BestSurvCutPoints

Meng-Ke Hsieh, Wen-Yi Chang, Chuang Chang

Modified by Payton Yau (tungon@gmail.com)

How to use our package?

To illustrate how to use our package, here we provide the following data that we generated. The data contained information of 30 individuals:

```
BMI=c(30,16,29,29,21,29,27,24,17,27,22,27,26,16,21,21,23,20,25,23,28,20,22,22,37,23,25,34,31,26)
event=c(1,1,1,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,1,0,0,1,1,0)
OS=c(138,92,64,15,62,235,214,197,41,33,257,115,123,44,154,71,61,182,75,214,25,217,113,200,175,117,166,0,57,186)
invasion=c(1.1,0.1,1.0,0.8,1.2,1.0,0.3,1.0,0.4,0.6,0.4,0.8,1.0,1.0,1.1,0.5,0.9,1.0,0.6,0.1,1.1,0.4,0.4,1.1,1.1,0.4,1.1,1.2,0.9,0.8)
LVSI=c(0,0,0,1,0,1,0,0,1,1,0,1,1,1,0,0,1,1,1,1,0,1,0,0,0,0,1,0,1)
```

BMI: body mass index
event: death (0=alive, 1=dead)
OS: time of observation in months (minimum of event time and right censoring time)
invasion: depth of tumor invasion into stroma
LVSI: lymphovascular space invasion (0=not seen, 1=present)

1. Survival data

First, we use function *findcutnum* to decide the optimal number of cutoff points based on AIC values

Function:

findcutnum(factor, outcome, datatype, nmin=20, segment=100)

factor	continuous risk factor which needs the optimal number of cutoffs
outcome	(event, time), event(1:event occurs;0:right censoring),time(minimum of event time and right censoring time), so outcome dimension is N by 2
nmin	the minimum number of each group, default is 20
segment	the total number of pieces and the default number is 100

How to run the function?

```
library(survival)
# Survival data
BMI=c(30,16,29,29,21,29,27,24,17,27,22,27,26,16,21,21,23,20,25,23,28,20,22,22,37,23,25,34,31,26)
event=c(1,1,1,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,1,0,0,1,1,0)
OS=c(138,92,64,15,62,235,214,197,41,33,257,115,123,44,154,71,61,182,75,214,25,217,113,200,175,117,166,0,57,186)

# run findcutnum function
findcutnum(factor=BMI, outcome=cbind(event,OS), datatype="survival", nmin=8, segment=100)
```

The outcome is as follows:

```
AIC have a minimum value When cutnum = 1
$min
[1] 50.52894 49.20829 51.10694      Inf      Inf
```

As the data here contained only 30 entries, we set the number of each segment to be 8 (nmin=8) while the number of segments was still the default 100 (segment=100). Here we wanted to find the optimal number of cutoff points for BMI in response to event death, so BMI was the variable factor and the outcome would be a 30 x 2 matrix that was generated with function cbind.

where 50.52894 is the AIC value when no cutoff point is made, 49.20829, one cutoff, and 51.10694, two. Since we set nmin to be 8 due to the small size of the example sample, 3 or more cutoff points were not realistic to make and the AIC values for 3 and 4 cutoffs were both Inf. The smallest AIC here is 49.20829, and it demonstrates that 1 cutoff point, if no clinical or other concerns are involved, would be optimal to divide these patients into low and high risk groups for death.

Now we need to use function *findcut* to find the optimal location for the above BMI data.

Function:

findcut(factor=NULL,outcome=NULL,cutnum=NA,datatype,nmin=20,segment=100)

factor	continuous risk factor which needs the optimal number of cutoffs
outcome	(event, time),event(1:event occurs;0:right censoring),time(minimum of event time and right censoring time),so outcome dimension is N by 2
cutnum	the number of cutoffs
nmin	the minimum number of each group and the default is 20
segment	the total number of pieces and the default is 100

How to run the function?

```
library(survival)
library(KMsurv)
library(xtable)
library(splines)
library("pROC")
library("aod")

# Survival data
BMI=c(30,16,29,29,21,29,27,24,17,27,22,27,26,16,21,21,23,20,25,23,28,20,22,22,37,23,25,34,31,26)
Event=c(1,1,1,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,1,0,0,1,1,0)
OS=c(138,92,64,15,62,235,214,197,41,33,257,115,123,44,154,71,61,182,75,214,25,217,113,200,175,117,166,0,57,186)

# run findcut function
findcut(factor=BMI, outcome=cbind (event,OS), cutnum=2, datatype = "survival", nmin=8, segment=100)
```

The outcome shows as follows:

allcut	Cut1	Cut2	Log.rank.test	Likelihood.ratio.test
1	21.00	25.00	0.13549462	0.14922833
2	21.00	25.21	0.13549462	0.14922833
3	21.00	25.42	0.13549462	0.14922833
4	21.00	25.63	0.13549462	0.14922833
5	21.00	25.84	0.13549462	0.14922833

Explanation:

allcut	(only the first five results are shown here)
Cut1 & Cut2	locations of the two cutoff points
Log.rank.test	when the cutoff points were placed at (Cut1,Cut2), log-rank test was used to test the significance of the points, i.e., p-value of the log-rank test.
Likelihood.ratio.test	when the cutoff points were placed at (Cut1,Cut2), likelihood ratio test was used to test the significance of the points, i.e., p-value of the likelihood ratio test (log test).

As the data here contained only 30 entries, we set the number of each segment to be 8 (nmin=8) while the number of segments was still the default, 100 (segment=100).

Here we wanted to find the optimal location of cutoff points for BMI in response to event death, so BMI was the variable factor and the outcome would be a 30 x 2 matrix that was generated with function cbind. Please note that in order to illustrate how to use this function to find the best locations of two cutoff points, we arbitrarily decided that we wanted 3 groups of BMI, or 2 cutoff points.

Final results:

	Cut1	Cut2
Log.rank.test	22.05	26
Likelihood.ratio.test	22.05	26

logranktest	the minimal p-value among allcut tested by the log rank test gave the best locations of cutoff points.
logtest_likelihood.ratio.test	the minimal p-value among allcut tested by the likelihood ratio test gave the best locations of cutoff points.

2. Dichotomized data

Again, we first use function findcutnum to help decide the optimal number of cutoff points based on AIC values.

```
library(survival)

invasion=c(1.1,0.1,1.0,0.8,1.2,1.0,0.3,1.0,0.4,0.6,0.4,0.8,1.0,1.0,1.1,0.5,0.9,1.0,0.6,0.1,1.1,0.4,0.4,1.1,1.1,0.4,1.1,1.2,0.9,0.8)
LVSI=c(0,0,0,1,0,1,0,0,1,1,0,1,1,1,0,0,0,1,1,1,1,0,1,0,0,0,0,1,0,1)

findcutnum(factor=invasion,outcome=LVSI,datatype="logistic",nmin=8,segment=100)
```

The outcome is as follows:

```
AIC have a minimum value When cutnum = 1
$min
[1] 45.43792 44.29145 46.29145      Inf      Inf
```

As the data here contained only 30 entries, we set the number of each segment to be 8 (nmin=8) while the number of segments was still the default, 100 (segment=100). Here we wanted to find the optimal number of cutoff points for invasion in response to presence of LVSI, so invasion was the variable factor and the outcome was LVSI.

where 45.43792 is the AIC value when no cutoff point is made, 44.29145, one cutoff, and 46.29145, two. Since we set nmin to be 8 due to the small size of the example sample, 3 or more cutoff points were not realistic to make and the AIC values for 3 and 4 cutoffs were both Inf. The smallest AIC here is 44.29145, and it demonstrates that 2 cutoff points, if no clinical or other concerns are involved, would be optimal.

Now we need to find the best locations of the two cutoff points using function findcut.

```
library(survival)
library(KMsurv)
library(xtable)
library(splines)
library("pROC")
library("aod")

# Survival data
invasion<-c(1.1,0.1,1.0,0.8,1.2,1.0,0.3,1.0,0.4,0.6,0.4,0.8,1.0,1.0,1.1,0.5,0.9,1.0,0.6,0.1,1.1,0.4,0.4,1.1,1.1,0.4,1.1,1.2,0.9,0.8)
LVSI<-c(0,0,0,1,0,1,0,0,1,1,0,1,1,1,0,0,0,1,1,1,1,0,1,0,0,0,0,1,0,1)

# run findcut function
findcut(factor=invasion,outcome=LVSI,cutnum=2,datatype="logistic",nmin=8,segment=100)
```

The outcome shows as follows:

allcut	*Cut1*	*Cut2*	*Likelihood.ratio.test*	*AUC*
1	*0.4*	*0.9*	*0.5587942*	*0.5*
2	*0.4*	*0.911*	*0.5587942*	*0.5*
3	*0.4*	*0.922*	*0.5587942*	*0.5*
4	*0.4*	*0.933*	*0.5587942*	*0.5*
5	*0.4*	*0.944*	*0.5587942*	*0.5*

Explanation:

allcut	(only the first five results are shown here)
Cut1 & Cut2	locations of the two cutoff points
Likelihood.ratio.test	when the cutoff points were placed at (Cut1,Cut2), likelihood ratio test was used to test the significance of the points, i.e., p-value of the likelihood ratio test (log test).
AUC	when the cutoff points were placed at (Cut1,Cut2), the maximal area under the ROC curve (AUC) was used to test the significance of the points.

As the data here contained only 30 entries, we set the number of each segment to be 8 (nmin=8) while the number of segments was still the default, 100 (segment=100). Here we wanted to find the optimal locations of cutoff points for invasion in response to presence of LVSI, so invasion was the variable factor and the outcome was LVSI. The number of cutoff points we needed here was 2, as illustrated above.

Final results:

	Cut1	**Cut2**
Likelihood.ratio.test	0.4	0.9
AUC	0.4	0.9

logtest_likelihood.ratio.test	the minimal p-value among allcut tested by the likelihood ratio test gave the best locations of cutoff points.
AUC	the maximal AUC among allcut rested by the AUC method gave the best locations of the cutoff points.