Cross-Validation of Survival Bump Hunting by Recursive Peeling Methods

Jean-Eudes Dazard * Michael Choe † Michael LeBlanc ‡ J. Sunil Rao §

Abstract

We introduce a survival/risk bump hunting framework to build a bump hunting model with a possibly censored time-to-event type of response and to validate model estimates. First, we describe the use of adequate survival peeling criteria to build a survival/risk bump hunting model based on recursive peeling methods. Our method called "Patient Recursive Survival Peeling" is a rule-induction method that makes use of specific peeling criteria such as hazards ratio or log-rank statistics. Second, to validate our model estimates and improve survival prediction accuracy, we describe a resampling-based validation technique specifically designed for the joint task of decision rule making by recursive peeling (i.e. decision-box) and survival estimation. This alternative technique, called "combined" cross-validation is done by combining test samples over the cross-validation loops, a design allowing for bump hunting by recursive peeling in a survival setting. We provide empirical results showing the importance of cross-validation and replication.

Key Words: K-Fold Cross-Validation, Bump Hunting, Non-Parametric Survival Analysis, Patient Rule-Induction Method, Survival/Risk Estimation, Survival/Risk Prediction

1. Introduction

Model Development and Validation in Discovery-Based Research

The primary problem encountered in discovery-based research has been non-reproducible results. For instance, early biomarker discovery studies using modern high-throughput datasets with large number of features have often been characterized by false or exaggerated claims [9, 12, 22, 26, 28]. This has been attributed to a lack of proper rules to assess the analytical validity of studies simply because they were either under-developed or not routinely applied. Since then, however, the problem has received considerable attention and developmental work from statisticians in the fields of feature selection, predictive model building and model validation (see e.g. reviews on guidelines and checklists [3, 9, 20]), as well as from recent editors and US regulators [21].

Regardless of dimensionality issues, one strategy to address the lack of model reliability and reproducibility is to use large enough sample sizes in conjunction with proper validation techniques for model development and model performance assessment. The problems of model reliability and reproducibility have usually been characterized by issues of severe model overfitting, biased model estimates, and under-estimated errors. A common situation where this arises is when, for instance, model performance estimates are made from the same data that was used for model building, eventually resulting in initially promising results, but often non-reproducible [2, 14, 26]. These so-called "resubstitution estimates" are severely (optimistically) biased. Another problematic situation is when not all the steps of model building (such as pre-selection, creation of the prediction rule and parameter tuning) are internal to the cross-validation process, thereby creating a selection bias [2, 14, 31]. In addition, findings might not be

*Division of Bioinformatics, Center for Proteomics and Bioinformatics, Case Western Reserve University. Cleveland, OH 44106, USA. Corresponding author Email (JED): jxd101@case.edu

†Division of Bioinformatics, Center for Proteomics and Bioinformatics, Case Western Reserve University. Cleveland, OH 44106, USA.

‡Department of Biostatistics, School of Public Health, University of Washington, Seattle, WA 98195, USA; Public Health Sciences, Fred Hutchinson Cancer Research Center, Seattle, WA 98109.

§Division of Biostatistics, Dept. of Epidemiology and Public Health, The University of Miami. Miami, FL 33136, USA.

reproducible even when using proper independent sample and validation procedures. Problems may arise in the validation steps itself because cross-validated error estimates are well-known to have very large variance, a situation that is obviously more prevalent when few independent observations or small sample size n are used [3, 8, 26].

Predictive Survival/Risk Modeling by Rule-Induction Methods

One important application of survival/risk modeling is to identify and segregate samples for predictive diagnostic and/or prognosis purposes. Direct applications include the stratification of patients by diagnostic and/or prognostic groups and/or responsiveness to treatment. Therefore, survival modeling is usually performed to predict/classify patients into risk or responder groups (not to predict exact survival time) from which one usually derives survival/risk functions estimates for these groups (e.g. by KaplanMeier estimates). However, for the reasons mentioned above, KaplanMeier estimates for the risk groups computed on the same set of data used to develop the survival model may be very biased [23, 31].

Although validation tools typically for evaluating classification models are often useful in assessing the prediction accuracy of classifier models, resampling methods are not directly applicable for predictive survival modeling applications. Simon et *al.* have reviewed the literature of such applications and identified serious deficiencies in the validation of survival risk models [9, 27, 28]. They noted for instance that in order to utilize the cross-validation approach developed for classification problems, some studies have dichotomized their survival or disease-free survival data .... The problem on how to cross-validate the estimation of survival distributions (e.g. by KaplanMeier curves) is not obvious [27]. In addition, beside Subramanian and Simon's initial study on the usefulness of resampling methods for assessing survival prediction models in high-dimensional data [29], no comparative study has been done for rule-induction methods and specifically recursive peeling methods such as our "Patient Recursive Survival Peeling" algorithm (see section 2.2.4).

In the context of a time-to-event outcome, rule-induction methods such as regression survival trees have proven to be useful. Several methods have been proposed for fitting decision trees to non-informative censored survival times [1, 4, 6, 11, 16, 17, 25, 30]. Although decision trees are powerful techniques for understanding patient outcome and for forming multiple prognostic groups, often times interest focuses only on the *extreme* prognostic groups. So, in contrast to usual regression survival trees, survival bump hunting aims not at estimating the survival/risk probability function over the entire variable space, but at searching regions where this probability is larger than its average over the entire space.

Also, one possible drawback of decision trees is that the data splits at an exponential rate as the space undergo partitioning (typically by binary splits) as opposed to a more patient rate in decision boxes (typically by controlled data quantile). In this sense, bump hunting by recursive peeling may be a more efficient way of learning from the data. With the exception of the work of LeBlanc *et al.* on Adaptive Risk Group Refinement [19], it has not been studied whether decision boxes, obtained from box-structured recursive peelings, would yield better estimates for constructing prognostic groups than their tree-structured counterparts.

Goal and Scope of the Paper

Our first objective in this paper is to describe the use of appropriate survival peeling criteria to fit a survival/risk bump hunting model based on recursive peeling methods. Our second objective is to develop a validation strategy of model estimates using a resampling technique amenable to the joint task of decision rule estimation and survival predictive accuracy. To develop our survival bump hunting model, we focused on a non-parametric rule-induction method, derived from a recursive peeling procedure, namely the Patient Rule Induction Method (PRIM) [10, 24], which we have extended to allow for survival/risk response, possibly censored. Although, several

resampling techniques are available (see section 3) such as full/complete cross-validation (K-fold CV), leave-one-out cross-validation (LOOCV) or bootstrap-based methods like the out-of-box 0.632 bootstrap cross-validation (0.632 OOB), in this study, we describe a full (K-fold) cross-validation-based resampling technique adapted to the task. We have limited ourselves to simulated datasets where $n \geqslant p$ for the only reason that the implementation used so far for fitting our survival bump hunting models do not allow for high-dimensional situation yet. The development of high-dimensional survival bump hunting models is work in progress beyond the scope of this paper. Although we did not specifically use simulated datasets where $p > n$, we posit that the cross-validation techniques presented here will be applicable to high-dimensional data as well.

Organization of the Paper
We first introduce the regular bump hunting framework upon which we built our survival bump hunting model to accommodate a possibly censored time-to-event type of response. In the following section, we show how we derived our so-called "Patient Recursive Survival Peeling" algorithm from the original Patient Rule Induction Method (PRIM) for bump hunting by recursive peeling in a survival setting. In the process, we describe which peeling criteria one may use as well as what specific survival endpoint statistics are of interest. In the subsequent section, we develop our own resampling and replication cross-validation technique, specifically designed for the joint task of decision rule making by recursive peeling (i.e. decision-box) and survival estimation. This allowed to get combined cross-validated survival bump hunting estimates, namely decision boxes and rules, survival distributions and endpoints statistics. Finally, we provide empirical results from simulated data, illustrating the efficiency of our alternative cross-validation technique in comparison to none.

2. Survival Bump Hunting for Exploratory Survival Analysis

2.1 Bump Hunting Model

2.1.1 Notations - Goal

The formal setup of bump hunting is as follows [see also 10, 24]. Let us consider a supervised problem with a univariate output (response) random variable, denoted $\mathbf{y} \in \mathbb{R}$. Further, let us consider a p-dimensional random vector $\mathbf{X} \in \mathbb{R}^p$ of support S, also called input space, in an Euclidean space. Let us denote the p input variables by $\mathbf{X} = [\mathbf{x}_j]_{j=1}^p$, of joint probability density function $p(\mathbf{X})$, and by $f(\mathbf{x}) = E(\mathbf{y}|\mathbf{X} = \mathbf{x})$ the target function to be optimized (e.g. any regression function or e.g. the p.m.f or p.d.f $f_{\mathbf{X}}(\mathbf{x})$).

Briefly, the goal in bump hunting is to find a sub-space or region R (not necessarily contiguous) of the input space ($R \subseteq S$) within which the average value $\bar{f}_R$ of $f(\mathbf{x})$ is expected to be significantly larger (or smaller) than its average value $\bar{f}_S$ over the entire input space. In addition, one wishes that the corresponding support (mass) of R, say β_{0R}, be not too small, that is, greater than a minimal support threshold, say $0 < \beta_0 < 1$. Formally, in the continuous case of $\mathbf{X}$: $\bar{f}_R = \frac{\int_{\mathbf{x}\in R} f(\mathbf{x})p(\mathbf{x})d\mathbf{x}}{\int_{\mathbf{x}\in R} p(\mathbf{x})d\mathbf{x}} \gg \bar{f}_S$ and $\beta_R = \int_{\mathbf{x}\in R} p(\mathbf{x})d\mathbf{x} \gg \beta_0$.

Let S_j be the support of the jth variable $\mathbf{x}_j$, such that the input space can be written as the (Cartesian) outer product space $S = \times_{j=1}^p S_j$. Let $s_j \subseteq S_j$ denotes the unknown subset of values of variable $\mathbf{x}_j$ corresponding to the unknown support of the target region R. Let $J \subseteq \{1, \ldots, p\}$ be the subset of indices of selected variables in the process. The goal in bump hunting amounts to finding the value-subsets $\{s_j\}_{j\in J}$ of the corresponding variables $\{\mathbf{x}_j\}_{j\in J}$ such that $R = \{\mathbf{x} \in \bigcap_{j\in J} (\mathbf{x}_j \in s_j) : (\bar{f}_R \gg \bar{f}_S)(\beta_R \gg \beta_0)\}$.

2.1.2 Estimates

Since the underlying distribution is not known, the estimates of $\bar{f}_R$ and β_R must be used. Assume a supervised setting, where the outcome response variable is $\mathbf{y} = (y_1 \ldots y_n)^T$ and the explana-

tory/input variables are $\mathbf{X} = (\mathbf{x}_1 \dots \mathbf{x}_n)^T$, where each observation is the p-dimensional vector of covariates $\mathbf{x}_i = [x_{i,1} \dots x_{i,p}]^T$, for $i \in \{1, \dots, n\}$. Plug-in estimates of the conditional expectation $\bar{f}_R$ of the target function $f_{\mathbf{X}}(\mathbf{x})$ and of the support β_R of the region R are respectively derived as: $\hat{\bar{f}}_R = \frac{1}{n\hat{\beta}_R} \sum_{\mathbf{x}_i \in \hat{R}} y_i = \frac{1}{n\hat{\beta}_R} \sum_{i=1}^{n} y_i I(\mathbf{x}_i \in \hat{R})$ and $\hat{\beta}_R = \frac{1}{n} \sum_{\mathbf{x}_i \in \hat{R}} I(\mathbf{x}_i \in \hat{R}) = \frac{1}{n} \sum_{i=1}^{n} I(\mathbf{x}_i \in \hat{R})$.

2.1.3 Remarks

1. Note that the goal amounts to comparing the conditional expectation of the response over the target region R: $\bar{f}_R = E[f(\mathbf{x})|\mathbf{x} \in R]$ with the unconditional one $\bar{f}_S = E[f(\mathbf{x})]$.
2. In the bump hunting objective stated in section 2.1.1, note that larger target function average $\bar{f}_R$ is associated with smaller support β_R of the region R. So, in practice, one is to use a coverage so as to trade-off between maximizing $\bar{f}_R$ and maximizing β_R.
3. If the target function to be optimized is for instance the p.m.f or p.d.f $f_{\mathbf{X}}(\mathbf{x})$, then $\Pr(\mathbf{X} \in R)$ is the probability mass/density of a local maximum and the task is equivalent to a mode(s) hunting.
4. In the case of real-valued inputs, the entire input space is the p-dimensional outer product space $S \subseteq \mathbb{R}^p$; the support S_j of each individual input variable (and of each corresponding value-subset s_j) is the usual interval of the form $S_j = \left[t_j^-, t_j^+\right] \subset \mathbb{R}$ for $j = 1, \dots, p$; the target region R has the shape of a (possibly contiguous) $|J|$-dimensional hyper-rectangle in $\mathbb{R}^{|J|}$, called a *box*, which can be written as the outer product of the form $B = \times_{j \in J}[t_j^-, t_j^+]$.

2.1.4 Estimation by the Patient Rule Induction Method (PRIM)

Let the data be $\{\mathbf{x}_i, y_i\}_{i=1}^n$. The Patient Rule Induction Method (PRIM) is used to get the region estimate $\hat{R}$ and the corresponding output response mean estimate $\hat{\bar{f}}_R$. Essentially, the method is one of recursive peeling/pasting algorithm that explores the input space target region, where the response is expected to be larger on average. The method generates a sequence of boxes that collectively cover the solution region R. The way the space is covered and the box induction is done as well as how the patience and stopping rules are controlled is detailed in the original paper of Friedman & Fisher [10], later formalized by Polonik & Wang [24].

Briefly, a sequence of boxes $\{B_m\}_{m=1}^M$ is generated from the data $\{\mathbf{x}_i, y_i\}_{i=1}^n$ to collectively cover the target region R. To induce the box B_m at the mth iteration ($m > 1$), the top-down peeling algorithm generates a subsequence of nested sub-boxes $\{B_{m,l}\}_{l=1}^L$ starting from an initial box $B_{m,1}$ that covers all the data remaining at the mth iteration of the covering loop. At the lth iteration, a sub-box is peeled off from within the current sub-box $B_{m,l}$ to produce the next smaller sub-box $B_{m,l+1}$. The current sub-box $B_{m,l}$ is then updated: $B_{m,l+1} \leftarrow B_{m,l}$ and the peeling procedure is looped until some stopping rule is met. Eventually, the solution region R is described by logical statements or decision rules of the input space involving the value-subset of each selected input variable.

2.2 Recursive Peeling Methods for Survival Bump Hunting

Assume a supervised problem, where the function of interest is a univariate survival/risk response variable (possibly censored) in a multivariate setting of real-valued (continuous or discrete) inputs variables $\mathbf{X} = [\mathbf{x}_j]_{j=1}^p$. The goal is to characterize an extreme-survival-response support in the predictor space and identify the corresponding box-defined group of samples using a recursive peeling method derived from the Patient Rule Induction Method (PRIM).

2.2.1 Survival Model Notations

We focus on a univariate right-censored survival outcome under the assumptions of independent observations, non-competitive risks, and (type I or II) random or non-informative censoring. Because the response variable is subject to censoring, we use the general random censoring

model. Denote the *true* survival time by the random variable U and the *true* censoring time by the random variable C, then the *observed* survival time is the random variable $T = \min(U, C)$. Also, under our assumptions, C is assumed independent of U conditionally on $\mathbf{X}$. Let the observed event (non-censoring) random variable indicator be $\Delta = I(U \leqslant C)$. Using previous notations (2.1.2), for each observation $i \in \{1, \ldots, n\}$, the individual true survival time, observed censoring time, observed survival time and observed indicator event variable are the realizations denoted by u_i, c_i, $t_i = \min(u_i, c_i)$ and $\delta_i = I(u_i \leqslant c_i)$, respectively, so that the observed data consists of $(t_i, \delta_i, \mathbf{x}_i)_{i=1}^n$, where $\mathbf{x}_i = [x_{i,1} \ldots x_{i,p}]^T$, for $i \in \{1, \ldots, n\}$. Here, the outcome response variable is denoted $\mathbf{t} = [t_1 \ldots t_n]^T$.

Let $S(t) = \Pr(T \geqslant t)$ be the probability that an observation from the population of interest will have an observed time-to-event T free of the event until time t. The non-parametric Kaplan-Meier estimator was used to estimate the survival probability function $S(t)$ of time-to-event in each box-defined subgroup, whether it was observed or not. We used the log-rank test to assess statistical significance of difference between survival distributions of each box-defined subgroups. Let the hazard function and the cumulative hazard function be denoted by $\lambda(t)$ and $\Lambda(t)$, respectively, where $\lambda(t) = \frac{d\Lambda(t)}{dt} = -\frac{d\log(S(t))}{dt}$. As is commonly done, the hazard rate may be estimated by the maximum likelihood estimator (MLE) $\hat{\lambda}_{ML}$ of the simple exponential hazard rate, or by regressing the individual hazard rate of an observation $i \in \{1, \ldots, n\}$ on the covariate vector $\mathbf{x}_i^T$ in a Cox Proportional Hazards (CPH) regression model: $\lambda(t|\mathbf{x}_i) = \lambda_0(t)\exp(\boldsymbol{\beta}^T\mathbf{x}_i)$ where $\boldsymbol{\beta} = [\beta_1 \ldots \beta_p]^T$ is the p-dimensional vector of regression coefficients [5],

$$\hat{\lambda}_{ML} = \frac{\sum_{i=1}^n \delta_i}{\sum_{i=1}^n t_i} \tag{1}$$

$$\hat{\lambda}_{CPH} = \sum_{i=1}^n \log \frac{\lambda(t|\mathbf{x}_i)}{\lambda_0(t)} = \sum_{i=1}^n \left(\sum_{j=1}^p \beta_j x_{i,j} \right). \tag{2}$$

In addition to the above assumption on the censoring mechanism, by definition the MLE assumes exponential distribution of survival time $(t_i)_{i=1}^n$ and the Cox-PH model assumes proportional hazards.

2.2.2 Survival-Specific Peeling Criteria

As mentioned earlier, rule-induction methods such as decision tree-based methods have proven to be useful to estimate relative risk in groups in the context of a time-to-event outcome. Here, we describe several survival-specific peeling criteria for fitting decision boxes to non-informative censored survival times, most of which are borrowed from the survival splitting rules used to grow regression survival decision trees [1, 16, 17, 25, 30] or from their ensemble versions [15]. Specifically, survival-specific peeling criteria/rules are to be used to decide which variable will be selected to give the best peel between two sub-boxes from two consecutive generations (parent-child descendance) of the box induction/peeling loop in a recursive peeling algorithm (see next section 2.2.4).

To account for censoring we simply supervise by proxy for extreme time-to-event outcome, turning the censored outcome $\mathbf{t}$ into an uncensored "surrogate" outcome y. Using previous notations (see section 2.1.4), the focus is on selecting a sub-box $b_{m,l}$ from within the current sub-box $B_{m,l}$ to be peeled off along one of its faces (i.e. direction of peeling := dimension j) to induce the next smaller sub-box $B_{m,l+1}$ of the box induction/peeling sequence. This is done by maximizing the "surrogate" outcome rate of increase between two consecutive generations of sub-boxes $B_{m,l}$ and $B_{m,l+1}$. Denote by $y(m, l)$ the box "surrogate" outcome at sub-step or generation (m, l) of the box induction/peeling sequence (see Algorithm 1). The rate of increase

in $y(m,l)$ between two consecutive generations of sub-boxes $B_{m,l}$ and $B_{m,l+1}$ is defined as:

$$\eta(m,l) = \frac{y(m,l+1) - y(m,l)}{\beta_{m,l} - \beta_{m,l+1}} \tag{3}$$

The use of hazard rates was originally proposed by LeBlanc et al. [18, 19]. Since they are always estimable, we can use them to maximize the hazards ratio or relative risk of the observations inside the sub-box $B_{m,l}$ compared to the observations inside the sub-box $B_{m,l+1}$ of the next generation. Let $\gamma_i(B) = I(\mathbf{x}_i \in B)$ be the box B membership indicator for each observation $i \in \{1,\ldots,n\}$, and let $\boldsymbol{\gamma}(B) = [\gamma_1(B)\ldots\gamma_n(B)]^T$ be the corresponding indicator n-vector. If the MLE of the simple exponential hazard rate is used (1), then one considers $\hat{\lambda}_{ML}$ for the elements in box B. Likewise, if the Cox-PH estimate of the hazard rate is used (2), then one considers $\hat{\lambda}_{CPH}$ for the elements in box B by letting the indicator n-vector $\boldsymbol{\gamma}^T(B)$ be the only covariate in the CPH model, so that $\boldsymbol{\beta} = \beta_1$, $\mathbf{x}_i = x_{i,1} = \gamma_i(B)$, $\hat{\lambda}_{ML}(B) = \frac{\sum_{i=1}^n \delta_i \gamma_i(B)}{\sum_{i=1}^n t_i \gamma_i(B)}$ and $\hat{\lambda}_{CPH}(B) = \beta_1 \sum_{i=1}^n \gamma_i(B)$, which leads to the derivation of the estimated relative risks:

$$\begin{cases} \hat{\eta}_{ML}(m,l) & = \frac{\hat{\lambda}_{ML}(B_{m,l+1}) - \hat{\lambda}_{ML}(B_{m,l})}{\hat{\beta}_{m,l} - \hat{\beta}_{m,l+1}} \\ & = \frac{1}{\hat{\beta}_{m,l} - \hat{\beta}_{m,l+1}} \left(\frac{\sum_{i=1}^n \delta_i \gamma_i(B_{m,l+1})}{\sum_{i=1}^n t_i \gamma_i(B_{m,l+1})} - \frac{\sum_{i=1}^n \delta_i \gamma_i(B_{m,l})}{\sum_{i=1}^n t_i \gamma_i(B_{m,l})} \right) \\ \hat{\eta}_{CPH}(m,l) & = \frac{\hat{\lambda}_{CPH}(B_{m,l+1}) - \hat{\lambda}_{CPH}(B_{m,l})}{\hat{\beta}_{m,l} - \hat{\beta}_{m,l+1}} \\ & = \beta_1 \frac{\sum_{i=1}^n \gamma_i(B_{m,l+1}) - \sum_{i=1}^n \gamma_i(B_{m,l})}{\hat{\beta}_{m,l} - \hat{\beta}_{m,l+1}} \end{cases} \tag{4}$$

Finally, the particular sub-box $\hat{b}^*_{m,l}$ that is chosen to yield the largest box increase rate $\hat{\eta}(m,l)$ of relative risk between sub-box $\hat{B}_{m,l}$ and the next one $\hat{B}_{m,l+1} = \hat{B}_{m,l} \backslash \hat{b}^*_{m,l}$ is such that $\hat{b}^*_{m,l} = \text{argmax}_{\hat{b}_{m,l} \in C(b)} [\hat{\eta}(m,l)]$, where $C(b)$ represents the class of potential sub-boxes $\hat{b}_{m,l}$ eligible for removal at sub-step or generation (m,l).

For the record, alternative survival-specific peeling criteria may be used as well to maximize the "surrogate" outcome rate of increase between two consecutive sub-boxes $\hat{B}_{m,l}$ and $\hat{B}_{m,l+1}$ of the box induction/peeling loop. Currently, our implementation of Survival Bump Hunting in our R package `PrimSRC` [7] uses the following three criteria:

1. The Log Hazards Ratio or Relative Risk statistic $\hat{\eta}(m,l)$ can be used to maximize the difference of hazards between observations inside two consecutive sub-boxes $\hat{B}_{m,l}$ and $\hat{B}_{m,l+1}$ of the box induction/peeling sequence (see also [18, 19])
2. The two-sample Log-Rank Test statistic $\hat{\zeta}(m,l) = \sum_{i=1}^n \gamma_i(m,l) \left(\delta_i - \hat{\Lambda}_0(t_i) \right)$ can be used to maximize the difference of survival distributions between observations inside two consecutive sub-boxes $\hat{B}_{m,l}$ and $\hat{B}_{m,l+1}$ of the box induction/peeling sequence. An approximate log-rank test introduced by LeBlanc and Crowley can also be used to greatly reduce computations [17]
3. The Concordance Error Rate $1 - C$, where C is Harrel's rank correlation U-statistic or concordance index [13], to maximize the probability of concordance between predicted and observed survival, i.e. to minimize the prediction error estimate

2.2.3 Survival End Points Statistics

The first two end-points statistics defined for each sub-step or generation (m,l) in Survival Bump Hunting are: (i) The Event-Free Probability $P_0(m,l)$ or probability of non-event until a certain time $T(m,l)$ in the highest-risk group/box (For instance: the Probability of Event-Free Survival (PEFS), or the Survival Rate that indicates the probability to be alive for a given period of

time after diagnosis); and (ii) The Event-Free Time $T_0(m,l)$ or time to reach a certain end-point probability $P(m,l)$ in the highest-risk group/box (Frequently themedianis used so that the end-point can be calculated once 50% of subjects have reached the end-point. For instance: the Median-Survival-Probability Time, also known as the Median Survival (MS), indicates the period of time (survival duration) once 50% of subjects have reached survival).

However, often times these statistics are not observable: the survival/risk probability in a group may be large enough that $P_0(m,l)$ may not be reached for a specified time $T(m,l)$. Similarly, $T_0(m,l)$ may not always be reached for a certain probability $P(m,l)$. In any of these cases, we determine the limit end-points $P'_0(m,l)$ and $T'_0(m,l)$, which are always observable and computable for each sub-step or generation (m,l). These, along with the subsequent end-points, are the cross-validated statistics that are implemented in our R package `PrimSRC` [7]:

1. Minimal Event-Free Probability (*MEFP*) $P'_0(m,l)$ and corresponding max. time $T'(m,l)$
2. Maximal Event-Free Time (*MEFP*) $T'_0(m,l)$ and corresponding min. probability $P'(m,l)$
3. Log Hazards Ratios (*LHR*) $\lambda(m,l)$ between the highest-risk group/box and lower-risk groups/boxes of the same generation (see Algorithm 1)
4. Log-Rank Test statistic (*LRT*) $z(m,l)$ between the highest-risk group/box and lower-risk groups/boxes of the same generation (see Algorithm 1)
5. Prediction Error (*PE*) $pe(m,l)$ using the Concordance Error Rate $1-C$, where C is Harrel's Concordance Index [13] (see also section 2.2.2)

2.2.4 Estimation by a Patient Recursive Survival Peeling Algorithm

The strategy employed here is one of recursive peeling algorithm for survival bump hunting that we derived from the PRIM algorithm. Our "Patient Recursive Survival Peeling" algorithm (annotated below w.l.o.g for a maximization problem) proceeds similarly as in PRIM except for the Box Induction peeling/pasting criteria and Induction Stopping Rule (see section 2.1.4):

Algorithm 1 Patient Recursive Survival Peeling.

- Start with the training data $\mathcal{L}_{(1)}$ and a maximal box B_1 containing it
- For $m \in \{1,\ldots,M\}$:
  1: Generate a box B_m using the remaining training data $\mathcal{L}_{(m)}$
  2: For $l \in \{1,\ldots,L\}$:
    - Top-down peeling: Generate a box $B_{m,l}$ by conducting a stepwise variable selection/usage: shrink the box by compressing one face (peeling), so as to peel off a quantile α_0 of observations of a variable $\mathbf{x}_j$ for $j \in \{1,\ldots,p\}$. Choose the direction of peeling j that yields the largest box increase rate $\hat{\eta}(m,l)$ of hazards ratio or relative risk between sub-box $\hat{B}_{m,l}$ and the next one $B_{m,l+1}$. The current sub-box $\hat{B}_{m,l}$ is then updated: $\hat{B}_{m,l+1} = \hat{B}_{m,l} \backslash \hat{b}^*_{m,l}$, where $\hat{b}^*_{m,l} = \text{argmax}_{b_{m,l} \in C(b)} [\hat{\eta}(m,l)]$
    - Bottom-up pasting: Expand the box along any face (pasting) as long as the resulting box increase rate $\hat{\eta}(m,l) > 0$
    - Stop the peeling looped until a minimal box support $\hat{\beta}_{m,L}$ of $\hat{B}_{m,L}$ is such that it reached a minimal box support $0 \leqslant \beta_0 \leqslant 1$, expressed as a fraction of the data: $\hat{\beta}_{m,L} \leqslant \beta_0$
    - $l \leftarrow l+1$
  3: Step #2 give a sequence of nested boxes $\{\hat{B}_{m,l}\}_{l=1}^{L}$, where L is the estimated number of peeling/pasting steps with different numbers of observations in each box. Call the next box $\hat{B}_{m+1} = \hat{B}_{m,L}$. Remove the data in box $\hat{B}_m$ from the training data: $\mathcal{L}_{(m+1)} = \mathcal{L}_{(m)} \backslash \hat{B}_m$
  4: Stop the covering loop when running out of data or when a minimal number of observations remains within the last box $\hat{B}_M$, say $\hat{\beta}_M \leqslant \beta_0$
  5: $m \leftarrow m+1$
- Steps #1 – #5 produce a sequence of (not necessarily nested) boxes $\{\hat{B}_m\}_{m=1}^{M}$, where M is the estimated total number of boxes covering $\mathcal{L}_{(1)}$
- Collect the decision rules of all boxes $\{\hat{B}_m\}_{m=1}^{M}$ into a simple final decision rule $\hat{\mathcal{R}}$ of the solution region $\hat{R}$ of the form: $\hat{\mathcal{R}} = \bigcup_{m=1}^{M} \hat{\mathcal{R}}_m$, where $\hat{\mathcal{R}}_m = \bigcap_{j \in J} (\mathbf{x}_j \in [t^-_{j,m}, t^+_{j,m}])$ giving a full description of the estimated bumps in the entire input space

3. Cross-Validation for Recursive Peeling Methods

3.1 K-fold Cross-Validation for Recursive Peeling Methods

3.1.1 Setup

Cross-validation of box estimates should include all steps of the box generation sequence $\{B_m\}_{m=1}^{M}$ i.e. for the (outer) coverage loop of our "Patient Recursive Survival Peeling" algorithm (1), each step of which involves a peeling sequence $\{B_{m,l}\}_{l=1}^{L}$ of the (inner) box peeling/induction loop. For simplicity, cross-validation designs of box estimates $\{B_{m,l}\}_{l=1}^{L}$ and of resulting decision rule $\hat{\mathcal{R}}_m$ are shown for *fixed* $m \in \{1, \ldots, M\}$, so that subscript m is further dropped. Without loos of generality, fix $m = 1$ (first coverage box).

3.1.2 Estimated Box Quantities of Interest

Using previous notations and assuming m fixed ($m = 1$), if we let $\hat{B}_l$ be the lth trained box and $\hat{\beta}_l$ be its estimated box support for $l \in \{1, \ldots, L\}$ of the box peeling sequence $\{\hat{B}_l\}_{l=1}^{L}$, then useful box quantities of interest are (for the lth peeling step $l \in \{1, \ldots, L\}$):

- Highest-risk box definition: box $2p$ edges $\left[\hat{t}_{j,l}^{-}, \hat{t}_{j,l}^{+}\right]_{j=1}^{p}$, box support $\hat{\beta}_l$, and box membership indicator $\hat{\boldsymbol{\gamma}}(B_l)$
- Traces of Variable Usage (VU) $\widehat{vu}_l$ and Variable Importance (VI) $\widehat{vi}_l$
- Profiles of Log Hazard Ratios (LHR) $\hat{\lambda}_l$, Log-Rank Tests (LRT) $\hat{z}_l$, Minimal Event-Free Probability ($MEFP$) $\hat{P}'_{0,l}$ and Maximal Event-Free Time ($MEFT$) $\hat{T}'_{0,l}$
- Kaplan-Meir curves of survival probability values with p-values $\hat{p}_l$ (see section 3.3)
- Prediction Error (PE) $\widehat{pe}_l$

3.1.3 Resampling Design

Although using a fully independent test set in evaluating a predictive bump hunting model is always advisable, the sample size n in discovery-based studies is often too small to effectively split the data into training and testing sets and provide accurate estimates [3, 8, 26]. In such cases, re-sampling techniques mentioned in the introduction such as K-fold Cross-Validation are required [2, 23]. Similarly as in Split Sample Validation, the re-use of training data is proper if performed using data resampling methods that iteratively partition the data to hold out data subsets that are not used for model building [23].

In resampling based on full K-fold cross-validation, the whole data $\mathcal{L}$ is randomly partitioned into K approximately equal parts of test samples or test subsets $(\mathcal{L}_1, \ldots, \mathcal{L}_k, \ldots, \mathcal{L}_K)$. For each test subset $(\mathcal{L}_k, (k \in \{1, \ldots, K\})$, a training set $\mathcal{L}_{(k)}$ is formed from the union of the remaining $K - 1$ subsets: $\mathcal{L}_{(k)} = \mathcal{L} \setminus \mathcal{L}_k$. The process is repeated K times, so that K test subsets $\mathcal{L}_k$ are formed of about equal size and K corresponding training subsets $\mathcal{L}_{(k)}$, for $k \in \{1, \ldots, K\}$. Typically, $K \in \{3, \ldots, 10\}$. The training samples are approximately of size $\approx n(K-1)/K$ and the test samples are of size $n^t \approx n/K$.

3.1.4 Nested K-fold Cross-Validation

Here, assuming m fixed ($m = 1$) and using previous notations, a specific *nested* K-fold Cross-Validation (CV) strategy for training a cross-validated box peeling sequence $\{\hat{B}_l\}_{l=1}^{L}$ in a predictive bump hunting model is as follows:

- *Overall-CV*: A peeling model of a certain fixed length $\hat{L}$ (estimated by *Internal-CV*) and a resulting training decision rule, abbreviated $\hat{\mathcal{R}}_k$, are generated from each training subset $\mathcal{L}_{(k)}$, leaving out the test subset $\mathcal{L}_k$ during all aspects of model building including variable selection and calibration (see step #2 of Algorithm 1).
- *Internal-CV*: To estimate the optimal number of peeling steps/boxes $\hat{L}$ from the training subset $\mathcal{L}_{(k)}$, one uses one of the cross-validated end-points statistics described in 2.2.3 as a criterion or measure of performance taking censoring into account.

The trained model is in turn used to generate cross-validated box estimates from which cross-validated survival estimates as well as cross-validated predictions are made in the left out test subset $\mathcal{L}_k$. This process is repeated K times, for $k \in \{1, \ldots, K\}$, repeating model building for each models $\{\hat{\mathcal{R}}_k\}_{k=1}^{k=K}$. After K rounds of training and testing are complete, all the test set predictions are used to estimate the accuracy. The CV error is then given by the average of the prediction errors computed from the K models $\{\hat{\mathcal{R}}_k\}_{k=1}^{k=K}$ generated from each loop of the cross-validation. The prediction or classification error is estimated from the discrepancies between the true and predictive classifications of the n independent observations.

3.1.5 K-fold Cross-Validation Techniques

There are remaining issues to deal with K-fold CV: how to cross-validate a simple box peeling sequence $\{\hat{B}_l\}_{l=1}^{L}$ and related statistics is not straightforward, and how to cross-validate survival curve estimates and related statistics is also not intuitive (see also [27]). So, regular K-fold cross-validation is not directly applicable to the tasks of generating a simple box decision rule and estimating survival curves. Therefore, one must design a specific CV technique that is amenable to the dual task of bump hunting by recursive peeling and survival analysis together. In addition, in regular averaged K-fold CV, all final test subset mean estimates of interest (including survival estimates) are computed on test samples of size $n^t \approx n/K$, which could be a problem in case of small sample size n.

Hence, we propose a technique by which our overall K-fold cross-validation estimates can be computed by the so-called *combining* technique. In it, all the test sets are first collected from all cross-validation loops and the CV estimates are computed *once* on this combined test set to give the final "Combined Cross-Validation" estimates (see section below 3.2). In this article, this strategy was compared with the situation where no cross-validation was done at all (see result section 4.2).

Finally, cross-validation estimates are known to be quite variable. To address this issue, K-fold cross-validation must somehow account for this by averaging over some replications. This technique is further detailed in the Replicated Cross-Validation section below (3.4).

3.2 K-fold Combined Cross-Validation for Recursive Peeling Methods

For each loop, samples from the training subset are used to train a peeling model of a certain cross-validated length, then samples from the test subset are used to generate estimates. At the end, all test samples in the box are combined together and all test samples outside the box are combined together as well. This allows the estimation box quantities and of survival distribution curves for both groups. In addition, all final statistical quantities of interest (including survival estimates) are now computed on samples of full size n instead of n/K, as it would be in regular averaged K-fold CV.

Using previous notations and assuming m fixed ($m = 1$), we let $\hat{B}_{l(k)}$ and $\hat{\beta}_{l(k)}$ be respectively the $l(k)$th trained box and its estimated box support from the box peeling sequence $\{\hat{B}_{l(k)}\}_{l=1}^{L(k)}$, where $l(k)$ indicates the lth peeling step in the kth trajectory for $l \in \{1, \ldots, L(k)\}$ and $k \in \{1, \ldots, K\}$. In K-fold Combined CV, estimates of a box quantity of interest q are indexed by the resulting combined test boxes, and are computed *once* on the combined K test subsets collected from all cross-validation loops.

Below, one considers K peeling trajectories of this kind, each of length $L(k)$, from the kth training subset $\mathcal{L}_{(k)}$, for $k \in \{1, \ldots, K\}$. Denote by $\hat{L}^{cv} = \min_{k \in \{1,\ldots,K\}} [L(k)]$ the minimum length of the peeling trajectories over the K loops. A trained box $\hat{B}_{l(k)}$ of support $\hat{\beta}_{l(k)}$ is constructed from each training subset $\mathcal{L}_{(k)}$. Each test subset $\mathcal{L}_k$ is used to estimate the test box $\hat{B}^t_{l(k)}$ and its membership indicator $\hat{\gamma}^t_{l(k)}$ from the model grown on the kth training subset $\mathcal{L}_{(k)}$. The lth *combined* CV test box $\hat{B}^{cv}(l)$ and its membership indicator $\hat{\gamma}^{cv}(l)$ are formed over the K

cross-validation loops by union of the K boxes $\{\hat{B}^t_{l(k)}\}_{k=1}^K$ and logical disjunction of the K box membership indicators $\{\hat{\boldsymbol{\gamma}}^t_{l(k)}\}_{k=1}^K$, respectively. Here, the *combined* CV trajectory curve $\hat{q}^t_k(x)$ of length $\hat{L}^{cv}$ is defined as the piecewise constant curve, evaluated at the lth combined test box $\hat{B}^{cv}(l)$ or its membership indicator $\hat{\boldsymbol{\gamma}}^{cv}(l)$.

	First peeling step in the kth trajectory			direction of peeling $\longrightarrow$		Last peeling step in the kth trajectory
For the kth training subset:						
kth training trajectory and estimates $\longrightarrow$	$1(k)$		$l(k)$			$L(k)$
	$\boxed{\hat{B}_{1(k)}}$	$\cdots$	$\boxed{\hat{B}_{l(k)}}$	$\cdots$		$\boxed{\hat{B}_{L(k)}}$
	$\hat{\beta}_{1(k)}$	$\cdots$	$\hat{\beta}_{l(k)}$	$\cdots$		$\hat{\beta}_{L(k)}$
kth test boxes and membership indicators $\longrightarrow$	$\hat{B}^t_{1(k)}$	$\cdots$	$\hat{B}^t_{l(k)}$	$\cdots$		$\hat{B}^t_{L(k)}$
	$\hat{\boldsymbol{\gamma}}^{t\ T}_{1(k)}$	$\cdots$	$\hat{\boldsymbol{\gamma}}^{t\ T}_{l(k)}$	$\cdots$		$\hat{\boldsymbol{\gamma}}^{t\ T}_{L(k)}$
	$=\left[I[\mathbf{x}_i \in \hat{B}^t_{1(k)}]\right]_{i=1}^n$		$=\left[I[\mathbf{x}_i \in \hat{B}^t_{l(k)}]\right]_{i=1}^n$			$=\left[I[\mathbf{x}_i \in \hat{B}^t_{L(k)}]\right]_{i=1}^n$
Union of CV test boxes over K trajectories $\longrightarrow$	$\hat{B}^{cv}(1)=\bigcup_{k=1}^K \hat{B}^t_{1(k)}$	$\cdots$	$\hat{B}^{cv}(l)=\bigcup_{k=1}^K \hat{B}^t_{l(k)}$	$\cdots$	$\hat{B}^{cv}(\hat{L}^{cv})=\bigcup_{k=1}^K \hat{B}^t_{\hat{L}^{cv}}$	
Disjunction of CV test box membership indicators over K trajectories $\longrightarrow$	$\hat{\boldsymbol{\gamma}}^{cv}(1)=\bigvee_{k=1}^K \hat{\boldsymbol{\gamma}}^t_{1(k)}$	$\cdots$	$\hat{\boldsymbol{\gamma}}^{cv}(l)=\bigvee_{k=1}^K \hat{\boldsymbol{\gamma}}^t_{l(k)}$	$\cdots$	$\hat{\boldsymbol{\gamma}}^{cv}(\hat{L}^{cv})=\bigvee_{k=1}^K \hat{\boldsymbol{\gamma}}^t_{\hat{L}^{cv}}$	
Combined CV test box quantities over K trajectories $\longrightarrow$	$\hat{q}^{cv}(1)$	$\cdots$	$\hat{q}^{cv}(l)$	$\cdots$		$\hat{q}^{cv}(\hat{L}^{cv})$
	$=\hat{q}\left[\hat{B}^{cv}(1)\right]$		$=\hat{q}\left[\hat{B}^{cv}(l)\right]$			$=\hat{q}\left[\hat{B}^{cv}(\hat{L}^{cv})\right]$

Wherefrom one derives "Combined CV" estimates over the K test trajectories, for each step $l \in \{1, \ldots, \hat{L}^{cv}\}$:

- The "Combined CV" estimate of the box definition ($2p$ edges $\left[\hat{t}^-_{j,l}, \hat{t}^+_{j,l}\right]_{j=1}^p$): $\hat{B}^{cv}(l) = \bigcup_{k=1}^K \hat{B}^t_{l(k)}$
- The "Combined CV" estimate of the box membership indicator (Boolean n-vector): $\hat{\boldsymbol{\gamma}}^{cv\ T}(l) = [\hat{\gamma}^{cv}_i(l)]_{i=1}^n = \bigvee_{k=1}^K \hat{\boldsymbol{\gamma}}^t_{l(k)} = \bigvee_{k=1}^K \left[I[\mathbf{x}_i \in \hat{B}^t_{l(k)}]\right]_{i=1}^n = \left[I[\mathbf{x}_i \in \hat{B}^{cv}(l)]\right]_{i=1}^n$
- The "Combined CV" estimates of box quantities of interest, each taken as the combined CV trajectory curve evaluated at the combined CV test box $\hat{B}^{cv}(l)$: $\hat{q}^{cv}(l) = \hat{q}\left[\hat{B}^{cv}(l)\right]$. The latter is done for the "Combined CV" box estimates of: (i) The box support: $\hat{\beta}^{cv}(l) = \hat{\beta}\left[\hat{B}^{cv}(l)\right]$; (ii) The Log Hazard Ratio (LHR) in the high-risk box: $\hat{\lambda}^{cv}(l) = \hat{\lambda}\left[\hat{B}^{cv}(l)\right]$; (iii) The Log-rank Test (LRT) between the high vs. low-risk box: $\hat{z}^{cv}(l) = \hat{z}\left[\hat{B}^{cv}(l)\right]$; (iv) The Minimal Event-Free Probability ($MEFP$): $\hat{P}'^{\ cv}_0(l) = \hat{P}'_0\left[\hat{B}^{cv}(l)\right]$; (v) The Minimal Event-Free Time ($MEFT$): $\hat{T}'^{\ cv}_0(l) = \hat{T}'_0\left[\hat{B}^{cv}(l)\right]$.

3.3 K-fold Cross-Validation of P-Values

In order to evaluate the statistical significance of spread among the cross-validated survival curves, the log-rank test statistic is a classical measure. However, null distribution of the log-rank test (χ^2_1 for a two group comparison) is not valid because the observations used to cross-validate the curves are not independent anymore.

for each step $l \in \{1, \ldots, \hat{L}^{cv}\}$, we generate the null distribution of the cross-validated log-rank statistic $\hat{z}^{cv(a)}(l)$ for $a \in \{1, \ldots, A\}$ by randomly permuting the correspondence of survival times and censoring indicators to the data and computing the cross-validated survival curves and the cross-validated log-rank statistic for that permutation. By repeating A times the entire K-fold cross-validation process for many random permutations (typically $A = 1000$), one generates a null distribution of the cross-validated log-rank statistics $\{\hat{z}^{cv(a)}(l)\}_{a=1}^{A}$.

The proportion of replicates with log-rank statistic greater than or equal to the observed statistic $\hat{z}^{cv}(l)$ for the un-permuted data is the statistical significance level for the test. Cross-validated permutation test p-values are then calculated for each step $l \in \{1, \ldots, \hat{L}^{cv}\}$ as: $\hat{p}^{cv}(l) = \frac{1}{A}\sum_{a=1}^{A} I\left[\hat{z}^{cv(a)}(l) \geqslant \hat{z}^{cv}(l)\right]$. These p-values may be discrete: the precision depends on the number A of random permutations.

3.4 Replicated K-fold Cross-Validation for Recursive Peeling Methods

To account for the high variability of cross-validated estimates, K-fold cross-validation is repeated (typically $B = 10 - 100$ times) and results averaged over the Monte-Carlo replications. We call these estimates "Replicated Combined CV". Denote by superscript rcv a replicated cross-validated estimate, and by superscript b each Monte-Carlo replicate, for $b \in \{1, \ldots, B\}$:

- The "Replicated Combined CV" box peeling sequence length is taken as the floored-mean of the box peeling sequence lengths obtained from the B replicates:

$$\bar{L}^{rcv} = \left\lfloor \frac{1}{B}\sum_{b=1}^{B} \hat{L}^{cv(b)} \right\rfloor \tag{5}$$

- The "Replicated Combined CV" estimate of the box definition ($2p$ edges $\left[\hat{t}^{-}_{j,l}, \hat{t}^{+}_{j,l}\right]_{j=1}^{p}$):

$$\bar{B}^{rcv}(l) = \underset{b\in\{1,\ldots,B\}}{\text{ave}} \left[\hat{B}^{cv(b)}(l)\right] \tag{6}$$

- The "Replicated Combined CV" estimate of the box membership indicator (Boolean n-vector, observed to be approximately equal to the point-wise majority vote over the B replicates):

$$\bar{\boldsymbol{\gamma}}^{rcv}(l) = \left[I[\mathbf{x}_i \in \hat{B}^{rcv}(l)]\right]_{i=1}^{n} \approx \left[I\left(\sum_{b=1}^{B} \hat{\gamma}_i^{cv(b)}(l) \geqslant \left\lceil \frac{B}{2} \right\rceil\right)\right]_{i=1}^{n} \tag{7}$$

- The "Replicated Combined CV" estimates of the quantities of interest, each taken as the average estimate over the B replicates:

$$\bar{q}^{rcv}(l) = \frac{1}{B}\sum_{b=1}^{B} \hat{q}^{cv(b)}(l) \tag{8}$$

  The latter is done for the "Replicated Combined/Averaged CV" estimates of: (i) The box support: $\bar{\beta}^{rcv}(l)$; (ii) The The Log Hazard Ratio (LHR) in the high-risk box: $\bar{\lambda}^{rcv}(l)$; (iii) The Log-rank Test (LRT) between the high vs. low-risk box: $\bar{z}^{rcv}(l)$; (iv) The Minimal Event-Free Probability ($MEFP$): $\bar{P}_0'^{\,rcv}(l)$; (v) The Minimal Event-Free Time ($MEFT$): $\bar{T}_0'^{\,rcv}(l)$.

4. Simulations

4.1 Design

The p-dimensional covariates $\mathbf{x}_i = [x_{i,1} \dots x_{i,p}]^T$ were simulated by drawing independent variates for $i \in \{1, \dots, n\}$ from a p-multivariate uniform distribution on the interval $[0, 1]$: $\mathbf{x}_i \sim U_p[0, 1]$ with $n = 250$ and $p = 3$.

Simulations were carried out according to section 2.2.1. Simulated realizations of true survival times u_i were drawn independently from an exponential distribution with rate parameter λ (and mean $\frac{1}{\lambda}$): $u_i \sim \text{Exp}(\lambda)$. Individual hazards rates λ_i were estimated by the CPH model as described in section 2. Simulated realizations c_i of true censoring times were independently sampled from a uniform distribution: $c_i \sim U[0, v]$, so that approximately $100 \times \pi(\%)$ of the simulated realizations of observed survival times $t_i = \min(u_i, c_i)$ were censored, where $\pi \in \{0.3, 0.5, 0.7\}$. Finally, the simulated realizations of observed event (non-censoring) random variable indicator were as follows: $\delta_i = I(u_i \leqslant c_i)$.

For simplification, subsequent simulations shown here were done:

- by characterization of the first coverage box $\hat{B}_1$, using constrained peeling, without pasting and with default meta-parameter values $(\alpha_0, \beta_0) = (0.05, 0.05)$
- using the log hazards ratios or relative risk $\hat{\eta}(l)$ as peeling criterion, with $\pi = 0.5$
- for three concurrent models: Values of the regression parameter $\boldsymbol{\beta} = [\beta_1 \dots 0_j \dots \beta_p]^T$ were set with $j \in \varnothing \cup \{1, \dots, p\}$ to simulate various levels of relationship between survival times and variables (i.e. variable informativeness):
$$\begin{cases} \text{Model \#1:} & \boldsymbol{\beta} = [\beta_1 \ \beta_2 \ \beta_3]^T \\ \text{Model \#2:} & \boldsymbol{\beta} = [\beta_1 \ \beta_2 \ 0]^T \\ \text{Model \#3:} & \boldsymbol{\beta} = [0 \ \ 0 \ \ 0]^T \end{cases}$$
- by using $K = 5$-fold cross-validation, $A = 1000$ for the cross-validated p-values and $B = 100$ independent Monte-Carlo replications for generating sampling distributions and inferring point and CI of statistics of interest.

4.2 Results

The results are compared on the basis of the box cross-validated estimates of the recovered decision rule and/or of the descriptive survival endpoints statistics as stated in section 3.1.2. We first compared the performance of our "Replicated Combined CV" K-fold cross-validation technique, denoted RCCV (section 3.1.5) with the situation where no cross-validation was done at all. Results for model #2 are shown in Figure 1. We then compared in Figure 2 the survival distribution curves estimates for each model and cross-validation technique (including none). Finally, we show in Table 1 the comparison between all three models for our RCCV technique.

4.2.1 On Peeling Trajectories

Peeling trajectories are estimated by a step function versus the box support/mass (Figure 1). They are read from right to left as they track the top-down direction of box induction process (peeling loop) of our Patient Recursive Survival Peeling method (Algorithm 1). Cross-validated peeling trajectories are, up to sampling variability:

- Monotone functions for each input variable $\mathbf{x}_j$, for $j \in \{1, \dots, p\}$.
- Monotone increasing functions for Log Hazard Ratios (LHR) $\bar{\lambda}_l$.
- Monotone increasing functions for Log-Rank Tests (LRT) $\bar{z}_l$.
- Monotone decreasing functions for Minimal Event-Free Probability ($MEFP$) $\bar{P}'_{0,l}$.
- Monotone decreasing functions for Maximal Event-Free Time ($MEFT$) $\bar{T}'_{0,l}$.
- Converging towards the input space coordinates of the maximum of the uncensored "surrogate" outcome y (see section 2.2.2).

4.2.2 On Trace Curves

Trace curves of variable importance and variable usage are estimated by piece-wise linear and step functions, respectively, vs the box support/mass (Figure 1). Similarly to peeling trajectories, they are read from right to left. Trace curves of variable importance show on a single plot: (i) the amplitude of used variables, (ii) the order (prioritization) with which these variables are used and (iii) the extent to which each variable is used. Variable traces are reminiscent of the concept of variable selection from the fields of decision tree and regularization.

4.2.3 On Survival Curves

Each subplot of Figure 2 corresponds to the last peeling step of our Patient Recursive Survival Peeling method for each tested model and cross-validation technique (including none). They show Kaplan-Meir estimates of the survival functions (as a function of survival time) of both in-box (red) and out-of-the-box (black) samples, corresponding respectively to the high-risk vs. the low-risk groups, along with cross-validated p-values $\hat{p}^{cv}(l)$ (see section 3.3). A single survival curve exists at the first peeling step, corresponding to the first box covering the entire data (not shown). As the peeling progresses, the survival curves of in-box and out-of-the-box samples further separate until the peeling stops (Figure 2).

4.2.4 Specific Comments on Plots

Our "Replicated Combined CV" (RCCV) technique tend to effectively: (i) smooth out peeling trajectories/profiles, and (ii) prune off peeling trajectories/profiles lengths. Compare for instance results for model #2: $\bar{L}^{rcv} = 27$ without CV (NOCV), as compared to $\bar{L}^{rcv} = 17$ for RCCV (Figure 1). Figure 1 also shows traces of variable importance $\widehat{vi}_l$ (top) and variables usage $\widehat{vu}_l$ (bottom) for model #2. Non-informative (noise/random) variables can be selectively eliminated from the model after RCCV: while the noise variable $\mathbf{x}_3$ in model #2 has been used in the absence of cross-validation (NOCV - dotted blue curve), as can be seen from its peeling profile and variable usage plots, its peeling profile is mostly horizontal (unused) in the presence of our cross-validation technique RCCV (solid blue curve). In fact, variable trace and variable usage plots also show that $\mathbf{x}_3$ (in model #2) is not used at all for $\bar{\beta}^{rcv}(l = 6) \lessapprox 0.384$ (third column, middle and bottom row of Figure 1).

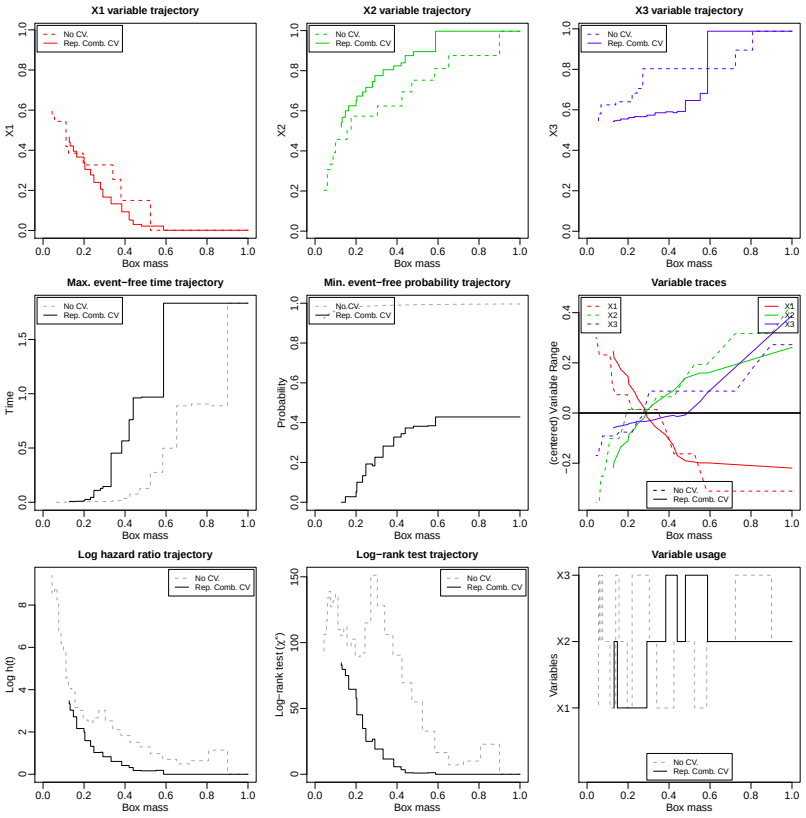

Figure 1: *Comparison of cross-validation techniques results on peeling trajectories and trace plots of variable importance and variable usage in model #2. Results without any cross-validation (NOCV) are compared to those obtained with "Replicated Combined CV" (RCCV).*

The very high LHR and LRT values in the non-cross-validated results ($\bar{\lambda}^{rcv}(l = 27) = 9.411$ and $\bar{z}^{rcv}(l = 27) = 93.321$) clearly reflect bias and/or overfitting. This is evident when comparing

to the much more conservative values obtained from the corresponding "Replicated Combined CV" (RCCV) peeling profiles: $\bar{\lambda}^{rcv}(l = 17) = 3.474$ and $\bar{z}^{rcv}(l = 17) = 84.634$ (Figure 1).

Further, the overly impressive p-values of separation of survival distributions in the non-cross-validated (NOCV) results of Figure 2 reflects again bias and/or overfitting. This is especially evident for the results of model #3: the survival probability curves of the high-risk (red curve in-box) and low-risk (black curve out-of-box) groups are well separated ($p \approx 0.0254$) in the absence of cross-validation (NOCV), while the corresponding "Replicated Combined CV" (RCCV) curves overlap with a cross-validated p-value ($p^{cv} \approx 0.9326$) that is no longer significant (bottom row of Figure 2).

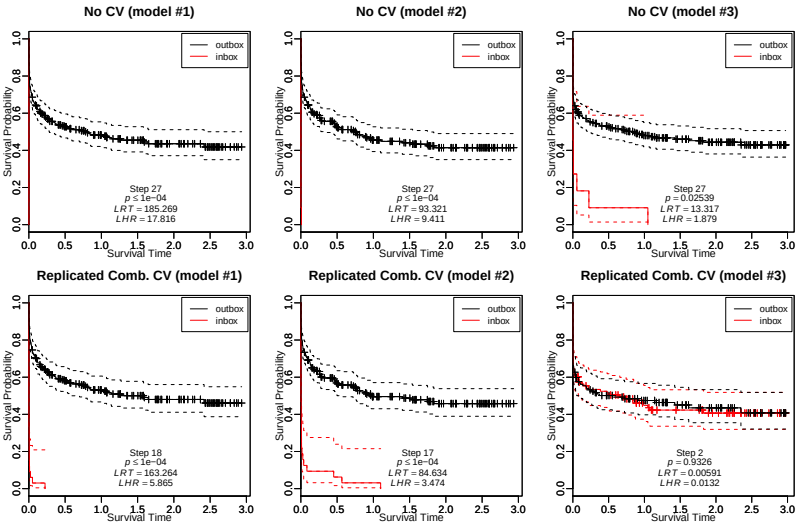

Figure 2: *Comparison of CV results for the Kaplan-Meir survival probability curves of the high-risk (red curve in-box) and low-risk (black curve out-of-box) groups in all models. Top row: No cross-validation (NOCV), bottom row: 'Replicated Combined CV" (RCCV). Left column: model #1, middle column: model #2, right column: model #3. For conciseness, only the last peeling step of the peeling sequence is shown for each tested model and cross-validation technique (including none).*

Using now "Replicated Combined Cross-Validation" alone (RCCV), we noticed striking differences in cross-validated peeling trajectories and variable traces between all models, that is, either (i) when all noise covariates $(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$ are introduced in the model (#3), or (ii) when one noise covariate only $(\mathbf{x}_3)$ is introduced in the model (#2) or (iii) when none is introduce in the model (#1). For instance, all peeling trajectories related to model #3 are much shorter in length and especially flatter than in the other models (not shown), indicating little or no usage as expected (Table 1). Similar observations can be made for variable importance and variable usage curves of model #3: the curves stop at box mass $\bar{\beta}^{rcv}(\bar{L}^{rcv}) \approx 0.432$ after only two steps ($\bar{L}^{rcv} = 2$), as compared to $\bar{\beta}^{rcv}(\bar{L}^{rcv}) \approx 0.132$ with $\bar{L}^{rcv} = 18$ for model #1 and $\bar{\beta}^{rcv}(\bar{L}^{rcv}) \approx 0.128$ with $\bar{L}^{rcv} = 17$ for model #2 (Table 1). Also, we noticed how LHR and LRT cross-validated peeling trajectories in model #1 reach higher levels than in model #2 (Table 1). This was expected due to our simulation design where each covariate of model #1 additively contributes to the hazards and to the separation of survival distributions.

Table 1: *Comparison of cross-validated decision rules between all models #1, #2, #3 for the "Replicated Combined CV" (RCCV) technique. For conciseness, only the last steps $\bar{L}^{rcv}$ are shown.*

	$\bar{L}^{rcv}$	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\bar{T}_0'^{\,rcv}(l)$	$\bar{P}_0'^{\,rcv}(l)$	$\bar{\lambda}^{rcv}(l)$	$\bar{z}^{rcv}(l)$	$\bar{\beta}^{rcv}(l)$
model #1	18	$\mathbf{x}_1 \geqslant 0.513$	$\mathbf{x}_2 \leqslant 0.514$	$\mathbf{x}_3 \leqslant 0.615$	0.001	0.000	5.865	163.264	0.132
model #2	17	$\mathbf{x}_1 \geqslant 0.466$	$\mathbf{x}_2 \leqslant 0.517$	$\mathbf{x}_3 \leqslant 0.541$	0.006	0.000	3.474	84.634	0.128
model #3	2	$\mathbf{x}_1 \leqslant 0.910$	$\mathbf{x}_2 \geqslant 0.363$	$\mathbf{x}_3 \leqslant 0.773$	1.827	0.407	0.013	0.006	0.432

5. Conclusion

Our replicated cross-validation strategy, namely the "Replicated Combined CV" (RCCV) procedure is effective in attenuating the over-fitting and/or bias issues for the sample size tested.

Overall, results testify of the effectiveness of the Replicated Combined Cross-Validation (RCCV) technique to help select the informative variables and possibly to rank them by im-

portance in a survival bump hunting model.

The stepwise variable usage procedure in the peeling loop of Algorithm 1 naturally induces an inflation of variance estimates simply because each peeling step is conditional on the previous step. Therefore, replications of our K-fold Combined Cross-Validation for recursive peeling methods, although optional, are recommended to reduce the variability of box and survival estimates for both CV procedures and attenuates the non-monotonicity of peeling trajectories.

REFERENCES

[1] Ahn, H. and Loh, W. Y. (1994), "Tree-structured proportional hazards regression modeling," *Biometrics*, 50, 471–85.

[2] Ambroise, C. and McLachlan, G. J. (2002), "Selection bias in gene extraction on the basis of microarray gene-expression data," *Proc Natl Acad Sci U S A*, 99, 6562–6.

[3] Baker, S., Kramer, B., and Srivastava, S. (2002), "Markers for early detection of cancer: Statistical guidelines for nested case-control studies," *BMC Medical Research Methodology*, 2, 4.

[4] Ciampi, A., J., T., Nakache, J. P., and B., A. (1986), "Stratification by stepwise regression, correspondence analysis and recursive partition," *Comput. Statist. Data Anal.*, 4, 185–204.

[5] Cox, D. (1972), "Regression models and life-tables," *J R Statist Soc*, 30, 248–275.

[6] Davis, R. B. and Anderson, J. R. (1989), "Exponential survival trees," *Stat Med*, 8, 947–61.

[7] Dazard, J.-E., Choe, M., and Santana, A. (2015), "Contributed R Package PrimSRC for PRIM in Survival, Regression and Classification Settings," The Comprehensive R Archive Network (in prep), http://cran.r-project.org/web/packages/PrimSRC/.

[8] Dobbin, K. and Simon, R. (2007), "Sample size planning for developing classifiers using high dimensional DNA microarray data," *Biostatistics*, 8, 101–117.

[9] Dupuy, A. and Simon, R. (2007), "Critical Review of Published Microarray Studies for Cancer Outcome and Guidelines on Statistical Analysis and Reporting," *J. Nat. Cancer Institute*, 99, 147–157.

[10] Friedman, J. and Fisher, N. (1999), "Bump hunting in high-dimensional data," *Statistics and Computing*, 9, 123–143.

[11] Gordon, L. and Olshen, R. A. (1985), "Tree-structured survival analysis," *Cancer Treat Rep*, 69, 1065–9.

[12] Haibe-Kains, B., El-Hachem, N., Birkbak, N., Jin, A., Beck, A., Aerts, H., and Quackenbush, J. (2013), "Inconsistency in large pharmacogenomic studies," *Nature*, 504, 389–394.

[13] Harrell, F. E., J., Califf, R. M., Pryor, D. B., Lee, K. L., and Rosati, R. A. (1982), "Evaluating the yield of medical tests," *JAMA : the journal of the American Medical Association*, 247, 2543–6.

[14] Hastie, T., Tibshirani, R., and Friedman, J. (2009), *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, New York: Springer Science.

[15] Ishwaran, H., Kogalur, U., Blackstone, E., and Lauer, M. (2008), "Random survival forests," *The Annals of Applied Statistics*, 2, 841–860.

[16] LeBlanc, M. and Crowley, J. (1992), "Relative risk trees for censored survival data," *Biometrics*, 48, 411–25.

[17] — (1993), "Survival trees by goodness of split," *J Amer Stat Assoc*, 88, 457–67.

[18] LeBlanc, M., Jacobson, J., and Crowley, J. (2002), "Partitioning and peeling for constructing prognostic groups," *Stat Methods Med Res*, 11, 247–74.

[19] LeBlanc, M., Moon, J., and Crowley, J. (2005), "Adaptive Risk Group Refinement," *Biometrics*, 61, 370–378.

[20] McShane, L., Cavenagh, M., Lively, T., Eberhard, D., Bigbee, W., Williams, P., Mesirov, J., Polley, M.-Y., Kim, K., Tricoli, J., Taylor, J., Shuman, D., Simon, R., Doroshow, J., and Conley, B. (2013), "Criteria for the use of omics-based predictors in clinical trials: explanation and elaboration," *BMC Medicine*, 11, 220.

[21] McShane, M., Cavenagh, M., Lively, T., Eberhard, D., Bigbee, W., Mickey Williams, P., Mesirov, J., Polley, M.-Y., Kim, K., Tricoli, J., Taylor, J., Shuman, D., Simon, R., Doroshow, J., and Conley, B. (2013), "Criteria for the use of omics-based predictors in clinical trials," *Nature*, 502, 317–320.

[22] Michiels, S., Koscielny, S., and Hill, C. (2005), "Prediction of cancer outcome with microarrays: a multiple random validation strategy," *Lancet*, 365, 488–492.

[23] Molinaro, A., Simon, R., and Pfeiffer, R. (2005), "Prediction error estimation: a comparison of resampling methods," *Bioinformatics*, 21, 3301–3307.

[24] Polonik, W. and Wang, Z. (2010), "PRIM Analysis," *Journal of Multivariate Analysis*, 101, 525–540.

[25] Segal, M. R. (1988), "Regression Trees for Censored Data," *Biometrics*, 44, 35–47.

[26] Simon, R., Radmacher, M., Dobbin, K., and McShane, L. (2003), "Pitfalls in the Use of DNA Microarray Data for Diagnostic and Prognostic Classification," *J. Nat. Cancer Institute*, 95, 14–18.

[27] Simon, R., Subramanian, J., Li, M.-C., and Menezes, S. (2011), "Using cross-validation to evaluate predictive accuracy of survival risk classifiers based on high-dimensional data," *Briefings in Bioinformatics*, 12, 203–214.

[28] Subramanian, A. and Simon, R. (2010), "Gene expression-based prognostic signatures in lung cancer: ready for clinical use?" *J. Natl. Cancer Inst.*, 102, 464474.

[29] — (2011), "An evaluation of resampling methods for assessment of survival risk prediction in high-dimensional settings," *Stat. Med.*, 30, 642653.

[30] Therneau, T., Grambsch, P., and Fleming, T. (1990), "Martingale based residuals for survival models," *Biometrika*, 77, 147–160.

[31] Varma, S. and Simon, R. (2006), "Bias in error estimation when using cross-validation for model selection," *BMC bioinformatics*, 7, 91–99.