An introduction to **RLassoCox**

Wei Liu
freelw@qq.com
Heilongjiang Institute of Technology

October 28, 2020

1 Introduction

RLassoCox is a package that implements the RLasso-Cox model proposed by Wei Liu. The RLasso-Cox model integrates gene interaction information into the Lasso-Cox model for accurate survival prediction and survival biomarker discovery. It is based on the hypothesis that topologically important genes in the gene interaction network tend to have stable expression changes. The RLasso-Cox model uses random walk to evaluate the topological weight of genes, and then highlights topologically important genes to improve the generalization ability of the Lasso-Cox model. The RLasso-Cox model has the advantage of identifying small gene sets with high prognostic performance on independent datasets, which may play an important role in identifying robust survival biomarkers for various cancer types.

RLassoCox solves the following problem

$$\max_{\beta} \sum_{i=1}^{m} \left(x_{j(i)}^T \beta - \log \left(\sum_{j \in R_i} e^{x_i^T \beta} \right) \right) - \lambda \sum_{k=1}^{p} \varphi(w_k) |\beta_k| \tag{1}$$

over a grid of values of λ . Here the first term represents the log partial likelihood function, and the second term is a topologically weighted L_1-norm constraint.

2 Installation

Like many other R packages, the simplest way to obtain RLassoCox is to install it directly from Bioconductor. Type the following command in R console:

```
# if (!requireNamespace("BiocManager", quietly = TRUE))
#     install.packages("BiocManager")
# BiocManager::install("RLassoCox")
```

3 RLassoCox

In this section, we will go over the main functions, see the basic operations and have a look at the outputs. Users may have a better idea after this section what functions are available, which one to choose, or at least where to seek help.

First, we load the RLassoCox package:

```
library("RLassoCox")
```

```
## Loading required package:  igraph
##
## Attaching package:  'igraph'
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
## The following object is masked from 'package:base':
##
##     union
## Loading required package:  Matrix
## Loading required package:  survival
## Loading required package:  glmnet
## Loaded glmnet 4.0-2
```

The RLassoCox package trains the RLasso-Cox model based on gene expression profiles, survival information and gene interaction networks. We load a set of data created beforehand for illustration. Users can either load their own data or use those saved in the workspace.

```
data(mRNA_matrix) # gene expression profiles
data(survData)    # survival information
data(dGMMirGraph) # gene interaction network
```

The commands load an input gene expression matrix **mRNA_matrix**, a data frame **survData** that contains survival information, and an igraph object **survData** that contains the KEGG network constructed by the R package **iSubpathwayMiner**.

survData is an n x 2 matrix, with a column "time" of failure/censoring times, and "status" a 0/1 indicator, with 1 meaning the time is a failure time, and zero a censoring time.

```
head(survData)

##              status time
## TCGA-02-0001   TRUE  358
## TCGA-02-0003   TRUE  144
## TCGA-02-0006   TRUE  558
## TCGA-02-0007   TRUE  705
## TCGA-02-0009   TRUE  322
## TCGA-02-0010   TRUE 1077
```

In order to train and test the predictive performance of the RLasso-Cox model, we divide the data set into a training set and a test set.

```
set.seed(20150122)
train.Idx <- sample(1:dim(mRNA_matrix)[1], floor(2/3*dim(mRNA_matrix)[1]))
test.Idx <- setdiff(1:dim(mRNA_matrix)[1], train.Idx)
x.train <- mRNA_matrix[train.Idx ,]
x.test <- mRNA_matrix[test.Idx ,]
y.train <- survData[train.Idx,]
y.test <- survData[test.Idx,]
```

Train the RLasso-Cox model based on the training set data:

```
mod <- RLassoCox(x=x.train, y=y.train, globalGraph=dGMMirGraph)

## Calculating Cox p-value...Done
## Performing directed random walk...Done
```

The **RLassoCox** function depends on the **glmnet** package[1]. **mod** contains a list object that includes a **glmnet** object **glmnetRes** and a topological weight vector **PT**. **PT** is the topological weights of the genes.

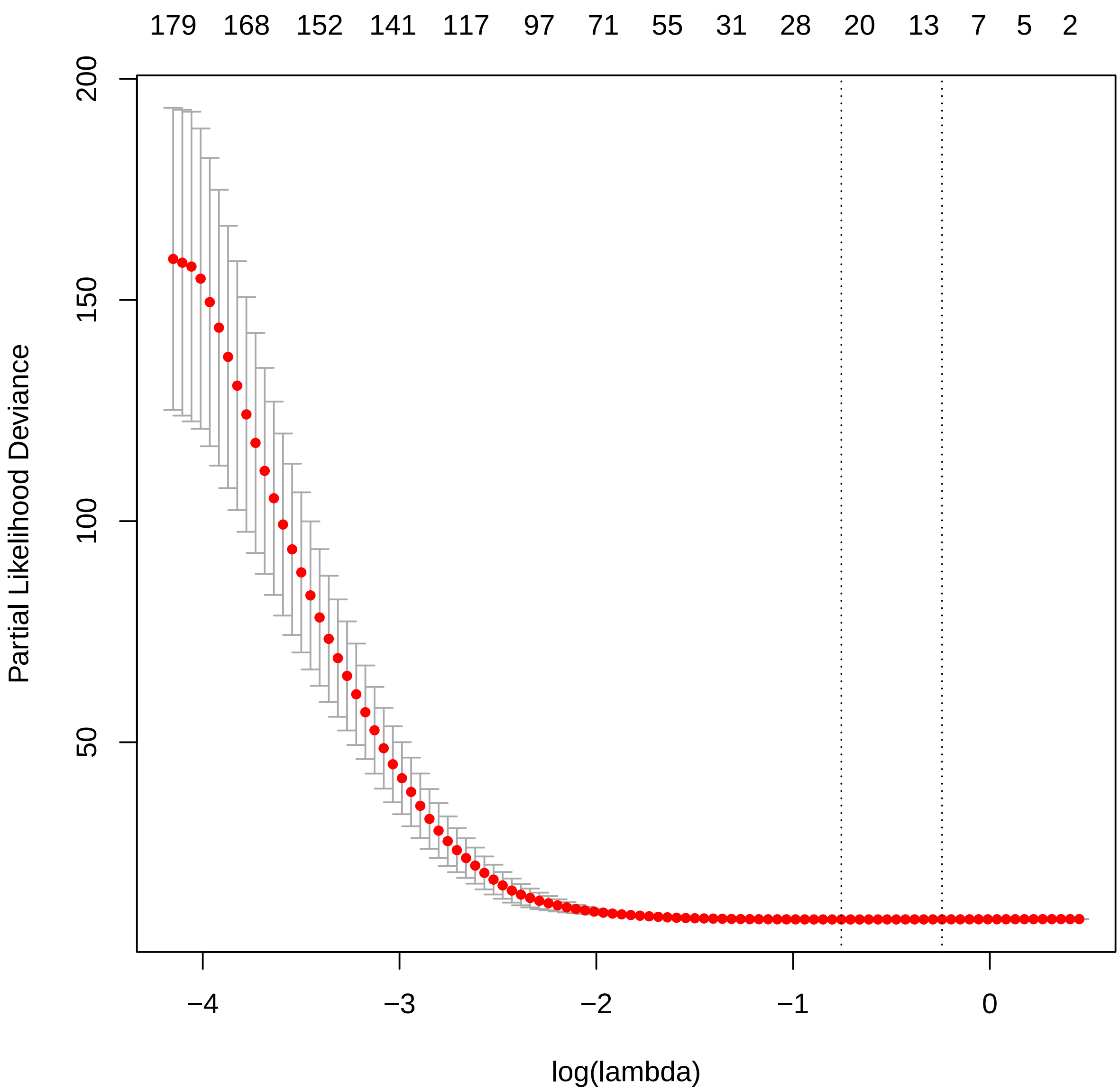
```
head(mod$PT)

##         7010         1261        10209        22798         2526        10965
## 0.0007118464 0.0004399878 0.0008614412 0.0012176743 0.0004669061 0.0004577181
```

glmnetRes contains all the relevant information of the fitted model for further use. We can use the **plot**, **print**, **coef** and **predict** functions in the **glmnet** package to easily extract the components of the model.

We can visualize the coefficients by executing the **plot** function:

```
plot(mod$glmnetRes)
```



Each curve in the figure corresponds to a variable (gene). It shows the path of the coefficient of each gene and L_1-norm when λ varies. The axis above indicates the number of nonzero coefficients at the current λ, which is the effective degrees of freedom for the RLasso-Cox model.

A summary of the **RLassoCox** path at each step is displayed if we just enter the object name or use the print function:

```
print(mod$glmnetRes)
```

```
##
## Call:  glmnet(x = x, y = Surv(time = y[, "time"], event = y[, "status"]),      family = "cox", alpha
##
##      Df  %Dev  Lambda
## 1     0  0.00 1.57600
## 2     2  0.13 1.50400
## 3     2  0.26 1.43600
## 4     3  0.40 1.37000
## 5     4  0.55 1.30800
## 6     5  0.69 1.24900
## 7     5  0.83 1.19200
## 8     5  0.96 1.13800
## 9     6  1.08 1.08600
## 10    6  1.20 1.03700
## 11    6  1.31 0.98960
## 12    7  1.44 0.94460
## 13    8  1.57 0.90170
## 14    9  1.71 0.86070
## 15   11  1.85 0.82160
## 16   12  2.02 0.78420
## 17   12  2.20 0.74860
## 18   13  2.36 0.71450
## 19   13  2.52 0.68210
## 20   14  2.67 0.65110
## 21   16  2.84 0.62150
## 22   17  3.04 0.59320
## 23   17  3.24 0.56630
## 24   18  3.42 0.54050
## 25   20  3.59 0.51600
## 26   21  3.76 0.49250
## 27   22  3.94 0.47010
## 28   24  4.15 0.44880
## 29   25  4.39 0.42840
## 30   27  4.64 0.40890
## 31   29  4.89 0.39030
## 32   28  5.11 0.37260
## 33   28  5.30 0.35560
## 34   29  5.48 0.33950
## 35   30  5.65 0.32400
## 36   29  5.83 0.30930
## 37   29  5.99 0.29520
## 38   31  6.15 0.28180
## 39   31  6.34 0.26900
## 40   36  6.52 0.25680
## 41   39  6.84 0.24510
## 42   43  7.28 0.23400
## 43   45  7.70 0.22330
## 44   48  8.12 0.21320
```

```
## 45   51  8.55 0.20350
## 46   55  9.02 0.19430
## 47   59  9.51 0.18540
## 48   61 10.00 0.17700
## 49   62 10.47 0.16900
## 50   61 10.96 0.16130
## 51   65 11.45 0.15390
## 52   67 11.95 0.14690
## 53   71 12.45 0.14030
## 54   75 13.02 0.13390
## 55   80 13.63 0.12780
## 56   81 14.20 0.12200
## 57   83 14.75 0.11650
## 58   87 15.30 0.11120
## 59   90 15.88 0.10610
## 60   97 16.50 0.10130
## 61  101 17.18 0.09668
## 62  104 17.89 0.09229
## 63  107 18.65 0.08809
## 64  106 19.37 0.08409
## 65  108 20.01 0.08027
## 66  113 20.64 0.07662
## 67  116 21.30 0.07314
## 68  117 21.96 0.06981
## 69  118 22.67 0.06664
## 70  120 23.44 0.06361
## 71  127 24.26 0.06072
## 72  135 25.16 0.05796
## 73  134 26.10 0.05532
## 74  137 26.93 0.05281
## 75  139 27.85 0.05041
## 76  141 28.83 0.04812
## 77  143 29.87 0.04593
## 78  148 31.09 0.04384
## 79  148 32.22 0.04185
## 80  147 33.31 0.03995
## 81  151 34.37 0.03813
## 82  148 35.42 0.03640
## 83  152 36.52 0.03475
## 84  152 37.54 0.03317
## 85  154 38.56 0.03166
## 86  158 39.65 0.03022
## 87  159 40.76 0.02885
## 88  159 41.89 0.02754
## 89  163 42.99 0.02628
## 90  163 44.04 0.02509
## 91  163 45.27 0.02395
## 92  168 46.38 0.02286
## 93  168 47.35 0.02182
## 94  171 48.38 0.02083
## 95  175 49.50 0.01988
## 96  176 50.52 0.01898
## 97  178 51.67 0.01812
## 98  178 52.72 0.01729
```

```
## 99  177 53.76 0.01651
## 100 179 54.78 0.01576
```

The first column **df** represents the number of non-zero coefficients, the second column **%Dev** represents the percent (of null) deviance explained, and the third column **Lambda** represents the value of λ.

The actual coefficients of genes at one or more λs within the range of the sequence can be obtained:

```
head(coef(mod$glmnetRes, s = 0.2))

## 6 x 1 sparse Matrix of class "dgCMatrix"
##                1
## 7010  .
## 1261  .
## 10209 0.04169522
## 22798 .
## 2526  .
## 10965 .
```

The **glmnetRes** model can be used to predict the risk of new patients at one or more λs.

```
lp <- predict(object = mod, newx = x.test, s = c(0.1, 0.2))
head(lp)

##                       1          2
## TCGA-02-0003  0.6575433  0.5633245
## TCGA-02-0011 -2.0901828 -1.0954352
## TCGA-02-0014 -0.7554994 -0.9513547
## TCGA-02-0024 -0.2260848 -0.5894672
## TCGA-02-0055  0.9075345  0.1954023
## TCGA-02-0060 -0.6100948 -0.6059411
```

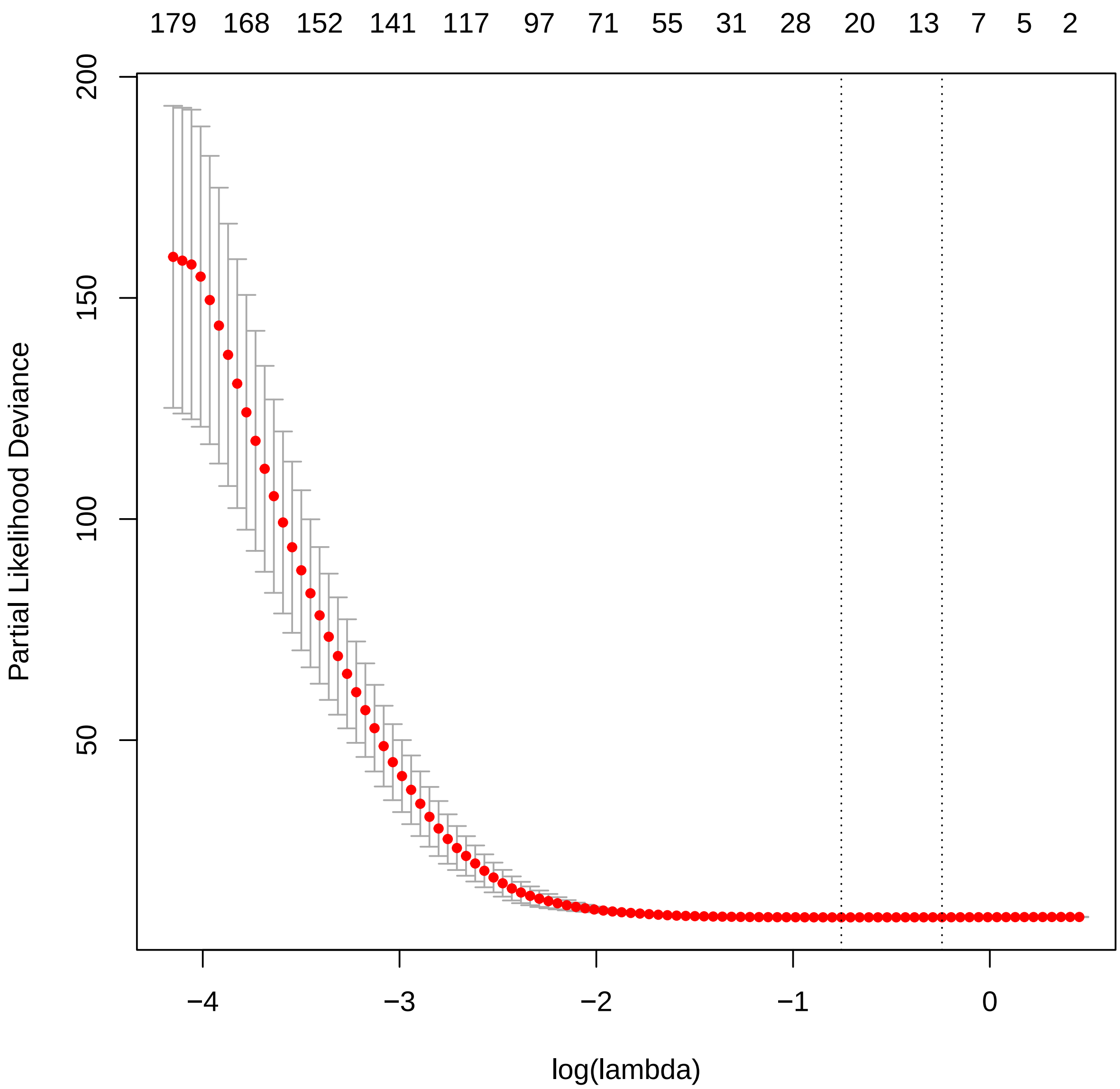
The function **cvRLassoCox** can be used to compute k-fold cross-validation for the RLasso-Cox model.

```
cv.mod <- cvRLassoCox(x=x.train, y=y.train,
                       globalGraph=dGMMirGraph, nfolds = 5)

## Calculating Cox p-value...Done
## Performing directed random walk...Done
## Performing cv.glmnet...Done
```

The **cvRLassoCox** function also returns a list object that contains a **cv.glmnet** object **glmnetRes** and a topological weight vector **PT**. In addition, the optimal λ value and a cross validated error plot can be obtained to help evaluate our model.

```
plot(cv.mod$glmnetRes, xlab = "log(lambda)")
```



In this plot, the left vertical line shows where the CV-error curve hits its minimum. The right vertical line shows us the most regularized model with CV-error within 1 standard deviation of the minimum. The optimal λ can be obtained:

```
cv.mod$glmnetRes$lambda.min
```

```
## [1] 0.4701213
```

```
cv.mod$glmnetRes$lambda.1se
```

```
## [1] 0.7842097
```

We can check the active covariates (genes) in our model and see their coefficients.

```r
coef.min <- coef(cv.mod$glmnetRes, s = "lambda.min")
coef.min
```

```
## 383 x 1 sparse Matrix of class "dgCMatrix"
##                   1
## 7010     .
## 1261     .
## 10209    .
## 22798    .
## 2526     .
## 10965    .
## 1362     .
## 22929  -0.020897935
## 5876     .
## 3265     .
## 8302     .
## 645      .
## 23480    .
## 1311     .
## 25       .
## 4598     .
## 3269     .
## 767      .
## 3385     .
## 80380    .
## 55176    .
## 1819     .
## 3930     .
## 4478     .
## 5617   -0.126328253
## 55341   0.006056988
## 5042     .
## 22800    .
## 3939     .
## 822      .
## 79695    .
## 55540    .
## 5934     .
## 5806     .
## 259230   .
## 51330    .
## 7098     .
## 8513     .
## 5836     .
## 64374    .
## 2184     .
## 988      .
## 5464     .
## 8309     .
## 10451    .
## 3574     .
## 873      .
## 3710     .
## 7314     .
```

```
## 114049  .
## 79723  -0.016504110
## 8030    .
## 1508    .
## 5223    .
## 6696    .
## 1788    .
## 3696    .
## 10552   .
## 5532    .
## 1796    .
## 5599    .
## 51700   .
## 10505   .
## 79132   .
## 3178    .
## 478     .
## 3799    .
## 135    -0.056788179
## 1571    .
## 6908    .
## 23600   .
## 35      .
## 2053    0.014389764
## 2746   -0.011474410
## 4615    .
## 5371    .
## 51744   .
## 36      .
## 3066    .
## 50940   .
## 1398    .
## 7378    .
## 5407    .
## 586     .
## 56288   .
## 4689    .
## 6732    .
## 3280    .
## 3298    .
## 4938    .
## 109     .
## 3576    .
## 7132    .
## 605     .
## 3485    0.065929675
## 3689    .
## 29780   .
## 3588    .
## 200316  .
## 771     .
## 51384   .
## 3375    .
## 4223    .
```

```
## 783      .
## 10846    .
## 4216     .
## 1026     .
## 1528     .
## 5606     .
## 89       .
## 9547     .
## 2213     .
## 8228     .
## 7187     .
## 23283    .
## 2512     .
## 1968     .
## 3460     .
## 928      .
## 8985     .
## 8317     .
## 10683    .
## 26873    .
## 10961    .
## 23560    .
## 51706    .
## 10555    .
## 1593     .
## 815      .
## 3823     .
## 883      .
## 8795     .
## 242      .
## 63917    .
## 8443     .
## 1029     .
## 9601     .
## 6608     .
## 9126     .
## 3363     .
## 635    -0.084546785
## 4825   -0.028206145
## 2979     .
## 2261     .
## 1465     .
## 3597     .
## 2590     .
## 55506    .
## 3613     .
## 56681    .
## 7076     .
## 23049    .
## 79671    .
## 4160     .
## 435      .
## 50613    .
## 5154     .
```

```
## 7991     .
## 1142     .
## 10095    .
## 8526     .
## 8837     .
## 23162    .
## 680      .
## 2876     .
## 2768     .
## 301      .
## 57292    .
## 3805     .
## 3803     .
## 5078     .
## 8027   -0.055209814
## 8878     .
## 467      .
## 3956     .
## 8945     .
## 25956    .
## 2012     .
## 4940     .
## 6579     .
## 6347     .
## 6282     .
## 5982     .
## 706      .
## 134      .
## 9519     .
## 5269     .
## 8737     .
## 130      .
## 5742     .
## 348995   .
## 307      .
## 8031     .
## 8643     .
## 5137    0.047616877
## 79717    .
## 55970    .
## 6251     .
## 10010   0.013500854
## 834      .
## 55746    .
## 9071     .
## 1030     .
## 8667     .
## 4616     .
## 302      .
## 1495     .
## 4035     .
## 2064     .
## 6189     .
## 3958     .
```

```
## 6777    .
## 2235    .
## 5260    .
## 54961   .
## 3646    .
## 3486    .
## 3693    .
## 5937    .
## 4301    .
## 7029    .
## 4982    .
## 55740   .
## 6609    .
## 5957    .
## 6774    .
## 5479    .
## 5799    .
## 4303    .
## 5322    .
## 1948    .
## 5872    .
## 80228   .
## 821     .
## 60489   .
## 1635    .
## 4000    .
## 5395    .
## 2260    .
## 2923    .
## 5832   -0.010516256
## 22925   .
## 833     .
## 3568    .
## 29978   .
## 23590   .
## 4804    .
## 650    -0.008886853
## 1717    .
## 5358    .
## 3381    .
## 9184    .
## 716     .
## 1647    .
## 5327    .
## 64581   .
## 84790   .
## 5158    .
## 6885    .
## 57819   .
## 51056   .
## 6720    .
## 7253    .
## 5329    .
## 11051   .
```

```
## 8717     .
## 1263     .
## 3757     0.067367967
## 4863     .
## 2318     .
## 92815    .
## 2242     .
## 23401   -0.041951307
## 2847     .
## 3556     .
## 7037     .
## 7434     .
## 2107     .
## 22934    .
## 373156   .
## 2254     .
## 132      .
## 27350    .
## 362      .
## 5575     .
## 6929     .
## 55844    .
## 1488     .
## 5025     .
## 2232     .
## 602      .
## 831      .
## 5238     .
## 7417     .
## 8074     .
## 10282    .
## 960      .
## 9104     .
## 1355     .
## 2766     .
## 2246     .
## 3423     0.060113280
## 838      .
## 3665     .
## 4763     .
## 5899     .
## 2792     .
## 27243    .
## 4493     .
## 3842     .
## 221830   .
## 6718     .
## 10130    .
## 2954     0.041359039
## 5795     .
## 648      .
## 5906     .
## 2747     .
## 3691     .
```

```
## 9180    .
## 55742   .
## 3034    .
## 8324    .
## 976     .
## 6156    .
## 6048    .
## 4255    .
## 8454    0.010574130
## 4046    .
## 51013   .
## 4129    .
## 2897    .
## 79728   .
## 6125    .
## 30      .
## 7494    .
## 10652   .
## 2817    .
## 26528   .
## 1643    .
## 5634    .
## 2645    .
## 3821    .
## 654     .
## 51513   .
## 2890    .
## 2331    .
## 5664    .
## 3488    .
## 5715    .
## 5332    .
## 5792    .
## 2014    .
## 761     .
## 10901   .
## 5971    .
## 9337    .
## 4084    .
## 23411  -0.054728602
## 54472   .
## 2100    .
## 6304    .
## 5328    .
## 5728    .
## 1027    .
## 55240   .
## 30833   .
## 4499    0.055507586
## 56895   .
## 3625    .
## 2035    .
## 5334    .
## 3309    .
```

```
## 10105   .
## 10588   .
## 3720    .
## 975     .
## 8453    .
## 2208    .
## 80319   .
## 3783    .
## 56034   .
## 6197    .
```

The selected features and their coefficients can by obtained:

```
nonZeroIdx <- which(coef.min[,1] != 0)
features <- rownames(coef.min)[nonZeroIdx]
features

##  [1] "22929" "5617"  "55341" "79723" "135"   "2053"  "2746"  "3485"  "635"   "4825"
## [11] "8027"  "5137"  "10010" "5832"  "650"   "3757"  "23401" "3423"  "2954"  "8454"
## [21] "23411" "4499"

features.coef <- coef.min[nonZeroIdx]
names(features.coef) <- features
features.coef

##        22929         5617        55341        79723          135         2053
## -0.020897935 -0.126328253  0.006056988 -0.016504110 -0.056788179  0.014389764
##         2746         3485          635         4825         8027         5137
## -0.011474410  0.065929675 -0.084546785 -0.028206145 -0.055209814  0.047616877
##        10010         5832          650         3757        23401         3423
##  0.013500854 -0.010516256 -0.008886853  0.067367967 -0.041951307  0.060113280
##         2954         8454        23411         4499
##  0.041359039  0.010574130 -0.054728602  0.055507586
```

The fitted RLassoCox model can by used to predicted survival risk of new patients:

```
lp <- predict.cvRLassoCox(object = cv.mod, newx = x.test,
                          s = "lambda.min")
lp

##                         1
## TCGA-02-0003  0.252281003
## TCGA-02-0011 -0.436949368
## TCGA-02-0014 -0.995169701
## TCGA-02-0024 -0.239595696
## TCGA-02-0055  0.071474306
## TCGA-02-0060 -0.360736704
## TCGA-06-0876 -0.065961334
## TCGA-06-5411  0.100059113
## TCGA-06-6390  0.376513608
## TCGA-19-5956  0.060821492
## TCGA-26-1442 -0.700211388
## TCGA-76-6282  0.155664820
## TCGA-06-6388  0.156024517
## TCGA-06-6700  0.082704758
```

```
## TCGA-26-6173 -0.040156916
## TCGA-26-6174 -0.149566729
## TCGA-74-6575  0.484281556
## TCGA-74-6577  0.525763613
## TCGA-74-6578  0.007548998
## TCGA-74-6581  0.324888178
## TCGA-76-6280  0.754944757
## TCGA-76-6286 -0.082245778
## TCGA-76-6657  0.524578516
## TCGA-76-6662  0.555978661
## TCGA-76-6664 -0.292625930
## TCGA-06-1084  0.239712644
## TCGA-12-1094 -0.103088391
## TCGA-12-1096 -0.388725786
## TCGA-12-1099  0.051802682
## TCGA-14-0736  0.145505972
## TCGA-14-0783 -0.108736284
## TCGA-14-1452 -0.013442523
## TCGA-26-1438 -0.085805469
## TCGA-26-1440  0.385229649
## TCGA-02-0071  0.292308469
## TCGA-02-0075  0.517931268
## TCGA-02-0080 -0.732568718
## TCGA-02-0083  0.377743827
## TCGA-02-0086  0.424971599
## TCGA-02-0099 -0.281152373
## TCGA-02-0107  0.039249428
## TCGA-02-0113  0.171228733
## TCGA-02-0116 -0.008497652
## TCGA-12-1600 -0.249009548
## TCGA-14-1458 -0.425255984
## TCGA-14-1821 -0.855506168
## TCGA-14-1827  0.234014704
## TCGA-27-1833 -0.164585648
## TCGA-27-1834  0.099416830
## TCGA-28-1757  0.239602225
## TCGA-02-2483 -0.625135072
## TCGA-02-2485  0.019610409
## TCGA-02-2486  0.429028506
## TCGA-06-2565  0.039163607
## TCGA-06-2566  0.122523143
## TCGA-27-1831  0.160742044
## TCGA-27-2524 -0.102853286
## TCGA-28-1756 -0.722575783
## TCGA-32-1982 -0.106181217
## TCGA-06-0130  0.086154241
## TCGA-06-0141  0.153624876
## TCGA-12-3653  0.017825574
## TCGA-14-3477 -0.196865367
## TCGA-32-2615  0.239068020
## TCGA-06-0154 -0.068996804
## TCGA-06-0168 -0.086409139
## TCGA-06-0184  0.189219979
## TCGA-06-0211  0.097159491
```

```
## TCGA-06-0214 -0.033073256
## TCGA-06-0221 -0.745897535
## TCGA-02-0325  0.547408994
## TCGA-02-0326  0.174854040
## TCGA-02-0338 -0.076475403
## TCGA-02-0432 -0.838758622
## TCGA-02-0451  0.202940730
## TCGA-06-0164 -0.015400069
## TCGA-08-0525  0.386861198
## TCGA-02-0015 -0.230604005
## TCGA-02-0016  0.029175934
## TCGA-02-0068 -0.064594271
## TCGA-02-0070  0.089438349
## TCGA-08-0347 -0.077289033
## TCGA-08-0350 -0.712272488
## TCGA-08-0351 -0.749903263
## TCGA-08-0356 -0.009218560
## TCGA-08-0380 -0.031081622
## TCGA-08-0389 -0.255074540
## TCGA-06-0939 -0.062484467
## TCGA-02-0084 -0.413313653
## TCGA-12-0616 -0.202956066
## TCGA-12-0620  0.002716557
## TCGA-06-5417 -1.003987837
## TCGA-28-5204  1.046083326
## TCGA-28-5213 -0.084996638
## TCGA-28-5215  0.102295777
## TCGA-76-4929  0.255713799
## TCGA-76-4931  0.306492175
## TCGA-76-4934 -0.308837884
## TCGA-76-4935 -0.081979086
## TCGA-06-0216 -0.187663624
## TCGA-06-0747  0.145434937
## TCGA-12-0656  0.098363463
## TCGA-12-0662  0.250645980
## TCGA-12-0688 -0.139330818
## TCGA-12-0780  0.137194654
```

4 SessionInfo()

```
sessionInfo()

## R Under development (unstable) (2020-10-17 r79346)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows >= 8 x64 (build 9200)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=Chinese (Simplified)_China.936  LC_CTYPE=Chinese (Simplified)_China.936
## [3] LC_MONETARY=Chinese (Simplified)_China.936 LC_NUMERIC=C
```

```
## [5] LC_TIME=Chinese (Simplified)_China.936
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] RLassoCox_0.99.0 glmnet_4.0-2     survival_3.2-7   Matrix_1.2-18    igraph_1.2.6
## [6] knitr_1.30
##
## loaded via a namespace (and not attached):
##  [1] codetools_0.2-16 lattice_0.20-41  foreach_1.5.1    grid_4.1.0       magrittr_1.5
##  [6] evaluate_0.14    highr_0.8        stringi_1.5.3    splines_4.1.0    iterators_1.0.13
## [11] tools_4.1.0      stringr_1.4.0    xfun_0.18        yaml_2.2.1       compiler_4.1.0
## [16] pkgconfig_2.0.3  shape_1.4.5
```

References

[1] Simon, Noah, Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Regularization Paths for Cox's Proportional Hazards Model via Coordinate Descent. Journal of Statistical Software. 2011, 39(5): 1-13.