Biostatistics (2019), **0**, 0, *pp.* 1–14
doi:10.1093/biostatistics/SupplementaryMaterial˙R1

Supplementary Material: Gaussian process regression for survival time prediction with genome-wide gene expression

AARON J. MOLSTAD, LI HSU, WEI SUN*

Biostatistics Program, Fred Hutchinson Cancer Research Center, Seattle, WA

wsun@fredhutch.org

1. Derivation of Algorithm 2

To solve the optimization problem in Step 2 of Algorithm 1 when $M \geqslant 1$, we use a blockwise coordinate descent algorithm. Specifically, we update $\boldsymbol{\beta}$ with $\tilde{\boldsymbol{\sigma}}^2$ fixed and vice versa. Because the blockwise update for $\tilde{\boldsymbol{\sigma}}^2$ is computationally challenging, we use a modified version of the algorithm proposed by Zhou et al. (2018) to update $\tilde{\boldsymbol{\sigma}}^2$. Briefly, their algorithm exploits the minorize-maximize principle: its iterates *maximize* a *minorizing* function at the previous iterate. Let $h(\theta)$ denote the original objective function evaluated at θ. Let $g(\cdot \mid \theta^-)$ be a function of θ^-. For $g(\cdot \mid \theta^-)$ to a minorization of $h(\cdot)$, it must satisfy two conditions: (i) the tangency condition $g(\theta^- \mid \theta^-) = f(\theta^-)$ for all θ^- and (ii) the domination condition $g(\theta \mid \theta^-) \leqslant h(\theta)$ for all θ. Then, for $\theta^+ = \arg\max_\theta g(\theta \mid \theta^-)$, it follows that

$$h(\theta^+) \geqslant g(\theta^+ \mid \theta^-) \geqslant g(\theta^- \mid \theta^-) = h(\theta^-),$$

where the first inequality follows from the domination condition, the second follows from the definition of θ^+, and the equality follows from the tangency condition. Thus, since $h(\theta^+) \geqslant h(\theta^-)$,

*To whom correspondence should be addressed.

we are ensured to monotonically increase our objective function. Generalizing this approach suggests sequentially updating θ for $b = 1, 2, \ldots,$ using

$$\theta^{(b+1)} = \arg\max_{\theta} g(\theta \mid \theta^{(b)}),$$

until the original objective function value converges. This approach is especially useful when a minorizing function g exists such that $\arg\max_\theta g(\theta \mid \theta^{(b)})$ is easier to solve than $\arg\max_\theta h(\theta)$. See Lange (2016) for more on the minorize-maximize principle.

Focusing specifically on Algorithm 2, we start by deriving the update for $\boldsymbol{\beta}$ with $\tilde{\boldsymbol{\sigma}}^{2(b)}$ fixed. This subproblem is convex and can be solved in closed form, so we do not require the minorize-maximize approach. In particular, we solve

$$\boldsymbol{\beta}^{(b+1)} = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^{q+1}} \left[s_r^{-1} \sum_{j=1}^{s_r} \operatorname{tr}\left\{ (\hat{T}_j - Z\boldsymbol{\beta})(\hat{T}_j - Z\boldsymbol{\beta})' \tilde{K}(X, \tilde{\boldsymbol{\sigma}}^{2(b)})^{-1} \right\} \right],$$

where tr denotes the trace operator. The first order conditions for optimality are

$$Z'\tilde{K}(X, \tilde{\boldsymbol{\sigma}}^{2(b)})^{-1} Z\boldsymbol{\beta} = s_r^{-1} \sum_{j=1}^{s_r} Z\tilde{K}(X, \tilde{\boldsymbol{\sigma}}^{2(b)})^{-1} \hat{T}_j,$$

so that letting $\bar{T} = s_r^{-1} \sum_{j=1}^{s_r} \hat{T}_j$, we have

$$\boldsymbol{\beta}^{(b+1)} = \left[Z'\tilde{K}(X, \tilde{\boldsymbol{\sigma}}^{2(b)})^{-1} Z \right]^{-1} \left[Z'\tilde{K}(X, \tilde{\boldsymbol{\sigma}}^{2(b)})^{-1} \bar{T} \right],$$

which is Step 2 of Algorithm 2. We then derive the updating equations in Steps 3 and 4 using the minorize-maximize approach from Zhou et al. (2018). Treating $\boldsymbol{\beta}^{(b+1)}$ as fixed, redefine h as

$$h(\tilde{\boldsymbol{\sigma}}^2, \boldsymbol{\beta}^{(b+1)}) = -s_r^{-1} \sum_{j=1}^{s_r} (\hat{T}_j - Z\boldsymbol{\beta}^{(b+1)})' \tilde{K}(X, \tilde{\boldsymbol{\sigma}}^2)^{-1} (\hat{T}_j - Z\boldsymbol{\beta}^{(b+1)}) - \log\det\{\tilde{K}(X, \tilde{\boldsymbol{\sigma}}^2)\},$$

where det denotes the determinant operator. Let $\tilde{\boldsymbol{\sigma}}^{2(b)}$ denote the previous iterate of $\tilde{\boldsymbol{\sigma}}^2$. Modifying (5) from Zhou et al. (2018),

$$\begin{aligned} &-(\hat{T}_j - Z\boldsymbol{\beta}^{(b+1)})' \tilde{K}(X, \tilde{\boldsymbol{\sigma}}^2)^{-1} (\hat{T}_j - Z\boldsymbol{\beta}^{(b+1)}) \\ &\quad \geqslant -(\hat{T}_j - Z\boldsymbol{\beta}^{(b+1)})' \tilde{K}(X, \tilde{\boldsymbol{\sigma}}^{2(b)})^{-1} \left\{ \sum_{s=1}^{M+1} \frac{\sigma_s^{4(b)}}{\sigma_s^2} k_s(X, X) \right\} \tilde{K}(X, \tilde{\boldsymbol{\sigma}}^{2(b)})^{-1} (\hat{T}_j - Z\boldsymbol{\beta}^{(b+1)}) \end{aligned} \tag{1.1}$$

where for notational simplicity, we define $\sigma^2_{M+1} \equiv \sigma^2_\epsilon$ and $k_{M+1}(X,X) \equiv I_n$. Modifying (6) from Zhou et al. (2018),

$$-\log\det\{\tilde{K}(X,\tilde{\boldsymbol{\sigma}}^2)\} \geqslant -\log\det\{\tilde{K}(X,\tilde{\boldsymbol{\sigma}}^{2(b)})\} - \mathrm{tr}\left[-\tilde{K}(X,\tilde{\boldsymbol{\sigma}}^{2(b)})^{-1}\{\tilde{K}(X,\tilde{\boldsymbol{\sigma}}^2) - \tilde{K}(X,\tilde{\boldsymbol{\sigma}}^{2(b)})\}\right] \tag{1.2}$$

It is straightforward to check that both the domination and tangency conditions hold for the right hand sides of (1.1) and (1.2). Thus, a natural minorization of $h(\tilde{\boldsymbol{\sigma}}^2, \boldsymbol{\beta}^{(b+1)})$ combines (1.1) and (1.2): ignoring constants which do not depend on $\tilde{\boldsymbol{\sigma}}^2$, the minorizing function at $\tilde{\boldsymbol{\sigma}}^{2(b)}$ is

$$\begin{aligned} g(\tilde{\boldsymbol{\sigma}}^2, \boldsymbol{\beta}^{(b+1)} | \tilde{\boldsymbol{\sigma}}^{2(b)}) = -s_r^{-1} \sum_{j=1}^{s_r} \sum_{s=1}^{M+1} \Big[& \sigma_s^2 \mathrm{tr}\left\{ k_s(X,X) \tilde{K}(X,\tilde{\boldsymbol{\sigma}}^{2(b)})^{-1} \right\} \\ & + \frac{\sigma_s^{4(b)}}{\sigma_s^2} (\hat{T}_j - Z\boldsymbol{\beta}^{(b+1)})' \tilde{K}(X,\tilde{\boldsymbol{\sigma}}^{2(b)})^{-1} k_s(X,X) \tilde{K}(X,\tilde{\boldsymbol{\sigma}}^{2(b)})^{-1} (\hat{T}_j - Z\boldsymbol{\beta}^{(b+1)}) \Big]. \end{aligned} \tag{1.3}$$

Following the minorize-maximize principle, we update $\tilde{\boldsymbol{\sigma}}^2$ by solving:

$$\tilde{\boldsymbol{\sigma}}^{2(b+1)} = \underset{\tilde{\boldsymbol{\sigma}}^2 \in \mathbb{R}_+^M \times \mathbb{R}_+}{\arg\min}\ g(\tilde{\boldsymbol{\sigma}}^2, \boldsymbol{\beta}^{(b+1)} | \tilde{\boldsymbol{\sigma}}^{2(b)}). \tag{1.4}$$

As noted in Zhou et al. (2018), (1.4) is separable across components of $\tilde{\boldsymbol{\sigma}}^2$. In particular, $g(\tilde{\boldsymbol{\sigma}}^2, \boldsymbol{\beta}^{(b+1)} | \tilde{\boldsymbol{\sigma}}^{2(b)})$ is minimized with respect to the sth component of $\bar{\boldsymbol{\sigma}}^2$ when

$$\sigma_s^{2(b+1)} = \frac{\sigma_s^{2(b)}}{\sqrt{s_r}} \left[\frac{\sum_{j=1}^{s_r} (\hat{T}_j - Z\boldsymbol{\beta}^{(b+1)})' \tilde{K}(X,\tilde{\boldsymbol{\sigma}}^{2(b)})^{-1} k_s(X,X) \tilde{K}(X,\tilde{\boldsymbol{\sigma}}^{2(b)})^{-1} (\hat{T}_j - Z\boldsymbol{\beta}^{(b+1)})}{\mathrm{tr}\left[k_s(X,X) \tilde{K}(X,\tilde{\boldsymbol{\sigma}}^{2(b)})^{-1} \right]} \right]^{1/2},$$

which are the updating equations in Steps 3 and 4 of Algorithm 2. Putting both the $\boldsymbol{\beta}^{(b+1)}$ and $\tilde{\boldsymbol{\sigma}}^{2(b+1)}$ updates together ensures:

$$\begin{aligned} h(\tilde{\boldsymbol{\sigma}}^{2(b+1)}, \boldsymbol{\beta}^{(b+1)}) &\geqslant g(\tilde{\boldsymbol{\sigma}}^{2(b+1)}, \boldsymbol{\beta}^{(b+1)} \mid \tilde{\boldsymbol{\sigma}}^{2(b)}) && (1.5) \\ &\geqslant g(\tilde{\boldsymbol{\sigma}}^{2(b)}, \boldsymbol{\beta}^{(b+1)} \mid \tilde{\boldsymbol{\sigma}}^{2(b)}) && (1.6) \\ &= h(\tilde{\boldsymbol{\sigma}}^{2(b)}, \boldsymbol{\beta}^{(b+1)}) && (1.7) \\ &\geqslant h(\tilde{\boldsymbol{\sigma}}^{2(b)}, \boldsymbol{\beta}^{(b)}) && (1.8) \end{aligned}$$

where (1.5) follows from the minorization condition, (1.6) follows from the definition of $\tilde{\boldsymbol{\sigma}}^{2(b+1)}$ from (1.4), (1.7) follows from the tangency condition, and (1.8) follows from the fact that h is

convex with respect to $\boldsymbol{\beta}$ and that $\boldsymbol{\beta}^{(b+1)}$ is its global minimizer. Hence, Algorithm 2 has the "strict ascent" property, meaning that its iterates monotonically increase the objective function.

2. Sensitivity analysis of MC-EM algorithm

In this section, we consider the robustness of our MC-EM algorithm to initial values and examine the Monte-Carlo error at convergence. To do so, we fit the Gaussian process accelerated failure time model to the complete KIRC dataset one hundred times using both genome-wide and pathway-based kernels, i.e., `GPR:K` and `GPR:M`. Because Algorithm 1 requires random data generation, the path of iterates will be different each of the hundred times we fit the model. We also considered two types of initializing values: one sets all variance components equal to one, while the other uses initializing values randomly drawn from a uniform distribution on the interval $[0.5, 5]$.

In Figure 1, we display the values of the iterates for the four variations of our method. Within each plot, one hundred model fits are displayed, with each line denoting one variance component's value at a certain iteration for one model fit. We notice that when the same initial values are used, i.e., in (a) and (c), the paths of the iterates tend to be very similar, with some variation coming from the Monte-Carlo error. Nevertheless, we see that as the iteration count increases, the Monte-Carlo error seems to decrease and parameter values at convergence are nearly equivalent for all one hundred model fits.

3. Additional simulation studies

3.1 *Effect of censoring proportion*

To examine the effect of censoring on the performance of our method, we conducted additional simulations wherein we modified the settings from the main manuscript to have varying censoring rates. For one hundred independent replications, we generated uncensored survival times from

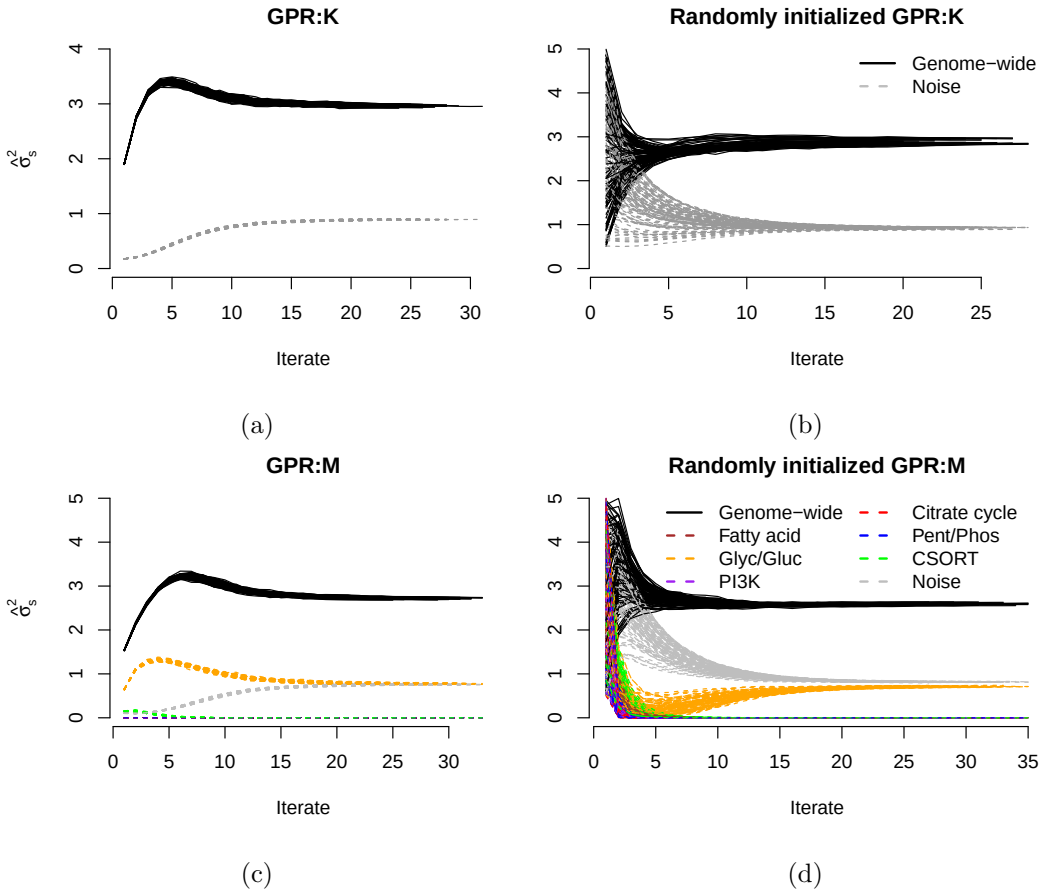
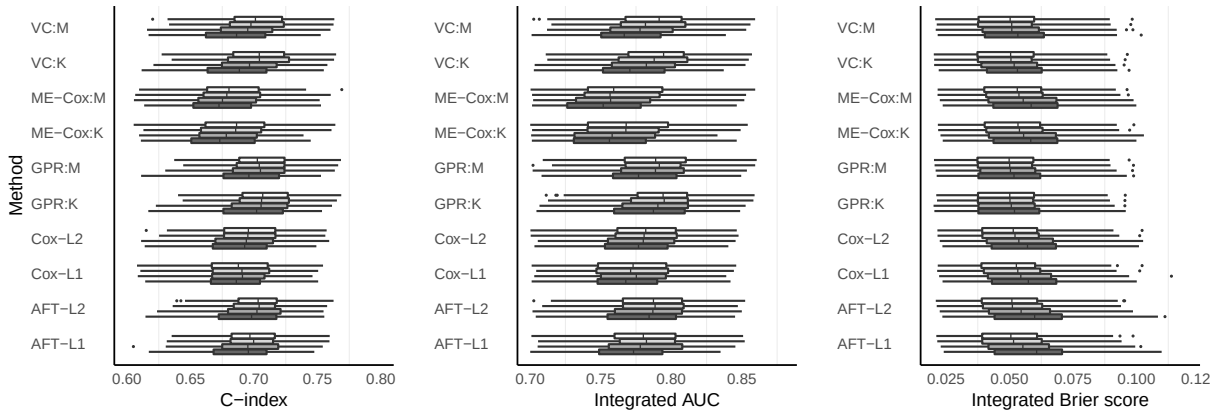


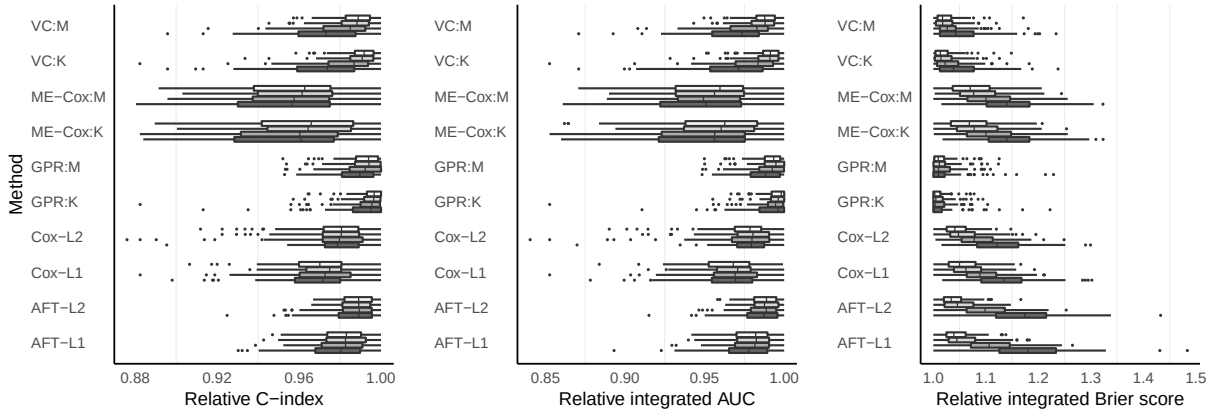
Fig. 1. Plots displaying the iterates of the variance components when fitting the Gaussian process accelerated failure time model to the complete KIRC data using Algorithm 1. We display iterates for the genome-wide kernel with (a) all initial values equal to one and (b) initial values randomly drawn from a uniform distribution. In (c), we display iterates for the pathway-based kernels with all initial values equal to one and in (d), with all initial values random drawn from a uniform distribution.

Model 1 of the main manuscript using the genome-wide kernel. For each of the hundred datasets, we also generated four independent sets of censoring times so that we could use the considered methods on the same dataset with multiple censoring rates.

Censoring times were drawn from an exponential distribution with mean $\left\{Q_\tau(\{S_j\}_{j=1}^n)\right\}^{-1}$ and Q_τ denotes the τth quantile. Unlike in the main manuscript, τ was the same for all i: we considered $\tau \in \{0.40, 0.60, 0.80, 0.90\}$ which led to datasets with average censoring rates of approximately $0.67, 0.53, 0.35$, and 0.24 respectively. In Figure 2, we display the C-index, integrated



(a)



(b)

Fig. 2. Boxplots of (relative) C-index, integrated AUC, and integrated Brier scores for multiple methods based on different censoring time data generating models. Boxplots in (a) display the metrics directly, whereas the boxplots in (b) display the relative metrics. Lightest grey boxplots correspond to the highest censoring proportion; darkest grey correspond to the lowest censoring proportion. Censoring proportions were approximately $0.67, 0.53, 0.35$, and 0.24.

AUC, and integrated Brier Scores for a subset of methods included in the main manuscript simulations. Pre-screened methods were also considered, but performed worse then their genome-wide counterparts, and thus were omitted from the displayed results.

As one may expect, the censoring proportion has an effect on all methods' performance. In Figure 2(a), we notice that C-index, integrated AUC, and integrated Brier scores are improved

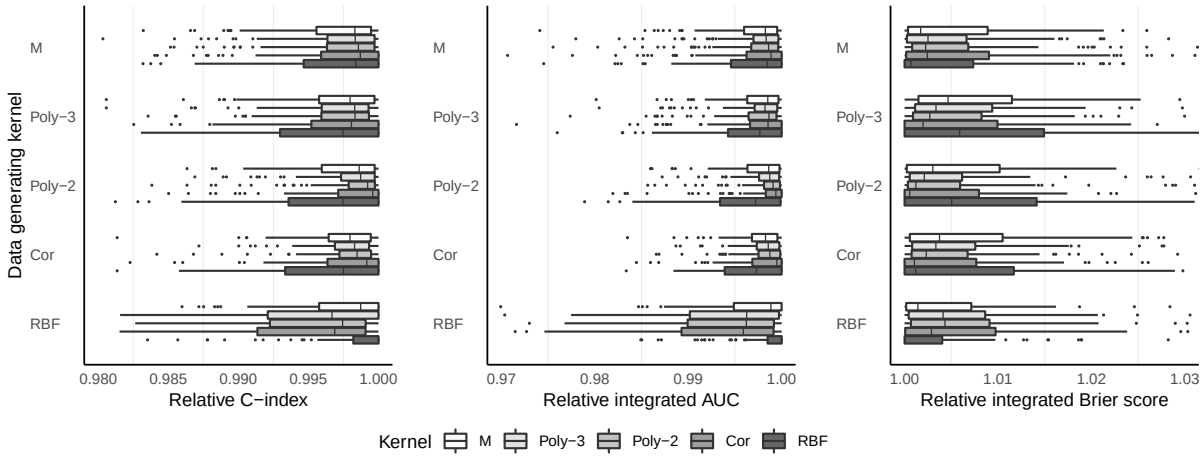


Fig. 3. Boxplots of relative C-index, integrated AUC, and integrated Brier score for five kernel specifications across five data generating models.

as censoring proportions decrease. In Figure 2(b), we display relative C-index, relative integrated AUC, and relative integrated Brier score. The relative performances suggest that as censoring proportions decrease, our method using the genome-wide kernel clearly outperforms all competitors. At the highest censoring rate, our method still performs best, although the differences are substantial for relative C-index and relative integrated AUC.

3.2 *Effect of kernel choice*

To study the sensitivity of our method to the choice of kernel, we performed additional simulations under multiple kernel specifications. For one hundred independent replications, we generated data from Model 1 of the main manuscript, using the same distribution for censoring times, where as before, $\eta \sim \mathrm{N}_n\left\{0, K(X, \boldsymbol{\sigma}^2)\right\}$ where $[K(X, \boldsymbol{\sigma}^2)]_{j,k} = 3\tilde{k}(x_j, x_k)$ for $j \neq k$ where $\tilde{k}$ had one of the following forms:

`RBF`: Radial basis kernel function:

$$\tilde{k}(x_j, x_k) = \exp\left\{\frac{\|x_j - x_k\|_2}{\max_{l,m}\|x_j - x_k\|_2}\right\}, \quad (j,k) \in \{1,\ldots,n\} \times \{1,\ldots,n\}\,.$$

`Cor`: Correlation matrix kernel function

$$\tilde{k}(x_j, x_k) = \frac{\tilde{x}_j' \tilde{x}_k}{\|\tilde{x}_j\|_2 \|\tilde{x}_k\|_2}, \quad (j,k) \in \{1,\dots,n\} \times \{1,\dots,n\},$$

where $\tilde{x}_j = x_j - 1_p \frac{1}{p}(\sum_{s=1}^{p} x_{js})$

`Poly-2`: Normalized polynomial kernel of order two

$$\tilde{k}(x_j, x_k) = \frac{(x_j' x_k)^2}{\|x_j\|_2^2 \|x_k\|_2^2} \in \{1,\dots,n\} \times \{1,\dots,n\}.$$

`Poly-3`: Normalized polynomial kernel of order three

$$\tilde{k}(x_j, x_k) = \frac{(x_j' x_k)^3}{(\|x_j\|_2^3 \|x_k\|_2^3)} \in \{1,\dots,n\} \times \{1,\dots,n\}.$$

`M`: The weighted sum of the four previous kernels with $\sigma_s^2 = 3/4$ for $s = 1,\dots,4$.

In each replication, we fit the Gaussian process accelerated failure time model using all five of the aforementioned kernels. Thus, for each replication, only one of the Gaussian process accelerated failure time models is correctly specified, although the multiple kernel version, `M`, always has the correctly specified model as a special case.

In Figure 3, we display the relative performances of each of the five kernel specifications. We see that when `RBF` is the correct kernel, the correctly specified model substantially outperforms the others. The multiple kernel version `M` performs only slightly worse in this case, which is expected. With the other data generating kernels, the correctly specified model performs best, except `Poly-3`, where we see that `Poly-2` and `Cor` also perform well.

3.3 *Effect of training sample size*

To analyze the performance of our method under varying sample sizes, we obtained RNAseq data from 832 patients with breast invasive carcinoma (BRCA) collected by TCGA using the `TCGA2STAT` package in `R`. We normalized the RNAseq counts in the same manner as described in

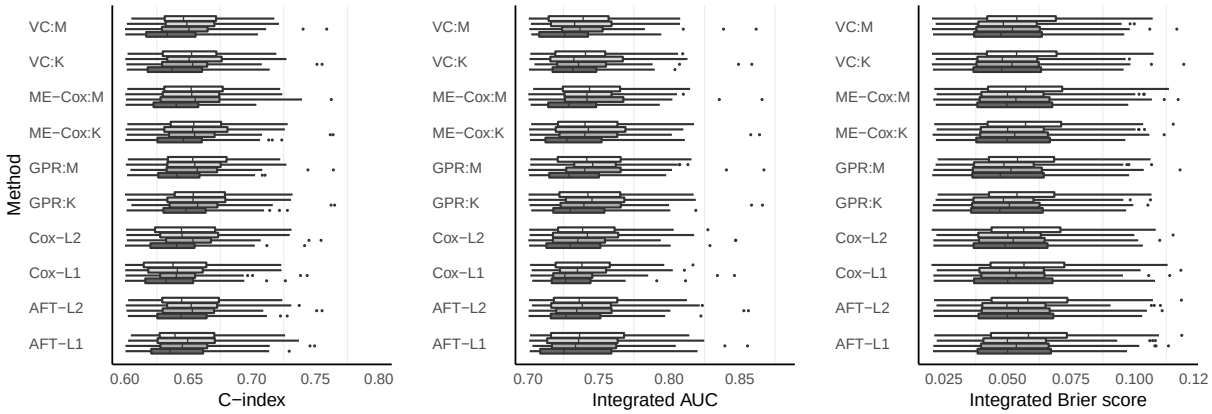
Section 5.1 of the main manuscript. We also obtained clinical/demographic data including age (in days) and tumor stage, which had three levels in the BRCA dataset we used.

To analyze the effect of training sample size, we used subsets of the complete BRCA dataset. Namely, for one hundred independent replications, we first randomly selected 100 subjects to serve as testing data and randomly selected either 432, 532, 632, or 732 subjects to serve as training data. In each replication, we generated data from the following model:

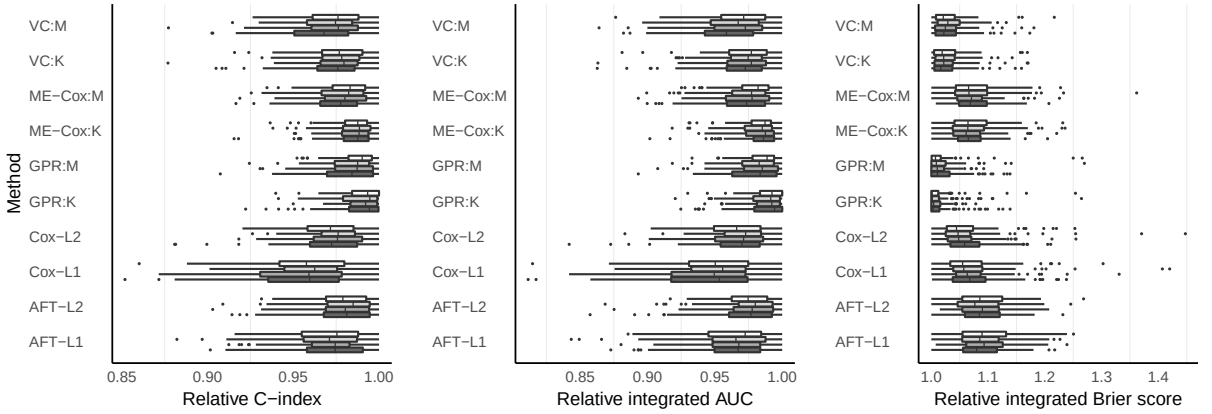
> *Gaussian process AFT model.* Log-survival times are generated as a realization of the Gaussian process accelerated failure time model:
>
> $$T = Z\boldsymbol{\beta} + \eta + \gamma,$$
>
> where $\gamma \sim \mathrm{N}_n\{0, 0.5I_n\}$ and $\eta \sim \mathrm{N}_n\left\{0, K(X, \boldsymbol{\sigma}^2)\right\}$ with $K(X, \boldsymbol{\sigma}^2)$ defined below and $\boldsymbol{\beta} = (6.2, -0.3, -1.1, -1 \times 10^{-5})$ where the columns of Z corresponds to the intercept, tumor stage II, tumor stage III, and age in days.

As in the main manuscript, we set $K(X, \boldsymbol{\sigma}^2)$ to equal the genome-wide kernel from equation (4.8) with $\sigma_G^2 = 3$. The ith subject's censoring time is drawn from an exponential distribution with mean $\left\{Q_{\tau_i}(\{S_j\}_{j=1}^n)\right\}^{-1}$ where Q_c denotes the cth quantile and $\tau_i = .20, .70$, or $.80$ for subjects with tumor stages I, II, or III respectively.

Relative performance metrics are displayed in Figure 4(b). Our primary observation is that as the training sample size increases to 732 (displayed in lightest grey boxplots), both `GPR:K` and `GPR:M` tend to more clearly outperform the competitors. Interestingly, while it appears that in general, increasing sample size seemed to improve both C-index and integrated AUC, integrated Brier scores actually increased slightly as the training sample size increases.



(a)



(b)

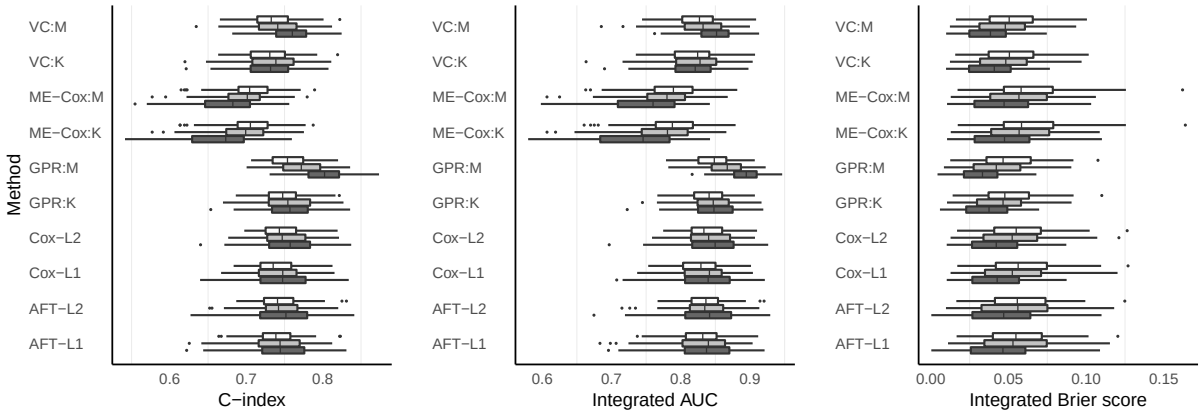
Fig. 4. Boxplots of (relative) C-index, integrated AUC, and integrated Brier scores for multiple methods under different training sample sizes as described in Section 3.3. Lightest grey boxplots correspond to the largest training sample size (732); and darkest grey correspond to the smallest training sample size (432).

3.4 *Performance under sparse AFT model*

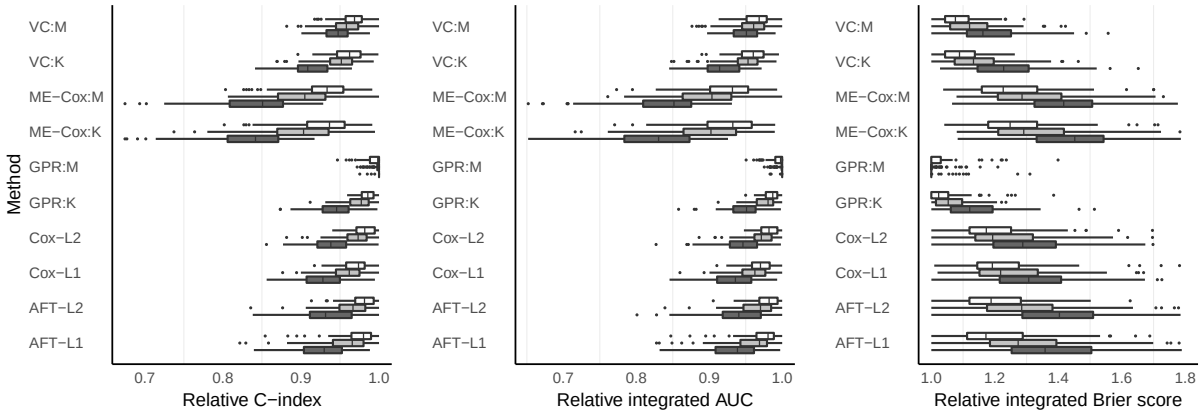
In this section, we analyze the performance of our method when the genetic contribution to survival is through a sparse linear model. For one hundred independent replications, we generate data from the following model:

> *Gaussian AFT model:* Log-survival times are generated as a realization of the accelerated failure time model:
>
> $$T = Z\boldsymbol{\beta} + X\boldsymbol{\eta} + \epsilon,$$
>
> where $X \in \mathbb{R}^{n\times p}$ denotes the standardized gene expression matrix (i.e., X has column-wise average zero and standard deviation one), and $\boldsymbol{\eta} \in \mathbb{R}^p$ has $50, 100$, or 150 randomly selected nonzero entries with magnitudes $5^{-1}, 10^{-1}$, or 15^{-1}, respectively, and the entries' signs assigned randomly. In addition, we set $\boldsymbol{\beta} = (6.2, -0.5, -1.2, -2.0, -1 \times 10^{-5})$ where the columns of Z corresponds to the intercept, tumor stage II, tumor stage III, tumor stage IV, and age in days. Finally, $\epsilon \sim \mathrm{N}_n\,(0, .5I_n)$.

We consider the same competitors as in other scenarios. However, in this setting, the pathway-based kernels used for `GPR:M` contain one kernel computed using only genes corresponding to the nonzero elements of $\boldsymbol{\eta}$, so in this sense, `GPR:M` has unrealistic oracle knowledge but serves as a best-case for our method. Results are displayed in Figure 5, where we note that `GPR:K`, the version of our method using a genome-wide gene expression kernel, actually outperforms both linear AFT variations. This may partly be due to the fact that our model correctly specifies the error distribution, so the imputation scheme we use is more accurate than the AFT variations which are nonparameteric.



(a)



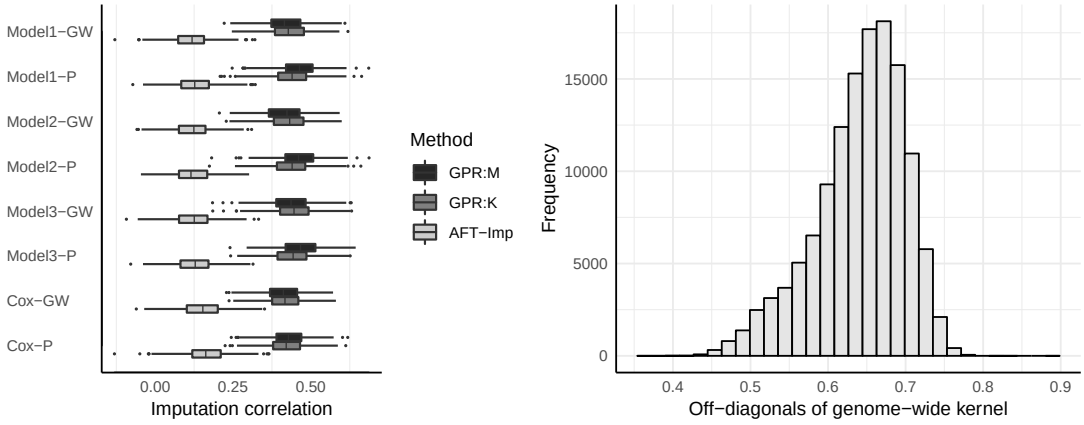
(b)

Fig. 5. Boxplots of (relative) C-index, integrated AUC, and integrated Brier scores for multiple methods under different levels of sparsity in the Gaussian AFT model described in Section 3.4. Boxplots in (a) display the metrics directly, whereas the boxplots in (b) display the relative metrics. Lightest grey boxplots correspond to 150 nonzero effects, grey correspond to 100 nonzero effects; and darkest grey correspond to 50 nonzero effects.

4. Additional figures

In Figure 6(a), we display boxplots showing the correlation between the true log-survival time and the imputed log-survival for the censored training data from the simulation studies. The imputations compared are those obtained using the iterative method of Grimes et al. (2018) based only on clinical/demographic variables, and those at convergence of our algorithm for both `GPR:K` and `GPR:M`.

In Figure 6(b), we display a histogram of the off-diagonal entries of the radial basis kernel used to define the genome-wide effect in the simulation studies.



(a) (b)

Fig. 6. (a) Correlations between the imputed log-survival time and true log-survival time for the censored outcomes in the training sets. (b) A histogram showing the off-diagonal entries of the normalized radial basis kernel using genome-wide gene expression.

References

Grimes, T., Walker, A. R., Datta, S., and Datta, S. (2018). Predicting survival times for neuroblastoma patients using rna-seq expression profiles. *Biology direct*, 13(1):11.

Lange, K. (2016). *MM optimization algorithms*, volume 147. SIAM.

Zhou, H., Hu, L., Zhou, J., and Lange, K. (2018). MM algorithms for variance components models. *Journal of Computational and Graphical Statistics*, (just-accepted):1–30.