each algorithm. Formal definitions of MFP, MTP are provided in Section 3 of the Supplementary Material (Nikooienejad et al., 2020).

Table 1 compares the performance of our method, BVSNLP, the default settings of ISIS-SCAD and GLMNET algorithms. The \( \lambda \) parameter in GLMNET was picked by cross validation. Table 2 compares the Monte Carlo standard errors (Morris, White and Crowther, 2019) of the MSEs for all three methods.

In the S5 algorithm, 30 iterations were used within each temperature. The parameter \( d \) was chosen as \( 2\lceil \log(p) \rceil \). As described in Section 3.2.1, \( d \) represents the number of candidate covariates that were added to the current model to make the addition set, \( \Gamma^+ \). Each S5 algorithm was run in parallel on 120 CPUs for all three simulation cases. The beta-binomial prior was imposed on the model space with \( a = 1, b = p - a \). The hyperparameters of the piMOM prior were selected using the algorithm discussed in Section 3.1 with \( \alpha = 0.8 \) for all three simulation cases, imposed as the prior mode.

Finally, the average run-times of the BVSNLP algorithm for the three simulation cases were 29, 20.23 and 27.99 seconds, respectively.

As demonstrated in Table 1, our method performs better than the other two methods according to all selected metrics, regardless of the size of the true model. The difference between BVSNLP and ISIS-SCAD is best illustrated as the size of the true model increases. GLMNET has significantly higher mean false positive rates than the other two methods.

Figures 2, 3a, and 3b compare the average IBS over 50 iterations between the methods discussed above. IBS is computed using the R package \texttt{pec} (Mogensen, Ishwaran and Gerds, 2012) based on a five-fold cross validation.
Table 1
Comparison between BVSNLP, ISIS-SCAD and GLMNET for simulation Cases 1, 2, and 3 with $n = 400$ and $p = 1000$.

<table>
<thead>
<tr>
<th></th>
<th>BVSNLP</th>
<th>ISIS-SCAD</th>
<th>GLMNET</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Case 1:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSE</td>
<td>0.141</td>
<td>0.792</td>
<td>1.441</td>
</tr>
<tr>
<td>Mean $l_1$ norm</td>
<td>0.488</td>
<td>2.200</td>
<td>4.072</td>
</tr>
<tr>
<td>MMS</td>
<td>4.96</td>
<td>8.84</td>
<td>51.46</td>
</tr>
<tr>
<td>MTP</td>
<td>4.92</td>
<td>4.62</td>
<td>4.00</td>
</tr>
<tr>
<td>MFP</td>
<td>0.04</td>
<td>4.22</td>
<td>47.46</td>
</tr>
<tr>
<td><strong>Case 2:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSE</td>
<td>0.141</td>
<td>0.792</td>
<td>1.441</td>
</tr>
<tr>
<td>Mean $l_1$ norm</td>
<td>0.505</td>
<td>0.552</td>
<td>3.891</td>
</tr>
<tr>
<td>MMS</td>
<td>6</td>
<td>5.94</td>
<td>50.94</td>
</tr>
<tr>
<td>MTP</td>
<td>6</td>
<td>5.88</td>
<td>5.92</td>
</tr>
<tr>
<td>MFP</td>
<td>0</td>
<td>0.06</td>
<td>45.02</td>
</tr>
<tr>
<td><strong>Case 3:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSE</td>
<td>0.602</td>
<td>5.287</td>
<td>4.701</td>
</tr>
<tr>
<td>Mean $l_1$ norm</td>
<td>2.680</td>
<td>22.962</td>
<td>22.824</td>
</tr>
<tr>
<td>MMS</td>
<td>20.08</td>
<td>14.76</td>
<td>105.62</td>
</tr>
<tr>
<td>MTP</td>
<td>19.94</td>
<td>12.80</td>
<td>19.96</td>
</tr>
<tr>
<td>MFP</td>
<td>0.14</td>
<td>1.96</td>
<td>85.66</td>
</tr>
</tbody>
</table>

A benchmark model based on Kaplan-Meier estimate, which includes no covariates, is also added to the figures as a reference for the comparison. The average c-index measures for all the methods are also reported in Table 3. The c-index measures are computed based on the method discussed in van Houwelingen and Putter (2011), using the `dynpred` package in R. Because a new dataset was created at each iteration, it was not possible to get the average prediction errors, due to the fact that the times points where prediction errors change were different for different data sets.

As shown in the IBS plots, all three methods perform better than the reference. BVSNLP and ISIS-SCAD have a very similar performance. For Case 3, where the true model has 20 covariates, BVSNLP outperforms the

Table 2
Monte Carlo Standard Errors for the MSE of the coefficient vector for all three methods.

<table>
<thead>
<tr>
<th></th>
<th>Case 1</th>
<th>Case 2</th>
<th>Case 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>BVSNLP</td>
<td>0.064</td>
<td>0.008</td>
<td>0.065</td>
</tr>
<tr>
<td>ISIS-SCAD</td>
<td>0.098</td>
<td>0.015</td>
<td>0.142</td>
</tr>
<tr>
<td>GLMNET</td>
<td>0.007</td>
<td>0.024</td>
<td>0.055</td>
</tr>
</tbody>
</table>
Fig 2: Average IBS for all methods in simulation case 1.

other two methods, whereas in Case 2, GLMNET has the best performance. The c-index is similar for all methods, and seems to provide a smaller penalty for model size. This feature of the c-index is discussed further in Section 6.

Table 3

<table>
<thead>
<tr>
<th>Method</th>
<th>Case 1</th>
<th>Case 2</th>
<th>Case 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>BVSNLP</td>
<td>0.890</td>
<td>0.881</td>
<td>0.960</td>
</tr>
<tr>
<td>ISIS-SCAD</td>
<td>0.895</td>
<td>0.876</td>
<td>0.841</td>
</tr>
<tr>
<td>GLMNET</td>
<td>0.911</td>
<td>0.908</td>
<td>0.970</td>
</tr>
</tbody>
</table>

5. Application to Real Data. We applied our method to selected genes associated with patient survival times for two common cancer types using datasets from The Cancer Genome Atlas (TCGA) projects: kidney renal clear cell carcinoma (KIRC) (Cancer Genome Atlas Research Network, 2013) and kidney renal papillary cell carcinoma (KIRP) (Cancer Genome Atlas Research Network, 2016). We compared the performance of our algorithm to ISIS-SCAD (Fan et al., 2010), GLMNET (Friedman, Hastie and Tibshirani, 2010) and Stability Selection (Meinshausen and Bühlmann, 2010). Stability Selection was combined with a high dimensional selection algorithm such as GLMNET and selects the most stable features for a given level of Type I error. To run the Stability Selection method, we used the c060 R package (Sill et al., 2014) and the recommended values for function arguments.
We included patients’ ‘Age’, ‘Gender’ and a clinical stage variable, ‘Stage’, in the design matrix. On the advice of a clinician, the ‘Stage’ variable was developed by combining the histological stage, pathological stage and clinical stage into one variable that is a summary of how advanced each subject’s cancer was when the tissue sample was taken. ‘Stage’, like ‘Gender’, is a categorical variable but with 3 levels, where ‘Stage $i$’ represents the $i^{th}$ class of that variable; ‘Stage 3’ represents the most advanced stage. To remove stromal contaminations from the gene expression data, the DeMixT algorithm (Wang et al., 2017), was performed on the design matrix and the tumor-specific expression data were used in the analyses for all algorithms.

The predictive performance was measured by a time-dependent AUC, as discussed in Section 3.3, based on a five-fold cross-validation. The observations in each fold were randomly chosen under a constraint which balanced censoring rate between folds. The AUC values were computed for the test set using the model that was obtained by performing variable selection on the training set. The selected covariates for each cancer type were also compared. For our method, we report the covariates associated with the highest posterior probability model. The hyperparameter $\tau$ of the piMOM prior was selected using the algorithm in Section 3.1 with $\alpha = 0.1$ as the mode of the piMOM prior. This is our choice of $\alpha$ for real datasets. The results for each cancer type are discussed in separate sections below. Note that GLMNET has a random output when the hyperparameter is selected by cross-validation. As a result, based on the recommendation of the inventors of that algorithm, we ran it 100 times for each fold and took the average of results as the outcome for that fold.
5.1. Kidney Renal Clear Cell Carcinoma (KIRC). The KIRC dataset (Cancer Genome Atlas Research Network, 2013) contains 490 observations with 13,267 covariates, after removing covariates with missing expressions and observations with missing survival times. The censoring rate for this dataset is 66.94%. Table 4 shows the covariates selected by each method. As mentioned previously, GLMNET produces random outputs at each run and therefore for this table, only the output for one of the runs are indicated; other runs produced a similar number of selected covariates.

In addition to the categorical covariate ‘Stage’, BVSNLP selects ‘AR’ and ‘SUDS3’ in the HPPM as the most significant covariates in the design matrix. The posterior inclusion probabilities for ‘AR’ and ‘SUDS3’ are 0.80 and 0.08, respectively. The ‘Age’, ‘Gender’, and ‘Stage’ were fixed in all models and thus were selected with probability 1. The MAP estimates for the coefficients of ‘Age’, ‘Gender Male’, ‘Stage 2’, ‘Stage 3’, ‘AR’ and ‘SUDS3’ were $0.33, -0.11, 0.45, 1.61, -0.60$ and $0.36$, respectively. These coefficients indi-

<table>
<thead>
<tr>
<th>Method</th>
<th>Selected Genes and Covariates for KIRC across different variable selection algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td>BVSNLP</td>
<td>Stage 3, AR, SUDS3, HEBP1, MTERF2, SERPIN11, INAFM2</td>
</tr>
<tr>
<td>ISIS-SCAD</td>
<td>Stage 3, AR, ATP2C1, ADGRL3, SP6, ZNF815P</td>
</tr>
<tr>
<td>GLMNET</td>
<td>Stage, AR, SEC61A2, PCBP4, E2F5, RAB28, HACD1, INAFM2, TRAIL, MCM8, RPL17P50, ACACA</td>
</tr>
<tr>
<td>Stability Selection</td>
<td>Stage 3, AR, INAFM2</td>
</tr>
</tbody>
</table>

We treated categorical variables ‘Stage’ and ‘Gender’, as well as the continuous variable ‘Age’, as fixed covariates in our model. However, available ISIS-SCAD and Stability Selection software packages are not able to fix pre-selected covariates to be included in all models. For this reason, dummy variables associated to ‘Stage’ and ‘Gender’ were manually added to the design matrix and were subject to the selection procedure for those methods.
cate that patients with the most advanced stages of cancer had the poorest survival rates, and that a patient with a tumor sample characterized as advanced had a hazard rate that was $\exp(1.61) \approx 5$ times higher than a patient with tumor sample characterized as localized, when all other covariates were the same. These coefficients also show that the hazard rate in females is 1.12 times that in males, and age has an unfavorable impact on the hazard rate, as expected. Moreover, the negative sign for the ‘AR’ gene indicates it has a favorable impact on survival for KIRC. ‘AR’, the Androgen Receptor gene, functions as a steroid-hormone activated transcription factor. It has been well documented that ‘AR’ promotes the progression of renal cell carcinoma (RCC) through hypoxia-inducible factors HIF-2α and vascular endothelial growth factor regulation (Fenner, 2016). The favorable impact of the ‘AR’ gene was also studied by Hata et al. (2017) in bladder cancer. ‘SUDS3’ is a regulatory protein that is part of the SIN3A corepressor complex component that potentially has a role in tumor suppressor pathways through regulation of apoptosis. There was previous evidence of the down-regulation of the SIN3A gene in tumorigenesis of lung cancer (Suzuki et al., 2008).

It is noteworthy that the algorithm selected the same highest posterior probability model for different values of the hyperparameter $\tau$ in the range $[0.01, 0.9]$. This shows the robustness of the proposed variable selection algorithm to the choice of hyperparameter $\tau$ for a range of plausible values.

For this particular run of GLMNET, a much larger model was selected with 24 variables including two of the variables reported by BVSNLP. ISIS-SCAD selected 13 covariates, which included the 4 covariates that were selected by the Stability Selection method. ‘AR’ and the last level of ‘Stage’ are the common covariates among all methods.

The time dependent AUC plot for all four methods, obtained by performing a five-fold cross validation, is depicted in Figure 4. BVSNLP has slightly better predictive accuracy than GLMNET and Stability Selection. However, it achieves this accuracy with a much sparser model. We investigated the covariates that were selected by each of those algorithms in all five folds and found that BVSNLP, in addition to those fixed covariates, selects only 10 unique genes in total, where ‘AR’ is selected in three of the 5 folds.

GLMNET selected 160 different covariates across all 5 folds. Only five out of 24 selected covariates in Table 4 were selected in all five training datasets in cross validation. Those include ‘Age’, ‘Stage’ and ‘AR’. GLMNET was run 100 times for each fold. ISIS-SCAD selected 45 different covariates and only ‘Stage 3’ was selected in all training datasets in cross validation. The Stability Selection method selected sparser models compared to ISIS-SCAD and GLMNET by selecting 13 different covariates. It picked only ‘Age’ and
Fig 4: Average AUC of different variable selection methods based on a five fold cross validation for KIRC dataset.

‘Stage 3’ in all five folds.

Figures 5a and 5b compare IBS and prediction error curves, respectively, between different methods for the KIRC dataset. These two measures were computed based on a five-fold cross validation. Computation of IBS and prediction error were done using the R package pec (Mogensen, Ishwaran and Gerds, 2012). A benchmark model based on the Kaplan-Meier estimate, which includes no covariates, was also added to the figures as a reference for the comparison. The c-index measures are also reported in Table 5. The c-index was computed as it was in Section 4 using the dynpred package in R.

<table>
<thead>
<tr>
<th>c-index measure</th>
<th>BVSNLP</th>
<th>GLMNET</th>
<th>ISIS-SCAD</th>
<th>Stability Selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.804</td>
<td>0.816</td>
<td>0.846</td>
<td></td>
<td>0.797</td>
</tr>
</tbody>
</table>

GLMNET has almost the same IBS curve as the reference Kaplan Meier curve. BVSNLP outperforms ISIS-SCAD and Stability selection has the best IBS performance among all. For prediction error curves, BVSNLP is second to ISIS-SCAD, and GLMNET and Stability Selection have almost the same performance. A different behavior can be seen for c-index measures where GLMNET and ISIS-SCAD have higher c-indices than BVSNLP.

BVSNLP was run on 120 CPUs, Stability Selection was run on 4 CPUs,
while GLMNET and ISIS-SCAD were run on a single CPU. The average run-time for different methods in each fold of the cross validation was 6.4, 180, 5, and 1.3 minutes for BVSNLP, GLMNET, ISIS-SCAD and Stability Selection.

In our previous study of binary outcomes using the same dataset (Nikooienejad, Wang and Johnson, 2016), we performed hierarchical clustering on the de-convolved tumor-specific expression matrix and identified two clusters of patient samples. We saw these two groups of patients present significantly different survival outcomes and therefore assigned good vs. bad survival to the groups. The dichotomization was based solely on the clustering results of de-convolved gene expression levels. Survival times and censoring did not play any role in that process. However, there was a loss of information in dichotomizing a survival dataset and analyzing it with logistic regression. Now, with BVSNLP, we are able to use the original survival time to event with censoring information. To compare the biological implications between the two analyses, we looked for known expression regulation networks between the gene sets found in the binary analysis, SAV1 and NUMBL, and the new genes found in this analysis, AR and SUDS3, using Pathway Studio® (Nikitin et al., 2003; Elsevier, 2018). We found that the well-studied cancer genes TGFB1, BCL2, PPARG, NEDD4, and CTNNB1, and a regulatory microRNA, MIR21, constitute the shortest paths between SAV1 and AR. Similarly, we found cancer genes CDKN1A, WNT3A, two genes that determine cell fate (SOX17 (connected with CTNNB1) and NANOG), and PAX6 that regulates transcription, to constitute the shortest paths between NUMBL and SUDS3. These are depicted in Section 6 of the Supplementary
Material (Nikooienejad et al., 2020). These findings suggest a high biological consistency between our two analyses that used BVSNLP to select features for binary and survival outcomes.

In summary, the binary model using SAV1 and NUMBL to predict overall survival of patients with kidney cancer is not as effective as the model using AR and SUDS3, as shown in Figure 6. Thus, although the findings of Nikooienejad, Wang and Johnson (2016) were biologically justified, some limitations were associated with those findings due to the information loss incurred by clustering and dichotomizing the data, and the BVSNLP survival model provides better insight on the genes associated with this cancer type.

![Figure 6: Comparison between BVSNLP model selection using survival and dichotomized versions of the KIRC dataset.](image)

5.2. Kidney Renal Papillary Cell Carcinoma (KIRP). The KIRP dataset (Cancer Genome Atlas Research Network, 2016) contains 244 samples with 13,335 covariates (after necessary data cleaning) and has a fairly high censoring rate of 85.7%. The covariates selected by each method are summarized in Table 6.

In addition to the fixed covariates ‘Age’, ‘Gender’, and ‘Stage’, BVSNLP selects the ‘CDK1’ gene in the HPPM as the most significant covariate in the design matrix. The posterior inclusion probability for ‘CDK1’ was 0.12. The MAP estimates for the coefficients of ‘Age’, ‘Gender Male’, ‘Stage 2’, ‘Stage 3’ and ‘CDK1’ were 0.12, –0.10, 0.11, 0.79 and 1.13, respectively. This shows that a unit increase in ‘CDK1’ (Cyclin dependent kinase 1) gene expression...
Table 6
Selected covariates for KIRP across different variable selection algorithms

<table>
<thead>
<tr>
<th>Method</th>
<th>Age</th>
<th>Gender</th>
<th>Stage</th>
</tr>
</thead>
<tbody>
<tr>
<td>BVSNLP</td>
<td></td>
<td></td>
<td>Age, Gender</td>
</tr>
<tr>
<td>CDK1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ISIS-SCAD</td>
<td>CDK1</td>
<td>COL6A1</td>
<td>C19orf33</td>
</tr>
<tr>
<td>GLMNET</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stability Selection</td>
<td>Stage 3 MTC02P12 RPL39P3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

increases the hazard rate by a factor of 3, for given values of the other covariates. CDK1 is a cell cycle regulator and has been reported previously as a prognostic marker gene for various cancer types. Many experimental studies have been performed to further understand the molecular mechanism behind the complex functions of CDK1 (Malumbres and Barbacid, 2009). This is the first time, however, that CDK1 has been reported as a prognostic marker gene in human data for papillary renal cell carcinoma. As expected, patients at the most advanced stage cancer have a hazard rate that is 2.2 times higher than patients at a localized stage of cancer, given the values of all other covariates. As in the case of KIRC patients, age and male gender have unfavorable and favorable impacts on the hazard rate, respectively.

Surprisingly, GLMNET does not select any covariates. Stability Selection picked 3 covariates, with only ‘Stage 3’ in common with BVSNLP. As in the previous dataset, we tested BVSNLP for different choices of $\tau$ in the interval

Fig 7: Average AUC of BVSNLP based on a five fold cross validation for the KIRP dataset.
[0.01, 0.9] and the same model was selected for all values within this range. The total run-time of BVSNLP for this dataset was around 5 minutes using 120 CPUs.

Figure 7 shows the predictive accuracy for the proposed method based on a five-fold cross-validation. The outcomes for GLMNET, ISIS-SCAD and Stability Selection are not displayed in the plot because those methods did not converge or failed to produce results for at least one of the five folds in the cross-validation experiment. The small AUC values in this plot for $t < 1$ warrant comment. Because there were few events soon after entry of tissue samples into the TCGA database, the AUC for early timepoints falls close to the 50% benchmark reflecting no predictive value.

Figures 8a and 8b respectively depict IBS and prediction error curves of the BVSNLP method, based on a five-fold cross validation for the KIRP dataset and compares it to the reference curve obtained by the Kaplan Meier method. The c-index measure for the BVSNLP method is 0.876. The average run-time for BVSNLP in each fold of the cross validation was 3.6 minutes on 120 CPUs.

6. Discussion. In this article a Bayesian variable selection method, BVSNLP, was proposed for selecting variables in high and ultrahigh dimensional datasets with survival time as outcomes. BVSNLP uses an inverse moment nonlocal prior density on non-zero regression coefficients. Analyses of simulated and real data suggest that BVSNLP performs comparably or better than other existing methods for variable selection for survival data. Moreover, the real data results indicated that the proposed algorithm is ro-
bust to the choice of the hyperparameter $\tau$ in the piMOM prior for values of $\tau$ in the range $[0.01, 0.9]$. 

Various outputs are provided by the algorithm. These include the HPPM, MPM and the posterior inclusion probability for each covariate in the model. For real datasets, Bayesian model averaging is used to incorporate uncertainty in selected models when computing time dependent AUC plots using Uno’s method (Uno et al., 2007). Finally, an R package named BVSNLP has been implemented to make the algorithm freely available and adaptable to interested researchers. The package can be run in parallel fashion where hundreds of CPUs can be exploited in order to increase the number of visited models in the search for the highest posterior probability model. The BVSNLP package is available in the R repository, CRAN, at https://CRAN.R-project.org/package=BVSNLP. The user manual for the package is also available from this site.

Two real cancer genomic datasets from the TCGA website were considered in this article. Compared to other methods, BVSNLP found sparser models with biologically relevant genes. The proposed method showed a reliable predictive accuracy as measured by AUC using substantially fewer variables.

We have based our assessments on time dependent AUC and biological interpretation of the results, but other measures, like IBS, prediction error and the concordance index (also known as the c-index or Harrell’s c-index) were also reported. Difficulties associated with such measures are identified in Blanche, Kattan and Gerds (2018). In particular, the authors of that article demonstrate that the concordance index can favor misspecified models over the correctly specified model because it is based on the order of event times rather than the event status at the prediction horizon. This may explain the slightly higher c-index values for GLMNET in both simulation and real data sets. The time dependent AUC does not suffer from this deficiency. Of course, different evaluation criteria can be expected to result in different rankings of models, and criteria that emphasize prediction error over low false positive rates can be expected to favor larger models. Similarly, criteria that place a higher premium on eliminating false positives will tend to select smaller models.

SUPPLEMENTARY MATERIAL

Supplementary Material: “Bayesian Variable Selection For Survival Data Using Inverse Moment Priors”.

(doi: Completed by the typesetter). The supplementary material to this article containing some of the details regarding the methodology, algorithm
for implementing that methodology, and some extended discussion on the results.

References.


