How to use the rbsurv Package

HyungJun Cho, Sukwoo Kim, Soo-heang Eo, and Jaewoo Kang

May 1, 2024

Contents

1 Introduction

The *rbsurv* package is designed to select survival-associated genes, based on a likelihood function. It utilizes the partial likelihood of the Cox model which has been the basis for many of the existing methods. Our algorithm is simple and straight-forward, but its functions such as the generation of multiple gene models and the incorporation of significant risk factors are practical. For robustness, this package also selects survival-associated genes by separating training and validation sets of samples because such a cross-validation technique is essential in predictive modeling for data with large variability. It employs forward selection, generating a series of gene models and selecting an optimal model. Furthermore, iterative runs after putting aside the previously selected genes can discover the masked genes that may be missed by forward selection (see Cho et al. for details). The *rbsurv* package employs libraries *survival* and *Biobase*.

2 Robust likelihood-based survival modeling

Suppose the data consists of G genes and N samples, and each sample has its observed (survival or censoring) time and censoring status. Thus, it consists of the triple $(Y_j, \delta_j, \mathbf{X}_j)$, $j = 1, \ldots, N$, where Y_j and δ_j are observed time and censoring status (usually, 1=died, 0=censored) for the j-th sample respectively, and $\mathbf{X}_j = (X_{1j}, X_{2j}, \ldots, X_{Kj})$ is the j-th vector of the expression values for K genes ($K < N$ and $K \subset G$). Let $Y_{(1)} < Y_{(2)} < \ldots < Y_{(D)}$ denote the ordered times with D distinct values and $X_{(i)k}$ be the k-th gene associated with the sample corresponding to $Y_{(i)}$. The Cox proportional hazards model is $h(y|X_1, X_2, \ldots, X_K) = h_0(y) \exp(\sum_{k=1}^{K} \beta_k X_k)$, where $h(y|X_1, X_2, \ldots, X_K)$ is the hazard rate at time y for a sample with risk vector $(X_1, X_2, \ldots, X_K)$, $h_0(y)$ is an arbitrary baseline hazard rate, and β_k is the coefficient for the k-th gene. The partial likelihood for the Cox model is

$$\sum_{i=1}^{D} \sum_{k=1}^{K} \beta_k X_{(i)k} - \sum_{i=1}^{D} \ln[\sum_{j \in R(Y_{(i)})} \exp(\sum_{k=1}^{K} \beta_k X_{jk})], \tag{1}$$

where $R(Y_{(i)})$ is the set of all samples that are still under study at a time just prior to $Y_{(i)}$. Maximizing the likelihood provides the maximum likelihood estimates (MLE) of the coefficients, so denote the MLEs by $\hat{\beta}_1, \hat{\beta}_2, \ldots, \hat{\beta}_k$. Then, as a goodness-of-fit, we can use the fitted partial likelihood:

$$\text{loglik} = \sum_{i=1}^{D} \sum_{k=1}^{K} \hat{\beta}_k X_{(i)k} - \sum_{i=1}^{D} \ln[\sum_{j \in R(Y_{(i)})} \exp(\sum_{k=1}^{K} \hat{\beta}_k X_{jk})]. \tag{2}$$

The negative log-likelihood (-loglik) is greater than zero, so the smaller -loglik the model better. For robustness, however, the model should be evaluated by independent validation samples rather than the training samples used for fitting the model such as

$$\text{loglik}^* = \sum_{i=1}^{D^*} \sum_{k=1}^{K} \hat{\beta}_k^0 X_{(i)k}^* - \sum_{i=1}^{D^*} \ln[\sum_{j \in R(Y_{(i)}^*)} \exp(\sum_{k=1}^{K} \hat{\beta}_k^0 X_{jk}^*)], \tag{3}$$

where $*$ indicate the use of the validation samples and the estimates $\hat{\beta}_1^0, \hat{\beta}_2^0, \ldots, \hat{\beta}_k^0$ are obtained by the training samples. For robust gene selection, we thus use training samples for model fitting and validation samples for model validation. This cross-validation is repeated many times independently. In other words, we fit the Cox model with a gene (or genes) and select a gene (or genes) maximizing mean loglik* (i.e., minimizing the mean negative loglik*).

3 Algorithm

Suppose the data consists of expression values X for G genes and N samples. Each sample has its observed (survival or censoring) time Y and censoring status δ. We assume

that expression values are already normalized and transformed in appropriate ways. Prior gene selection such as univariate survival modeling and/or biological pre-selection can also be performed if necessary. Univariate survival modeling can be performed in our software program. Our algorithm is summarized as follows. R function arguments are also included.

1. Randomly divide the samples into the training set with $N(1-p)$ samples and the validation set with Np samples, e.g., $p = 1/3$, (*n.fold=3*). Fit a gene to the training set of samples and obtain the parameter estimate $\hat{\beta}_i^0$ for the gene. Then evaluate loglik* with the parameter estimate, $\hat{\beta}_i^0$, and the validation set of samples, $(Y_i^*, \delta_i^*, X_i^*)$. Perform this evaluation for each gene.

2. Repeat the above procedure B times, e.g., $B = 100$, (*n.iter=100*), thus obtaining B loglik*s for each gene. Then select the best gene with the smallest mean negative loglik* (or the largest mean loglik*). The best gene is the most survival-associated one that is selected by the robust likelihood-based approach.

3. Let $g_{(1)}$ be the selected best gene in the previous step. Adjusting for $g_{(1)}$, find the next best gene by repeating the previous two steps. In other words, evaluate $g_{(1)} + g_j$ for every j and select an optimal two-gene model, $g_{(1)} + g_{(2)}$.

4. Continue this forward gene selection procedure until fitting is impossible because of the lack of samples, resulting in a series of K models $\mathcal{M}_1 = g_{(1)}$, $\mathcal{M}_2 = g_{(1)} + g_{(2)}$, ..., $\mathcal{M}_{K-1} = g_{(1)} + g_{(2)} + \ldots + g_{(K-1)}$, $\mathcal{M}_K = g_{(1)} + g_{(2)} + \ldots + g_{(K)}$.

5. Compute AICs for all the K candidate models, $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_K$, and select an optimal model with the smallest AIC.

6. Put aside the genes in the optimal model in the previous step. Then repeat steps 2-6. This can be repeated several times sequentially, e.g, 3 times, (*n.seq=3*), generating multiple optimal models.

In addition, suppose that p risk factors, $Z_1, Z_2, \ldots, Z_p$, are available for each sample. Then risk factors can be included in every modeling of the previous algorithm.

4 Example: Glioma Data

We demonstrate the use of the package with a microarray data set for patients with gliomas. This real data set consists of gene expression value, survival time, and censoring status of each of 85 patients with gliomas (Freije et al., 2004). For this study, Affymetrix U133A and U133B chips were used and dCHIP was used to convert data files (.CEL) into expression values with median intensity normalization. This data set originally consists of more than 40,000 probe sets, but only a sub-dataset made up of 100 probe sets was stored into the *rbsurv* package for demonstration.

To run *rbsurv*, the data can be prepared as follows.

```
> library(rbsurv)
> data(gliomaSet)
> gliomaSet

ExpressionSet (storageMode: lockedEnvironment)
assayData: 100 features, 85 samples
  element names: exprs
protocolData: none
phenoData
  sampleNames: Chip1 Chip2 ... Chip85 (85 total)
  varLabels: Time Status Age Gender
  varMetadata: labelDescription
featureData: none
experimentData: use 'experimentData(object)'
  pubMedIds: 15374961
Annotation:

> x <- exprs(gliomaSet)
> x <- log2(x)
> time <- gliomaSet$Time
> status <- gliomaSet$Status
> z <- cbind(gliomaSet$Age, gliomaSet$Gender)
>
```

We here took log2-transformation without any other normalizations. An appropriate normalization can be taken if needed. If the data is ready, *rbsurv* can be run as follows.

```
> fit <-  rbsurv(time=time, status=status, x=x, method="efron", max.n.genes=20)

Please wait... Done.
```

This sequentially selects genes one gene at a time to build an optimal gene model. Once a large gene model is constructed, an optimal gene model is determined by AICs. If there exist ties in survival times, Efron's method is used (*method="efron"*). Note that it is computationally expensive and the data is high-throughput. Therefore, you should be patient to obtain the output. To save time, we can reduce the number of genes considered up to 20 genes among 100 initial genes (*max.n.genes=20*). The 20 genes are selected by fitting univariate Cox models. The above command generates the following output.

```
> fit$model
```

```
    Seq Order Gene nloglik    AIC Selected
0     1     0    0  228.74 457.47
110   1     1   46  218.53 439.05 *
2     1     2   57  202.21 408.42 *
3     1     3   43  195.50 396.99 *
4     1     4   34  194.01 396.01 *
5     1     5   99  192.14 394.29 *
6     1     6   36  189.81 391.63 *
7     1     7    8  188.80 391.59 *
8     1     8   86  187.90 391.80
9     1     9   68  187.52 393.04
10    1    10   56  187.42 394.84
11    1    11   15  186.68 395.37
12    1    12   29  185.54 395.09
13    1    13   75  185.54 397.09
14    1    14   67  185.41 398.83
15    1    15   40  183.76 397.52
16    1    16   98  183.04 398.09
17    1    17   19  182.25 398.49
18    1    18   39  181.99 399.98
19    1    19   96  181.88 401.76
```

This large gene model contains survival-associated genes which were selected one at a time by forward selection. Note that the first row has no gene ID because it was fitted with no expression profile. The size of the large gene model was determined by the numbers of samples and genes considered. The AICs tend to decrease and then increase, while negative log-likelihoods (nlogliks) always decrease. Thus, we select an optimal model with the smallest AIC. The selected parsimonious model consists of the survival-associated genes, which are marked with asterisks (*).

Potential risk factors can be included in modeling and it can be run sequentially. For example, use $rbsurv(time = time, status = status, x = x, z = z, alpha = 0.05, n.seq = 3)$ for significant risk factors with significance level 0.05 and 3 sequential runs. All the risk factors are included if the significance level is 1. For the detailed algorithm, refer to Cho et al. (submitted).

5 Argument description

In this section, we describe the arguments with the following command.

```
library(rbsurv)
fit <- rbsurv(time=time, status=status, x=x, z=z, alpha=0.05, gene.ID=NULL,
              method="efron", max.n.genes=100, n.iter=100, n.fold=3,
              n.seq=3, seed = 1234)
```

Table 1: Argument description

Argument	Description
time	a vector for survival times
status	a vector for survival status, 0=censored, 1=event
x	a matrix for expression values (genes in rows, samples in columns)
z	a matrix for risk factors
alpha	a significance level for evaluating risk factors
gene.ID	a vector for gene IDs; if NULL, row numbers are assigned.
method	a character string specifying the method for tie handling.
n.iter	the number of iterations for gene selection
n.fold	the number of partitions of samples
n.seq	the number of sequential runs or multiple models
seed	a seed for sample partitioning
max.n.genes	the maximum number of genes considered

The required arguments *time* and *status* are vectors for survival times and survival status (0=censored, 1=event) and x is a matrix for expression values (genes in rows, samples in columns). The optional argument z is a matrix for risk factors. To include only the significant risk factors, a significance level less than 1 is given to *alpha*, e.g., $alpha = 0.05$. In addition, there are several controlled arguments. *gene.ID* is a vector for gene IDs; if NULL, row numbers are assigned. *method* is a character string specifying the method for tie handling. One of *efron*, *breslow*, *exact* can be chosen. If there are no tied death times all the methods are equivalent. In the algorithm, *n.fold* is the number of partitions of samples in step 1, *n.iter* is the number of iterations for gene selection in step 2, and *n.seq* is the number of sequential runs (or multiple models) in step 6. *seed* is a seed for sample partitioning. *max.n.genes* is the maximum number of genes considered. If the number of the input genes is greater than the given maximum number, it is reduced by fitting individual Cox models and selecting the genes with the smallest p-values. The input arguments of **rbsurv** are summarized in Table 1. The major output *fit$model* contains survival-associated gene models with gene IDs, nlogliks, and AICs. The genes in the optimal model with the smallest AIC are marked with asterisks (*).

6 Conclusion

This package allows ones to build multiple gene models sequentially rather than a single gene model. Furthermore, other covariates such as age and gender can also be incorporated into modeling with gene expression profiles. Each model contains survival-associated genes that are selected robustly by separating training and test sets many times.

References

Cho, H., Yu, A., Kim, S., Kang, J., Hong, S-M., (2009). Robust Likelihood-Based Survival Modeling with Microarray Data. *Journal of Statistical Software* 29(1), 1-16. URL http://www.jstatsoft.org/v29/i01/.

Freije, W.A., Castro-Vargas, F.E., Fang, Z., Horvath, S., Cloughesy, T., Liau, L.M., Mischel, P.S. and Nelson, S.F. (2004). Gene expression profiling of gliomas strongly predicts survival, *Cancer Research*, 64:6503-6510.