Package 'compound.Cox'

June 2, 2019

Type Package

Title Univariate Feature Selection and Compound Covariate for Predicting Survival

Version 3.17

Date 2019-6-1

Author Takeshi Emura, Hsuan-Yu Chen, Shigeyuki Matsui, Yi-Hau Chen

Maintainer Takeshi Emura <takeshiemura@gmail.com>

Description

Univariate feature selection and compound covariate methods under the Cox model with high-dimensional features (e.g., gene expressions).
Available are survival data for non-small-cell lung cancer patients with gene expressions (Chen et al 2007 New Engl J Med) <DOI:10.1056/NEJMoa060096>,
statistical methods in Emura et al (2012 PLoS ONE) <DOI:10.1371/journal.pone.0047627>,
Emura & Chen (2016 Stat Methods Med Res) <DOI:10.1177/0962280214533378>, and Emura et al. (2019)<DOI:10.1016/j.cmpb.2018.10.020>.
Algorithms for generating correlated gene expressions are also available.

License GPL-2

Depends numDeriv, survival

NeedsCompilation no

Repository CRAN

Date/Publication 2019-06-02 03:40:03 UTC

R topics documented:

compound.Cox-package 2
CG.Clayton 3
CG.Gumbel 4
cindex.CV 6
compound.reg 7
dependCox.reg 9
dependCox.reg.CV 11
Lung 13

PBC 16
uni.score 18
uni.selection 19
uni.Wald 20
X.pathway 21
X.tag 23

Index **24**

`compound.Cox-package` *Univariate Feature Selection and Compound Covariate for Predicting Survival*

Description

Univariate feature selection and compound covariate methods under the Cox model with high-dimensional features (e.g., gene expressions). Available are survival data for non-small-cell lung cancer patients with gene expressions (Chen et al 2007 New Engl J Med), statistical methods in Emura et al (2012 PLoS ONE), Emura & Chen (2016 Stat Methods Med Res), and Emura et al. (2019 Comput Methods Programs Biomed). Algorithms for generating correlated gene expressions are also available.

Details

Package:	compound.Cox
Type:	Package
Version:	3.17
Date:	2019-6-1
License:	GPL-2

Author(s)

Takeshi Emura, Hsuan-Yu Chen, Shigeyuki Matsui, Yi-Hau Chen; Maintainer: Takeshi Emura <takeshiemura@gmail.com>

References

Chen HY, Yu SL, Chen CH, et al (2007). A Five-gene Signature and Clinical Outcome in Non-small-cell Lung Cancer, N Engl J Med 356: 11-20.

Emura T, Chen YH, Chen HY (2012). Survival Prediction Based on Compound Covariate under Cox Proportional Hazard Models. PLoS ONE 7(10): e47627. doi:10.1371/journal.pone.0047627

Emura T, Chen YH (2016). Gene Selection for Survival Data Under Dependent Censoring: a Copula-based Approach, Stat Methods Med Res 25(No.6): 2840-57

Emura T, Matsui S, Chen HY (2019). compound.Cox: Univariate Feature Selection and Compound Covariate for Predicting Survival, Computer Methods and Programs in Biomedicine 168: 21-37

Matsui S (2006). Predicting Survival Outcomes Using Subsets of Significant Genes in Prognostic Marker Studies with Microarrays. BMC Bioinformatics: 7:156.

`CG.Clayton` *Copula-graphic estimator under the Clayton copula.*

Description

This function computes the copula-graphic (CG) estimator (Rivest & Wells 2001) under the Clayton copula.

Usage

```
CG.Clayton(t.vec, d.vec, alpha, S.plot = TRUE, S.col = "black")
```

Arguments

`t.vec`	Vector of survival times (time to either death or censoring)
`d.vec`	Vector of censoring indicators, 1=death, 0=censoring
`alpha`	Association parameter that is related to Kendall's tau through "tau= alpha/(alpha+2)"
`S.plot`	If TRUE, the survival curve is displayed
`S.col`	Color of the survival curve

Details

The computational formula of the CG estimator is available in Emura & Chen (2018). The outputs show the survival probabilities at given time points of "t.vec". The input requires to specify an association parameter "alpha" of the Clayton copula (alpha>0), where alpha=0 corresponds to the independence copula. Emura and Chen (2016, 2018) applied the CG estimator to assess survival prognosis for lung cancer patients.

Value

`tau`	Kendall's tau (=alpha/(alpha+2))
`time`	sort(t.vec)
`surv`	survival probability at "time"

Author(s)

Takeshi Emura

References

Emura T, Chen YH (2016). Gene Selection for Survival Data Under Dependent Censoring: a Copula-based Approach, Stat Methods Med Res 25(No.6): 2840-57.

Emura T, Chen YH (2018). Analysis of Survival Data with Dependent Censoring, Copula-Based Approaches, JSS Research Series in Statistics, Springer, Singapore.

Rivest LP, Wells MT (2001). A Martingale Approach to the Copula-graphic Estimator for the Survival Function under Dependent Censoring, J Multivar Anal; 79: 138-55.

Examples

```
## Example 1 (a toy example of n=8) ##
t.vec=c(1,3,5,4,7,8,10,13)
d.vec=c(1,0,0,1,1,0,1,0)
CG.Clayton(t.vec,d.vec,alpha=18,S.col="blue")
### CG.Clayton gives identical results with the Kaplan-Meier estimator with alpha=0 ###
CG.Clayton(t.vec,d.vec,alpha=0.000000001,S.plot=FALSE)$surv
survfit(Surv(t.vec,d.vec)~1)$surv

## Example 2 (Analysis of the lung cancer data) ##
data(Lung) # read the data
t.vec=Lung[,"t.vec"]
d.vec=Lung[,"d.vec"]
x.vec=Lung[,"MMP16"] # the gene associated with survival (Emura and Chen 2016, 2018) #
Poor=x.vec>median(x.vec) ## Indicator of poor survival
Good=x.vec<=median(x.vec) ## Indicator of good survival

par(mfrow=c(1,2))
###### Predicted survival curves via the CG estimator #####
t.good=t.vec[Good]
d.good=d.vec[Good]
CG.Clayton(t.good,d.good,alpha=18,S.plot=TRUE,S.col="blue")

t.poor=t.vec[Poor]
d.poor=d.vec[Poor]
CG.Clayton(t.poor,d.poor,alpha=18,S.plot=TRUE,S.col="red")
```

`CG.Gumbel` *Copula-graphic estimator under the Gumbel copula.*

Description

This function computes the copula-graphic (CG) estimator (Rivest & Wells 2001) under the Gumbel copula.

Usage

```
CG.Gumbel(t.vec, d.vec, alpha, S.plot = TRUE, S.col = "black")
```

Arguments

t.vec	Vector of survival times (time to either death or censoring)
d.vec	Vector of censoring indicators, 1=death, 0=censoring
alpha	Association parameter that is related to Kendall's tau through "tau= alpha/(alpha+1)"
S.plot	If TRUE, the survival curve is displayed
S.col	Color of the survival curve

Details

The computational formula of the CG estimator is available in Emura & Chen (2018). The outputs show the survival probabilities at given time points of "t.vec". The input requires to specify an association parameter "alpha" of the Gumbel copula (alpha>=0), where alpha=0 corresponds to the independence copula. Emura and Chen (2016, 2018) applied the CG estimator to assess survival prognosis for lung cancer patients.

Value

tau	Kendall's tau (=alpha/(alpha+1))
time	sort(t.vec)
surv	survival probability at "time"

Author(s)

Takeshi Emura

References

Emura T, Chen YH (2016). Gene Selection for Survival Data Under Dependent Censoring: a Copula-based Approach, Stat Methods Med Res 25(No.6): 2840-57.

Emura T, Chen YH (2018). Analysis of Survival Data with Dependent Censoring, Copula-Based Approaches, JSS Research Series in Statistics, Springer, Singapore.

Rivest LP, Wells MT (2001). A Martingale Approach to the Copula-graphic Estimator for the Survival Function under Dependent Censoring, J Multivar Anal; 79: 138-55.

Examples

```
## Example 1 (a toy example of n=8) ##
t.vec=c(1,3,5,4,7,8,10,13)
d.vec=c(1,0,0,1,1,0,1,0)
CG.Gumbel(t.vec,d.vec,alpha=9,S.col="blue")
### CG.Gumbel gives identical results with the Kaplan-Meier estimator with alpha=0 ###
CG.Gumbel(t.vec,d.vec,alpha=0,S.plot=FALSE)$surv
survfit(Surv(t.vec,d.vec)~1)$surv

## Example 2 (Analysis of the lung cancer data) ##
data(Lung) # read the data
t.vec=Lung[,"t.vec"]
```

```
d.vec=Lung[,"d.vec"]
x.vec=Lung[,"MMP16"] # the gene associated with survival (Emura and Chen 2016, 2018) #
Poor=x.vec>median(x.vec) ## Indicator of poor survival
Good=x.vec<=median(x.vec) ## Indicator of good survival

par(mfrow=c(1,2))
###### Predicted survival curves via the CG estimator #####
t.good=t.vec[Good]
d.good=d.vec[Good]
CG.Gumbel(t.good,d.good,alpha=9,S.plot=TRUE,S.col="blue")

t.poor=t.vec[Poor]
d.poor=d.vec[Poor]
CG.Gumbel(t.poor,d.poor,alpha=9,S.plot=TRUE,S.col="red")
```

`cindex.CV` *Cross-validated c-index for measuring the predictive accuracy of a prognostic index under a copula-based dependent censoring model.*

Description

This function calculates the cross-validated c-index (concordance index) for measuring the predictive accuracy of a prognostic index under a copula-based dependent censoring model. Here the prognostic index is calculated as a compound covariate predictor based on the univariate Cox regression estimates. The expression and details are given in Section 3.2 of Emura and Chen (2016). The association between survival time and censoring time is modeled via the Clayton copula.

Usage

```
cindex.CV(t.vec, d.vec, X.mat, alpha, K = 5)
```

Arguments

`t.vec`	Vector of survival times (time to death or time to censoring, whichever comes first)
`d.vec`	Vector of censoring indicators, 1=death, 0=censoring
`X.mat`	n by p matrix of covariates, where n is the sample size and p is the number of covariates
`alpha`	Association parameter of the Clayton copula; Kendall's tau = alpha/(alpha+2)
`K`	The number of cross-validation folds (K=5 is the defailt)

Details

Currently, only the Clayton copula is implemented for modeling association between survival time and censoring time. The Clayton model yields positive association between survival time and censoring time with the Kendall's tau being equal to alpha/(alpha+2), where alpha > 0. The independent copula corresponds to alpha = 0.

If the number of covariates p is large (e.g., p>=100), the computational time becomes very long. Pre-filtering for covariates is recommended to reduce p.

Value

concordant Cross-validated c-index

Author(s)

Takeshi Emura

References

Emura T, Chen YH (2016). Gene Selection for Survival Data Under Dependent Censoring: a Copula-based Approach, Stat Methods Med Res 25(No.6): 2840-57.

Examples

```
n=25 ### sample size ###
p=3  ### the number of covariates ###
set.seed(1)
T=rexp(n) ### survival time
U=rexp(n) ### censoring time
t.vec=pmin(T,U) ### minimum of survival time and censoring time
d.vec=as.numeric( c(T<=U) ) ### censoring indicator
X.mat=matrix(runif(n*p),n,p) ### covariates matrix

cindex.CV(t.vec,d.vec,X.mat,alpha=2) ### alpha=2 corresponds to Kendall's tau=0.5
```

---

compound.reg *Compound shrinkage estimation under the Cox model*

Description

This function implements the "compound shrinkage estimator" to calculate the regression coefficients of the Cox model, which was proposed by Emura, Chen & Chen (2012). The method is a variant of the Cox partial likelihood estimator such that the regression coefficients are mixed with the univariate Cox regression estimators. The resultant estimator is applicable even when the number of covariates is greater than the number of samples (the p>n setting). The standard errors (SEs) are calculated based on the asymptotic theory (see Emura et al., 2012).

Usage

```
compound.reg(t.vec, d.vec, X.mat, K = 5, delta_a = 0.025, a_0 = 0, var = FALSE,
plot=TRUE, randomize = FALSE, var.detail = FALSE)
```

Arguments

t.vec	Vector of survival times (time to either death or censoring)
d.vec	Vector of censoring indicators, 1=death, 0=censoring
X.mat	n by p matrix of covariates, where n is the sample size and p is the number of covariates
K	The number of cross validation folds, K=n corresponds to a leave-one-out cross validation (default=5)
delta_a	The step size for a grid search for the maximum of the cross-validated likelihood (default=0.025)
a_0	The starting value of a grid search for the maximum of the cross-validated likelihood (default=0)
var	If TRUE, the standard deviations and confidence intervals are given (default=FALSE, to reduce the computational cost)
plot	If TRUE, the cross validated likelihood curve and its maximized point are drawn
randomize	If TRUE, randomize the subject ID's so that the subjects in the cross validation folds are randomly chosen. Otherwise, the cross validation folds are constructed in the ascending sequence
var.detail	Detailed information about the covariance matrix, which is mainly used for theoretical purposes. Please consult Takeshi Emura for more details (default=FALSE)

Details

K=5 cross validation is recommended for computational efficiency, though the results appear to be robust against the choice of the number K. If the number of covariates is greater than 200, the computational time becomes very long. In such a case, the univariate pre-selection is recommended to reduce the number of covariates.

Value

a	An optimized value of the shrinkage parameter (0<=a<=1)
beta	Estimated regression coefficients
SE	Standard errors for estimated regression coefficients
Lower95CI	Lower ends of 95 percent confidence intervals (beta_hat-1.96*SE)
Upper95CI	Upper ends of 95 percent confidence intervals (beta_hat+1.96*SE)
Sigma	Covariance matrix for estimated regression coefficients
V	Estimates of the information matrix (-[Hessian of the loglikelihood]/n)
Hessian_CV	Second derivative of the cross-validated likelihood. Normally negative since the cross-validated curve is concave
h_dot	Derivative of Equation (8) of Emura et al. (2012) with respect to a shrinkage parameter "a"

Author(s)

Takeshi Emura & Yi-Hau Chen

References

Emura T, Chen Y-H, Chen H-Y (2012) Survival Prediction Based on Compound Covariate under Cox Proportional Hazard Models. PLoS ONE 7(10): e47627. doi:10.1371/journal.pone.0047627

Examples

```
### A simulation study ###
n=50 ### sample size
beta_true=c(1,1,0,0,0)
p=length(beta_true)
t.vec=d.vec=numeric(n)
X.mat=matrix(0,n,p)

set.seed(1)
for(i in 1:n){
  X.mat[i,]=rnorm(p,mean=0,sd=1)
  eta=sum( as.vector(X.mat[i,])*beta_true )
  T=rexp(1,rate=exp(eta))
  C=runif(1,min=0,max=5)
  t.vec[i]=min(T,C)
  d.vec[i]=(T<=C)
}
compound.reg(t.vec,d.vec,X.mat,delta_a=0.1)
### compare the estimates (beta) with the true value ###
beta_true

### Lung cancer data analysis (Emura et al. 2012 PLoS ONE) ###
data(Lung)
temp=Lung[,"train"]==TRUE
t.vec=Lung[temp,"t.vec"]
d.vec=Lung[temp,"d.vec"]
X.mat=as.matrix( Lung[temp,-c(1,2,3)] )
#compound.reg(t.vec=t.vec,d.vec=d.vec,X.mat=X.mat,delta_a=0.025) # time-consuming process
```

`dependCox.reg` *Univariate Cox regression under dependent censoring.*

Description

This function performs univariate Cox regression under dependent censoring, where dependence between survival time and censoring time is modeled via the Clayton copula (Emura and Chen 2016).

Usage

```
dependCox.reg(t.vec, d.vec, X.vec, alpha, var = TRUE, censor.reg=FALSE)
```

Arguments

t.vec	A vector of survival times (time-to-death or censoring)
d.vec	A vector of censoring indicators, 1=death, 0=censoring
X.vec	A vector of covariates (multiple covariates are not allowed)
alpha	An copula parameter (Kendall's tau = alpha/(alpha+2)
var	If TRUE, the standard deviations are given (use FALSE to reduce the computational cost)
censor.reg	If TRUE, show the fitted results for both survival and censoring models

Details

The Clayton model yields positive association between survival time and censoring time with Kendall's tau being equal to alpha/(alpha+2), where alpha > 0 is a copula parameter. The independence copula corresponds to alpha = 0.

Value

beta	The estimated regression coefficient
SE	The standard error for the estimated regression coefficient
Z	The Z-value for testing the null hypothesis of "beta=0" (the Wald test)
P	The P-value for testing the null hypothesis of "beta=0" (the Wald test)

Author(s)

Takeshi Emura

References

Emura T, Chen YH (2016). Gene Selection for Survival Data Under Dependent Censoring: a Copula-based Approach, Stat Methods Med Res 25(No.6): 2840-57.

Examples

```
### Joint Cox regression of survival and censoring ###
data(Lung)
t.vec=Lung[,"t.vec"]# death or censoring times #
d.vec=Lung[,"d.vec"]# censoring indicators #
# 16-gene prognostic index (Emura and Chen 2016; 2018) #
X.vec=0.51*Lung[,"ZNF264"]+0.50*Lung[,"MMP16"]+
  0.50*Lung[,"HGF"]-0.49*Lung[,"HCK"]+0.47*Lung[,"NF1"]+
  0.46*Lung[,"ERBB3"]+0.57*Lung[,"NR2F6"]+0.77*Lung[,"AXL"]+
  0.51*Lung[,"CDC23"]+0.92*Lung[,"DLG2"]-0.34*Lung[,"IGF2"]+
  0.54*Lung[,"RBBP6"]+0.51*Lung[,"COX11"]+
  0.40*Lung[,"DUSP6"]-0.37*Lung[,"ENG"]-0.41*Lung[,"IHPK1"]
dependCox.reg(t.vec,d.vec,X.vec,alpha=18,censor.reg=TRUE)

temp=c(Lung[,"train"]==TRUE)
t.vec=Lung[temp,"t.vec"]
```

```
d.vec=Lung[temp,"d.vec"]
dependCox.reg(t.vec,d.vec,Lung[temp,"ZNF264"],alpha=18)
# this reproduces Table 3 of Emura and Chen (2016) #

#### A simulation study under dependent censoring ####
beta_true=1.5 # true regression coefficient
alpha_true=2 # true copula parameter corresponding to Kendall's tau=0.5
n=150
t.vec=d.vec=X.vec=numeric(n)
set.seed(1)
for(i in 1:n){
  X.vec[i]=runif(1)
  eta=X.vec[i]*beta_true
  U=runif(1)
  V=runif(1)
  T=-1/exp(eta)*log(1-U) # Exp(eta) distribution
  W=(1-U)^(-alpha_true) # dependence produced by the Clayton copula
  C=1/alpha_true/exp(eta)*log(  1-W+W*(1-V)^(-alpha_true/(alpha_true+1))  ) # Exp(eta) distribution
  t.vec[i]=min(T,C)
  d.vec[i]=(T<=C)
}

dependCox.reg(t.vec,d.vec,X.vec,alpha=alpha_true,var=FALSE) # faster computation by "var=FALSE"
beta_true# the above estimate is close to the true value
coxph(Surv(t.vec,d.vec)~X.vec)$coef
# this estimate is biased for the true value due to dependent censoring
```

`dependCox.reg.CV` *Cox regression under dependent censoring.*

Description

This function performs estimation and significance testing for survival data under a copula-based dependent censoring model proposed in Emura and Chen (2016). The dependency between the failure and censoring times is modeled via the Clayton copula. The method is based on the semi-parametric maximum likelihood estimation, where the association parameter is estimated by maximizing the cross-validated c-index (see Emura and Chen 2016 for details).

Usage

```
dependCox.reg.CV(t.vec, d.vec, X.mat, K = 5, G = 20)
```

Arguments

`t.vec`	A vector of survival times (time-to-death or censoring)
`d.vec`	A vector of censoring indicators, 1=death, 0=censoring
`X.mat`	An (n*p) matrix of covariates, where n is the sample size and p is the number of covariates

K	The number of cross-validation folds
G	The number of grids to optimize c-index (c-index is computed for G different values of copula parameters)

Details

The Clayton model yields positive association between survival time and censoring time with Kendall's tau being equal to alpha/(alpha+2), where alpha > 0 is a copula parameter. The independence copula corresponds to alpha = 0.

If the number of covariates p is large (p>=100), the computational time becomes very long. We suggest using "uni.selection" to reduce the number such that p<100.

If the number of grids G is large, the computational time becomes very long. Please take 5<=G<=20.

Value

beta	The estimated regression coefficients
SE	The standard errors for the estimated regression coefficients
Z	The Z-values for testing the null hypothesis of "beta=0" (the Wald test)
P	The P-values for testing the null hypothesis of "beta=0" (the Wald test)
alpha	The estimated copula parameter by optimizing c-index
c_index	The optimized value of c_index

Author(s)

Takeshi Emura

References

Emura T, Chen YH (2016). Gene Selection for Survival Data Under Dependent Censoring: a Copula-based Approach, Stat Methods Med Res 25(No.6): 2840-57

Examples

```
### Reproduce Section 5 of Emura and Chen (2016) ###
data(Lung)
temp=Lung[,"train"]==TRUE
t.vec=Lung[temp,"t.vec"]
d.vec=Lung[temp,"d.vec"]
X.mat=as.matrix(Lung[temp,-c(1,2,3)])
#dependCox.reg.CV(t.vec,d.vec,X.mat,G=20) # time-consuming process #
```

Lung *Survival data for patients with non-small-cell lung cancer.*

Description

A subset of the lung cancer data (Chen et al. 2007) is given. The subset consists of 97 gene expressions from 125 patients with non-small-cell lung cancer. The 97 genes were selected with P-value<0.20 under univariate Cox regression analyses as previously done in Emura et al. (2012) and Emura and Chen (2016). The intensity of gene expression was transformed to an ordinal level using the quantile, i.e. if the intensity of gene expression was <=25th, >25th, >50th, or >75th percentile, it was coded as 1, 2, 3, or 4, respectively (Chen et al. 2007).

Usage

```
data("Lung")
```

Format

A data frame with 125 observations on the following 100 variables.

`t.vec` survival times (time to either death or censoring) in months

`d.vec` censoring indicators, 1=death, 0=censoring

`train` TRUE=training set, FALSE=testing set, as defined in Chen et al. (2007)

`VHL` gene expression, coded as 1, 2, 3, or 4

`IHPK1` gene expression, coded as 1, 2, 3, or 4

`HMMR` gene expression, coded as 1, 2, 3, or 4

`CMKOR1` gene expression, coded as 1, 2, 3, or 4

`PLAU` gene expression, coded as 1, 2, 3, or 4

`IGF2` gene expression, coded as 1, 2, 3, or 4

`FGB` gene expression, coded as 1, 2, 3, or 4

`MYBL2` gene expression, coded as 1, 2, 3, or 4

`ODC1` gene expression, coded as 1, 2, 3, or 4

`MTHFD2` gene expression, coded as 1, 2, 3, or 4

`GLIPR1` gene expression, coded as 1, 2, 3, or 4

`EZH2` gene expression, coded as 1, 2, 3, or 4

`HCK` gene expression, coded as 1, 2, 3, or 4

`CCNC` gene expression, coded as 1, 2, 3, or 4

`XRCC1` gene expression, coded as 1, 2, 3, or 4

`CYP1B1` gene expression, coded as 1, 2, 3, or 4

`CDC25A` gene expression, coded as 1, 2, 3, or 4

`CD44` gene expression, coded as 1, 2, 3, or 4

`LCK` gene expression, coded as 1, 2, 3, or 4

`MTHFS` gene expression, coded as 1, 2, 3, or 4

`PON3` gene expression, coded as 1, 2, 3, or 4

`PTPN6` gene expression, coded as 1, 2, 3, or 4

`KIDINS220` gene expression, coded as 1, 2, 3, or 4

`KLHL22` gene expression, coded as 1, 2, 3, or 4

`RBBP6` gene expression, coded as 1, 2, 3, or 4

`GABARAPL2` gene expression, coded as 1, 2, 3, or 4

`SEH1L` gene expression, coded as 1, 2, 3, or 4

`CITED2` gene expression, coded as 1, 2, 3, or 4

`BARD1` gene expression, coded as 1, 2, 3, or 4

`TLX1` gene expression, coded as 1, 2, 3, or 4

`CRMP1` gene expression, coded as 1, 2, 3, or 4

`CTNNA1` gene expression, coded as 1, 2, 3, or 4

`ANXA5` gene expression, coded as 1, 2, 3, or 4

`PTGS2` gene expression, coded as 1, 2, 3, or 4

`SMC4L1` gene expression, coded as 1, 2, 3, or 4

`LOC285086` gene expression, coded as 1, 2, 3, or 4

`ATP11B` gene expression, coded as 1, 2, 3, or 4

`CDK10` gene expression, coded as 1, 2, 3, or 4

`IRF4` gene expression, coded as 1, 2, 3, or 4

`MYH11` gene expression, coded as 1, 2, 3, or 4

`ME3` gene expression, coded as 1, 2, 3, or 4

`CCT6A` gene expression, coded as 1, 2, 3, or 4

`SNCG` gene expression, coded as 1, 2, 3, or 4

`MAK3` gene expression, coded as 1, 2, 3, or 4

`VCPIP1` gene expression, coded as 1, 2, 3, or 4

`JMJD1A` gene expression, coded as 1, 2, 3, or 4

`STAT2` gene expression, coded as 1, 2, 3, or 4

`DDX6` gene expression, coded as 1, 2, 3, or 4

`ERBB3` gene expression, coded as 1, 2, 3, or 4

`PAX2` gene expression, coded as 1, 2, 3, or 4

`PCTK2` gene expression, coded as 1, 2, 3, or 4

`NF1` gene expression, coded as 1, 2, 3, or 4

`DLG2` gene expression, coded as 1, 2, 3, or 4

`JMJD1A.1` gene expression, coded as 1, 2, 3, or 4

`SUCLA2` gene expression, coded as 1, 2, 3, or 4

`MMP16` gene expression, coded as 1, 2, 3, or 4

`AP3B2` gene expression, coded as 1, 2, 3, or 4

`HGF` gene expression, coded as 1, 2, 3, or 4

`MAP2K3` gene expression, coded as 1, 2, 3, or 4

`CPEB4` gene expression, coded as 1, 2, 3, or 4

`ZNF264` gene expression, coded as 1, 2, 3, or 4

`AXL` gene expression, coded as 1, 2, 3, or 4

`CDC23` gene expression, coded as 1, 2, 3, or 4

`MAST3` gene expression, coded as 1, 2, 3, or 4

`COX11` gene expression, coded as 1, 2, 3, or 4

`PRKAG2` gene expression, coded as 1, 2, 3, or 4

`MAN1B1` gene expression, coded as 1, 2, 3, or 4

`F8` gene expression, coded as 1, 2, 3, or 4

`RSU1` gene expression, coded as 1, 2, 3, or 4

`MMD` gene expression, coded as 1, 2, 3, or 4

`AK5` gene expression, coded as 1, 2, 3, or 4

`IDS` gene expression, coded as 1, 2, 3, or 4

`BNIP1` gene expression, coded as 1, 2, 3, or 4

`ENG` gene expression, coded as 1, 2, 3, or 4

`PCDHGC3` gene expression, coded as 1, 2, 3, or 4

`RALY` gene expression, coded as 1, 2, 3, or 4

`WDR33` gene expression, coded as 1, 2, 3, or 4

`RNF4` gene expression, coded as 1, 2, 3, or 4

`PRDX1` gene expression, coded as 1, 2, 3, or 4

`FXN` gene expression, coded as 1, 2, 3, or 4

`PTPRU` gene expression, coded as 1, 2, 3, or 4

`FRAP1` gene expression, coded as 1, 2, 3, or 4

`MMP7` gene expression, coded as 1, 2, 3, or 4

`CST3` gene expression, coded as 1, 2, 3, or 4

`TIMP2` gene expression, coded as 1, 2, 3, or 4

`TAL1` gene expression, coded as 1, 2, 3, or 4

`STAT1` gene expression, coded as 1, 2, 3, or 4

`CCND1` gene expression, coded as 1, 2, 3, or 4

`DUSP6` gene expression, coded as 1, 2, 3, or 4

`SNRPF` gene expression, coded as 1, 2, 3, or 4

`MMP13` gene expression, coded as 1, 2, 3, or 4

`NR2F6` gene expression, coded as 1, 2, 3, or 4

HOXA1 gene expression, coded as 1, 2, 3, or 4

RIPK1 gene expression, coded as 1, 2, 3, or 4

IL7R gene expression, coded as 1, 2, 3, or 4

SEC13L1 gene expression, coded as 1, 2, 3, or 4

RPL5 gene expression, coded as 1, 2, 3, or 4

Details

Survival data consisting of 125 patients.

Source

Chen HY, Yu SL, Chen CH, et al (2007). A Five-gene Signature and Clinical Outcome in Non-small-cell Lung Cancer, N Engl J Med 356: 11-20.

References

Chen HY, Yu SL, Chen CH, et al (2007). A Five-gene Signature and Clinical Outcome in Non-small-cell Lung Cancer, N Engl J Med 356: 11-20.

Emura T, Chen YH (2016). Gene Selection for Survival Data Under Dependent Censoring: a Copula-based Approach, Stat Methods Med Res 25(No.6): 2840-57

Examples

```
data(Lung)
Lung[1:3,] ## show the first 3 samples ##

## The five-gene signature in Chen et al. (2007) ##
temp=Lung[,"train"]==TRUE
t.vec=Lung[temp,"t.vec"]
d.vec=Lung[temp,"d.vec"]
coxph(Surv(t.vec,d.vec)~Lung[temp,"ERBB3"])
coxph(Surv(t.vec,d.vec)~Lung[temp,"LCK"])
coxph(Surv(t.vec,d.vec)~Lung[temp,"DUSP6"])
coxph(Surv(t.vec,d.vec)~Lung[temp,"STAT1"])
coxph(Surv(t.vec,d.vec)~Lung[temp,"MMD"])
```

PBC *Primary biliary cirrhosis (PBC) of the liver data*

Description

A subset of primary biliary cirrhosis (PBC) of the liver data in the book "Counting Process & Survival Analysis" by Fleming & Harrington (1991). This subset is used in Tibshirani (1997).

Usage

```
data(PBC)
```

Format

A data frame with 276 observations on the following 19 variables.

- `T` Survival times (either time to death or censoring) in days
- `d` Censoring indicator, 1=death, 0=censoring
- `trt` Treatment indicator, 1=treatment by D-penicillamine, 0=placebo
- `age` Age in years (days divided by 365.25)
- `sex` Sex, 0=male, 1=female
- `asc` Presence of ascites, 0=no, 1=yes
- `hep` Presence of hepatomegaly, 0=no, 1=yes
- `spi` Presence of spiders, 0=no, 1=yes
- `ede` Presence of edema, 0=no edema, 0.5=edema resolved by therapy, 1=edema not resolved by therapy
- `bil` log(bililubin, mg/dl)
- `cho` log(cholesterol, mg/dl)
- `alb` log(albumin, gm/dl)
- `cop` log(urine copper, mg/day)
- `alk` log(alkarine, U/liter)
- `SGO` log(SGOT, in U/ml)
- `tri` log(triglycerides, in mg/dl)
- `pla` log(platelet count, [the number of platelets per-cubic-milliliter of blood]/1000)
- `pro` log(prothrombin time, in seconds)
- `gra` Histologic stage of disease, graded 1, 2, 3, or 4

Details

Survival data consisting of 276 patients with 17 covariates. Among them, 111 patients died (d=1) while others were censored (d=0). The covariates consist of a treatment indicator (trt), age, sex, 5 categorical variables (ascites, hepatomegaly, spider, edema, and stage of disease) and 9 log-transformed continuous variables (bilirubin, cholesterol, albumin, urine copper, alkarine, SGOT, triglycerides, platelet count, and prothrombine).

Source

Fleming & Harrignton (1991); Tibshirani (1997)

References

Tibshirani R (1997), The Lasso method for variable selection in the Cox model, Statistics in Medicine, 385-395.

Examples

```
data(PBC)
PBC[1:5,] ### profiles for the first 5 patients ###
# See also Appendix D.1 of Fleming & Harrington, Counting Process & Survival Analysis (1991) #
```

`uni.score` *Univariate Cox score test*

Description

Univariate significance analyses via the score tests (Witten & Tibshirani 2010; Emura et al. 2018-) based on association between individual features and survival.

Usage

```
uni.score(t.vec, d.vec, X.mat, d0=0)
```

Arguments

`t.vec`	Vector of survival times (time to either death or censoring)
`d.vec`	Vector of censoring indicators, 1=death, 0=censoring
`X.mat`	n by p matrix of covariates, where n is the sample size and p is the number of covariates
`d0`	A positive constant to stabilize the variance (Witten & Tibshirani 2010)

Details

score test

Value

`beta`	Estimated regression coefficients (one-step estimator)
`Z`	Z-value for testing H_0: beta=0 (score test)
`P`	P-value for testing H_0: beta=0 (score test)

Author(s)

Takeshi Emura and Shigeyuki Matsui

References

Emura T, Matsui S, Chen HY (2018-). compound.Cox: Univariate Feature Selection and Compound Covariate for Predicting Survival, Computer Methods and Programs in Biomedicine, to appear.

Witten DM, Tibshirani R (2010) Survival analysis with high-dimensional covariates. Stat Method Med Res 19:29-51

Examples

```
data(Lung)
t.vec=Lung$t.vec[Lung$train==TRUE]
d.vec=Lung$d.vec[Lung$train==TRUE]
X.mat=Lung[Lung$train==TRUE,-c(1,2,3)]
uni.score(t.vec, d.vec, X.mat)
```

uni.selection *Univariate feature selection based on univariate significance tests*

Description

This function performs univariate feature selection using significance tests (Wald tests or score tests) based on association between individual features and survival. Features are selected if their P-values are less than a given threshold (P.value).

Usage

```
uni.selection(t.vec, d.vec, X.mat, P.value=0.001,K=10,score=TRUE,d0=0,
                        randomize=FALSE,CC.plot=FALSE,permutation=FALSE,M=200)
```

Arguments

t.vec	Vector of survival times (time to either death or censoring)
d.vec	Vector of censoring indicators (1=death, 0=censoring)
X.mat	n by p matrix of covariates, where n is the sample size and p is the number of covariates
P.value	A threshold for selecting features
K	The number of cross-validation folds
score	If TRUE, the score tests are used; if not, the Wald tests are used
d0	A positive constant to stabilize the variance of score statistics (Witten & Tibshirani 2010)
randomize	If TRUE, randomize patient ID's before cross-validation
CC.plot	If TRUE, the compound covariate (CC) predictors are plotted
permutation	If TRUE, the FDR is computed by a permutation method (Witten & Tibshirani 2010; Emura et al. 2018).
M	The number of permutations to calculate the FDR

Details

The cross-validated likelihood (CVL) value is computed for selected features (Matsui 2006; Emura et al. 2018-). A high CVL value corresponds to a better predictive ability of selected features. Hence, the CVL value can be used to find the optimal set of features. The CVL value is computed by a K-fold cross-validation, where the number K can be chosen by user. The false discovery rate (FDR) is also computed by a formula and a permutation test (if "permutation=TRUE"). The RCVL1 and RCVL2 are "re-substitution" CVL values and provide upper control limits for the CVL value. If the CVL value is less than RCVL1 and RCVL2 values, the CVL value would be in-control. On the other hand, if the CVL value exceeds either RCVL1 or RCVL2 value, then the CVL may be computed again after changing the sample allocation.

Value

gene	Gene symbols
beta	Estimated regression coefficients
Z	Z-values for significance tests
P	P-values for significance tests
CVL	The value of CVL, RCVL1, and RCVL2 (Emura et al. 2018-)
Genes	The number of genes, the number of selected genes, and the number of falsely selected genes
FDR	False discovery rate (by a formula or a permutation method)

Author(s)

Takeshi Emura

References

Emura T, Matsui S, Chen HY (2018-). compound.Cox: Univariate Feature Selection and Compound Covariate for Predicting Survival, Computer Methods and Programs in Biomedicine, to appear.

Matsui S (2006). Predicting Survival Outcomes Using Subsets of Significant Genes in Prognostic Marker Studies with Microarrays. BMC Bioinformatics: 7:156.

Witten DM, Tibshirani R (2010) Survival analysis with high-dimensional covariates. Stat Method Med Res 19:29-51

Examples

```
data(Lung)
t.vec=Lung$t.vec[Lung$train==TRUE]
d.vec=Lung$d.vec[Lung$train==TRUE]
X.mat=Lung[Lung$train==TRUE,-c(1,2,3)]
uni.selection(t.vec, d.vec, X.mat, P.value=0.05,K=5,score=FALSE)
## the outputs reproduce Table 3 of Emura and Chen (2016) ##
```

---

uni.Wald *Univariate Cox Wald test*

Description

Univariate significance analyses via the Wald tests (Witten & Tibshirani 2010; Emura et al. 2018-) based on association between individual features and survival.

Usage

```
uni.Wald(t.vec, d.vec, X.mat)
```

Arguments

t.vec	Vector of survival times (time to either death or censoring)
d.vec	Vector of censoring indicators, 1=death, 0=censoring
X.mat	n by p matrix of covariates, where n is the sample size and p is the number of covariates

Details

Wald test

Value

beta	Estimated regression coefficients
Z	Z-value for testing H_0: beta=0 (Wald test)
P	P-value for testing H_0: beta=0 (Wald test)

Author(s)

Takeshi Emura

References

Emura T, Matsui S, Chen HY (2018-). compound.Cox: Univariate Feature Selection and Compound Covariate for Predicting Survival, Computer Methods and Programs in Biomedicine, to appear.

Witten DM, Tibshirani R (2010) Survival analysis with high-dimensional covariates. Stat Method Med Res 19:29-51

Examples

```
data(Lung)
t.vec=Lung$t.vec[Lung$train==TRUE]
d.vec=Lung$d.vec[Lung$train==TRUE]
X.mat=Lung[Lung$train==TRUE,-c(1,2,3)]
uni.Wald(t.vec, d.vec, X.mat)
```

X.pathway *Generate a matrix of gene expressions in the presence of pathways*

Description

Generate a matrix of gene expressions in the presence of pathways (Scenario 2 of Emura et al. (2012)).

Usage

```
X.pathway(n,p,q1,q2)
```

Arguments

n	the number of individuals (sample size)
p	the number of genes
q1	the number of positive non-null genes
q2	the number of negative non-null genes

Details

n by p matrix of gene expressions are generated. Correlation between columns is introduced to reflect the presence of gene pathways. The distribution of each column is standardized to have mean=0 and SD=1. If two genes are correlated, the correlation is 0.5. Otherwise, the correlation is 0. Details are referred to p.4 of Emura et al. (2012). This deta generation scheme was used in the simulations of Emura et al. (2012), Emura and Chen (2016) and Emura et al. (2017).

Value

X	n by p matrix of gene expressions

Author(s)

Takeshi Emura & Yi-Hau Chen

References

Emura T, Chen YH, Chen HY (2012). Survival Prediction Based on Compound Covariate under Cox Proportional Hazard Models. PLoS ONE 7(10): e47627. doi:10.1371/journal.pone.0047627

Emura T, Chen YH (2016). Gene Selection for Survival Data Under Dependent Censoring: a Copula-based Approach, Stat Methods Med Res 25(No.6): 2840-57

Emura T, Nakatochi M, Matsui S, Michimae H, Rondeau V (2017) Personalized dynamic prediction of death according to tumour progression and high-dimensional genetic factors: meta-analysis with a joint model, Stat Methods Med Res, doi:10.1177/0962280216688032

Examples

```
X.mat=X.pathway(n=200,p=100,q1=10,q2=10)
round( colMeans(X.mat),3 ) ## mean ~ 0 ##
round( apply(X.mat, MARGIN=2, FUN=sd),3) ## SD ~ 1 ##
```

X.tag *Generate a matrix of gene expressions in the presence of tag genes*

Description

Generate a matrix of gene expressions in the presence of tag genes (Scenario 1 of Emura et al. (2012)).

Usage

```
X.tag(n, p, q, s = 1)
```

Arguments

n	the number of individuals (sample size)
p	the number of genes
q	the number of non-null genes
s	the number of null genes correlated with a non-null gene (tag)

Details

n by p matrix of gene expressions are generated. Correlation between columns is introduced to reflect the presence of tag genes. The distribution of each column is standardized to have mean=0 and SD=1. If two genes are correlated, the correlation is 0.5. Otherwise, the correlation is 0. Details are referred to p.4 of Emura et al. (2012). This deta generation scheme was used in the simulations of Emura et al. (2012) and Emura and Chen (2016).

Value

X	n by p matrix of gene expressions

Author(s)

Takeshi Emura & Yi-Hau Chen

References

Emura T, Chen YH, Chen HY (2012). Survival Prediction Based on Compound Covariate under Cox Proportional Hazard Models. PLoS ONE 7(10): e47627. doi:10.1371/journal.pone.0047627

Emura T, Chen YH (2016). Gene Selection for Survival Data Under Dependent Censoring: a Copula-based Approach, Stat Methods Med Res 25(No.6): 2840-57.

Examples

```
X.mat=X.tag(n=200,p=100,q=10,s=4)
round( colMeans(X.mat),3 ) ## mean ~ 0 ##
round( apply(X.mat, MARGIN=2, FUN=sd),3) ## SD ~ 1 ##
```

Index

∗Topic **PBC**
PBC, 16
∗Topic **Wald test**
uni.Wald, 20
∗Topic **c-index**
cindex.CV, 6
∗Topic **compound covariate**
compound.reg, 7
uni.score, 18
uni.selection, 19
uni.Wald, 20
∗Topic **copula-graphic estimator**
CG.Clayton, 3
CG.Gumbel, 4
∗Topic **copula**
CG.Clayton, 3
CG.Gumbel, 4
cindex.CV, 6
dependCox.reg, 9
dependCox.reg.CV, 11
∗Topic **cross-validated partial likelihood**
uni.selection, 19
∗Topic **cross-validation**
cindex.CV, 6
∗Topic **datasets**
Lung, 13
PBC, 16
∗Topic **dependent censoring**
CG.Clayton, 3
CG.Gumbel, 4
dependCox.reg, 9
dependCox.reg.CV, 11
∗Topic **false discovery rate**
uni.selection, 19
∗Topic **feature selection**
uni.score, 18
uni.selection, 19
uni.Wald, 20
∗Topic **gene expression**
Lung, 13
X.pathway, 21
X.tag, 23
∗Topic **gene pathway**
X.pathway, 21
∗Topic **lung cancer**
Lung, 13
∗Topic **package**
compound.Cox-package, 2
∗Topic **score test**
uni.score, 18
∗Topic **shrinkage estimation**
compound.reg, 7
∗Topic **tag gene**
X.tag, 23
∗Topic **univariate Cox regression**
dependCox.reg, 9
dependCox.reg.CV, 11
uni.score, 18
uni.selection, 19
uni.Wald, 20

CG.Clayton, 3
CG.Gumbel, 4
cindex.CV, 6
compound.Cox *(*compound.Cox-package*)*, 2
compound.Cox-package, 2
compound.reg, 7

dependCox.reg, 9
dependCox.reg.CV, 11

Lung, 13

PBC, 16

uni.score, 18
uni.selection, 19
uni.Wald, 20

X.pathway, 21
X.tag, 23