

# Package ‘survivalSL’

August 27, 2024

**Type** Package

**Title** Super Learner for Survival Prediction from Censored Data

**Version** 0.96

**Depends** R (>= 4.0.0), splines, survival

**Imports** date, graphics, MASS, glmnet, caret, flexsurv,  
randomForestSRC, hdnom, glmnetUtils, survivalPLANN, dplyr,  
rpart, methods

**Suggests** reticulate, survivalmodels

## Description

Several functions and S3 methods to construct a super learner in the presence of censored times-to-event and to evaluate its prognostic capacities.

**License** GPL (>= 2)

**LazyLoad** yes

**NeedsCompilation** no

**BugReports** <https://github.com/foucher-y/survivalSL/issues>

**Author** Yann Foucher [aut, cre] (<<https://orcid.org/0000-0003-0330-7457>>),  
Camille Sabathe [aut] (<<https://orcid.org/0000-0001-9353-691X>>)

**Maintainer** Yann Foucher <yohann.foucher@univ-poitiers.fr>

**Repository** CRAN

**Date/Publication** 2024-08-27 04:00:02 UTC

## Contents

dataDIVAT2 . . . . .	2
dataDIVAT3 . . . . .	3
dataOFSEP . . . . .	4
LIB_AFTgamma . . . . .	5
LIB_AFTggamma . . . . .	7
LIB_AFTllogis . . . . .	8
LIB_AFTweibull . . . . .	10
LIB_COXaic . . . . .	11

LIB_COXall . . . . .	13
LIB_COXen . . . . .	14
LIB_COXlasso . . . . .	16
LIB_COXridge . . . . .	18
LIB_PHexponential . . . . .	19
LIB_PHgompertz . . . . .	21
LIB_PHSpline . . . . .	22
LIB_PLANN . . . . .	24
LIB_RSF . . . . .	26
LIB_SNN . . . . .	27
metrics . . . . .	29
plot.libsl . . . . .	30
plot.rocrisca . . . . .	32
plot.sltime . . . . .	33
predict.libsl . . . . .	34
predict.sltime . . . . .	35
print.libsl . . . . .	36
print.sltime . . . . .	37
roc . . . . .	38
summary.libsl . . . . .	39
summary.sltime . . . . .	41
survivalSL . . . . .	42
tuneCOXaic . . . . .	45
tuneCOXen . . . . .	47
tuneCOXlasso . . . . .	48
tuneCOXridge . . . . .	50
tunePHspline . . . . .	51
tunePLANN . . . . .	53
tuneRSF . . . . .	55
tuneSNN . . . . .	56
<b>Index</b>	<b>58</b>

---

dataDIVAT2

*A Sample from the DIVAT Data Bank.*

---

## Description

A data frame with 1912 French kidney transplant recipients from the DIVAT cohort.

## Usage

```
data(dataDIVAT2)
```

**Format**

A data frame with the 4 following variables:

- age This numeric vector provides the age of the recipient at the transplantation (in years).
- h1a This numeric vector provides the indicator of transplantations with at least 4 HLA incompatibilities between the donor and the recipient (1 for high level and 0 otherwise).
- retransplant This numeric vector provides the indicator of re-transplantation (1 for more than one transplantation and 0 for first kidney transplantation).
- ecd The Expanded Criteria Donor (1 for transplantations from ECD and 0 otherwise). ECD are defined by widely accepted criteria, which includes donors older than 60 years of age or 50-59 years of age with two of the following characteristics: history of hypertension, cerebrovascular accident as the cause of death or terminal serum creatinine higher than 1.5 mg/dL.
- times This numeric vector is the follow up times of each patient.
- failures This numeric vector is the event indicator (0=right censored, 1=event). An event is considered when return in dialysis or patient death with functioning graft is observed.

**Source**

URL: [www.divat.fr](http://www.divat.fr)

**References**

Le Borgne F, Giraudeau B, Querard AH, Giral M and Foucher Y. Comparisons of the performances of different statistical tests for time-to-event analysis with confounding factors: practical illustrations in kidney transplantation. *Statistics in medicine*. 30;35(7):1103-16, 2016. <doi:10.1002/sim.6777>

**Examples**

```
data(dataDIVAT2)

# Compute the non-adjusted Hazard Ratio related to the ECD versus SCD
cox.ecd<-coxph(Surv(times, failures) ~ ecd, data=dataDIVAT2)
summary(cox.ecd) # Hazard Ratio = 1.97
```

---

dataDIVAT3

*A Sample from the DIVAT Data Bank.*

---

**Description**

A data frame with 4267 French kidney transplant recipients.

**Usage**

```
data(dataDIVAT3)
```

**Format**

A data frame with 4267 observations for the 8 following variables.

ageR This numeric vector represents the age of the recipient (in years)

sexer This numeric vector represents the gender of the recipient (1=men, 0=female)

year.tx This numeric vector represents the year of the transplantation

ante.diab This numeric vector represents the diabetes statute (1=yes, 0=no)

pra This numeric vector represents the pre-graft immunization using the panel reactive antibody (1=detectable, 0=undetectable)

ageD This numeric vector represents the age of the donor (in years)

death.time This numeric vector represents the follow up time in days (until death or censoring)

death This numeric vector represents the death indicator at the follow-up end (1=death, 0=alive)

**Source**

URL: [www.divat.fr](http://www.divat.fr)

**References**

Le Borgne et al. Standardized and weighted time-dependent ROC curves to evaluate the intrinsic prognostic capacities of a marker by taking into account confounding factors. Manuscript submitted. Stat Methods Med Res. 27(11):3397-3410, 2018. <doi: 10.1177/ 0962280217702416.>

**Examples**

```
data(dataDIVAT3)

### a short summary of the recipient age at the transplantation
summary(dataDIVAT3$ageR)

### Kaplan and Meier estimation of the recipient survival
plot(survfit(Surv(death.time/365.25, death) ~ 1, data = dataDIVAT3),
     xlab="Post transplantation time (in years)", ylab="Patient survival",
     mark.time=FALSE)
```

---

dataOFSEP

*A Simulated Sample from the OFSEP Cohort.*

---

**Description**

A data frame with 1300 simulated French patients with multiple sclerosis from the OFSEP cohort. The baseline is 1 year after the initiation of the first-line treatment.

**Usage**

```
data(dataOFSEP)
```

**Format**

A data frame with 1300 observations for the 3 following variables:

`time` This numeric vector represents the follow up time in years (until disease progression or censoring)

`event` This numeric vector represents the disease progression indicator at the follow-up end (1=progression, 0=censoring)

`age` This numeric vector represents the patient age (in years) at baseline.

`duration` This numeric vector represents the disease duration (in days) at baseline.

`period` This numeric vector represents the calendar period: 1 in-between 2014 and 2018, and 0 otherwise.

`gender` This numeric vector represents the gender: 1 for women.

`relapse` This numeric vector represents the diagnosis of at least one relapse since the treatment initiation : 1 if at least one event, and 0 otherwise.

`edss` This vector of character string represents the EDSS level: "miss" for missing, "low" for EDSS between 0 to 2, and "high" otherwise.

`t1` This vector of character string represents the new gadolinium-enhancing T1 lesion: "missing", "0" or "1+" for at least 1 lesion.

`t2` This vector of character string represents the new T2 lesions: "no" or "yes".

`rio` This numeric vector represents the modified Rio score.

**Examples**

```
data(dataOFSEP)

### Kaplan and Meier estimation of the disease progression free survival
plot(survfit(Surv(time, event) ~ 1, data = dataOFSEP),
     ylab="Disease progression free survival",
     xlab="Time after the first anniversary of the first-line treatment in years")
```

---

LIB\_AFTgamma

*Library of the Super Learner for an Accelerated Failure Time (AFT)  
Model with a Gamma Distribution*

---

**Description**

Fit an AFT parametric model with a gamma distribution.

**Usage**

```
LIB_AFTgamma(times, failures, group=NULL, cov.quantile=NULL, cov.quali=NULL, data)
```

**Arguments**

<code>times</code>	The name of the variable related the numeric vector with the follow-up times.
<code>failures</code>	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
<code>group</code>	The name of the variable related to the exposure/treatment. This variable shall have only two modalities encoded 0 for the untreated/unexposed patients and 1 for the treated/exposed ones. The default value is NULL: no specific exposure/treatment is considered. When a specific exposure/treatment is considered, it will be forced in the algorithm or related interactions will be tested when possible.
<code>cov.quant</code>	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
<code>cov.qual</code>	The name(s) of the variable(s) related to the possible qualitative covariates. These variables must be numeric with two levels: 0 and 1. A complete disjunctive form must be used for covariates with more levels.
<code>data</code>	A data frame for training the model in which to look for the variables related to the status of the follow-up time ( <code>times</code> ), the event ( <code>failures</code> ), the optional treatment/exposure ( <code>group</code> ) and the covariables included in the previous model ( <code>cov.quant</code> and <code>cov.qual</code> ).

**Details**

The model is obtained by using the `dist="gamma"` in the `flexsurvreg` package.

**Value**

<code>model</code>	The estimated model.
<code>group</code>	The name of the variable related to the exposure/treatment.
<code>cov.quant</code>	The name(s) of the variable(s) related to the possible quantitative covariates.
<code>cov.qual</code>	The name(s) of the variable(s) related to the possible qualitative covariates.
<code>data</code>	The data frame used for learning. The first column is entitled <code>times</code> and corresponds to the observed follow-up times. The second column is entitled <code>failures</code> and corresponds to the event indicators. The other columns correspond to the predictors.
<code>times</code>	A vector of numeric values with the times of the predictions.
<code>hazard</code>	A vector of numeric values with the values of the cumulative baseline hazard function at the prediction times.
<code>predictions</code>	A matrix with the predictions of survivals of each subject (lines) for each observed time (columns).

**References**

Jackson, C. (2016). `flexsurv`: A Platform for Parametric Survival Modeling in R. *Journal of Statistical Software*, 70(8), 1-33. doi:10.18637/jss.v070.i08

**Examples**

```

data(dataDIVAT2)

# The estimation of the model from the first 200 lignes
model <- LIB_AFTgamma(times="times", failures="failures", data=dataDIVAT2[1:200,],
  cov.quanti=c("age"), cov.quali=c("hla", "retransplant", "ecd"))

# The predicted survival of the first subject of the training sample
plot(y=model$predictions[1,], x=model$times, xlab="Time (years)",
  ylab="Predicted survival", col=1, type="l", lty=1, lwd=2, ylim=c(0,1))

```

---

LIB_AFTgamma	<i>Library of the Super Learner for an Accelerated Failure Time (AFT) Model with a Generalized Gamma Distribution</i>
--------------	-----------------------------------------------------------------------------------------------------------------------

---

**Description**

Fit an AFT parametric model with a generalized gamma distribution.

**Usage**

```
LIB_AFTgamma(times, failures, group=NULL, cov.quanti=NULL, cov.quali=NULL, data)
```

**Arguments**

<code>times</code>	The name of the variable related the numeric vector with the follow-up times.
<code>failures</code>	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
<code>group</code>	The name of the variable related to the exposure/treatment. This variable shall have only two modalities encoded 0 for the untreated/unexposed patients and 1 for the treated/exposed ones. The default value is NULL: no specific exposure/treatment is considered. When a specific exposure/treatment is considered, it will be forced in the algorithm or related interactions will be tested when possible.
<code>cov.quanti</code>	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
<code>cov.quali</code>	The name(s) of the variable(s) related to the possible qualitative covariates. These variables must be numeric with two levels: 0 and 1. A complete disjunctive form must be used for covariates with more levels.
<code>data</code>	A data frame for training the model in which to look for the variables related to the status of the follow-up time ( <code>times</code> ), the event ( <code>failures</code> ), the optional treatment/exposure ( <code>group</code> ) and the covariables included in the previous model ( <code>cov.quanti</code> and <code>cov.quali</code> ).

**Details**

The model is obtained by using the `dist="gengamma"` in the `flexsurvreg` package.

**Value**

model	The estimated model.
group	The name of the variable related to the exposure/treatment.
cov.quant	The name(s) of the variable(s) related to the possible quantitative covariates.
cov.quali	The name(s) of the variable(s) related to the possible qualitative covariates.
data	The data frame used for learning. The first column is entitled <code>times</code> and corresponds to the observed follow-up times. The second column is entitled <code>failures</code> and corresponds to the event indicators. The other columns correspond to the predictors.
times	A vector of numeric values with the times of the predictions.
hazard	A vector of numeric values with the values of the cumulative baseline hazard function at the prediction times.
predictions	A matrix with the predictions of survivals of each subject (lines) for each observed time (columns).

**References**

Jackson, C. (2016). `flexsurv`: A Platform for Parametric Survival Modeling in R. *Journal of Statistical Software*, 70(8), 1-33. doi:10.18637/jss.v070.i08

**Examples**

```
data(dataDIVAT2)

# The estimation of the model from the first 200 lignes
model <- LIB_AFTgamma(times="times", failures="failures", data=dataDIVAT2[1:200,],
  cov.quant=c("age"), cov.quali=c("hla", "retransplant", "ecd"))

# The predicted survival of the first subject of the training sample
plot(y=model$predictions[1,], x=model$times, xlab="Time (years)",
  ylab="Predicted survival", col=1, type="l", lty=1, lwd=2, ylim=c(0,1))
```

---

LIB_AFTllogis	<i>Library of the Super Learner for an Accelerated Failure Time (AFT) Model with a Log Logistic Distribution</i>
---------------	------------------------------------------------------------------------------------------------------------------

---

**Description**

Fit an AFT parametric model with a log logistic distribution.

**Usage**

```
LIB_AFTllogis(times, failures, group=NULL, cov.quant=NULL, cov.quali=NULL, data)
```



**Arguments**

<code>times</code>	The name of the variable related the numeric vector with the follow-up times.
<code>failures</code>	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
<code>group</code>	The name of the variable related to the exposure/treatment. This variable shall have only two modalities encoded 0 for the untreated/unexposed patients and 1 for the treated/exposed ones. The default value is NULL: no specific exposure/treatment is considered. When a specific exposure/treatment is considered, it will be forced in the algorithm or related interactions will be tested when possible.
<code>cov.quant</code>	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
<code>cov.qual</code>	The name(s) of the variable(s) related to the possible qualitative covariates. These variables must be numeric with two levels: 0 and 1. A complete disjunctive form must be used for covariates with more levels.
<code>data</code>	A data frame for training the model in which to look for the variables related to the status of the follow-up time ( <code>times</code> ), the event ( <code>failures</code> ), the optional treatment/exposure ( <code>group</code> ) and the covariables included in the previous model ( <code>cov.quant</code> and <code>cov.qual</code> ).

**Details**

The model is obtained by using the `dist="llogis"` in the `flexsurvreg` package.

**Value**

<code>model</code>	The estimated model.
<code>group</code>	The name of the variable related to the exposure/treatment.
<code>cov.quant</code>	The name(s) of the variable(s) related to the possible quantitative covariates.
<code>cov.qual</code>	The name(s) of the variable(s) related to the possible qualitative covariates.
<code>data</code>	The data frame used for learning. The first column is entitled <code>times</code> and corresponds to the observed follow-up times. The second column is entitled <code>failures</code> and corresponds to the event indicators. The other columns correspond to the predictors.
<code>times</code>	A vector of numeric values with the times of the predictions.
<code>hazard</code>	A vector of numeric values with the values of the cumulative baseline hazard function at the prediction times.
<code>predictions</code>	A matrix with the predictions of survivals of each subject (lines) for each observed time (columns).

**References**

Jackson, C. (2016). `flexsurv`: A Platform for Parametric Survival Modeling in R. *Journal of Statistical Software*, 70(8), 1-33. doi:10.18637/jss.v070.i08

**Examples**

```

data(dataDIVAT2)

# The estimation of the model from the first 200 lignes
model <- LIB_AFTllogis(times="times", failures="failures", data=dataDIVAT2[1:200,],
  cov.quant=c("age"), cov.quali=c("hla", "retransplant", "ecd"))

# The predicted survival of the first subject of the training sample
plot(y=model$predictions[1,], x=model$times, xlab="Time (years)",
  ylab="Predicted survival", col=1, type="l", lty=1, lwd=2, ylim=c(0,1))

```

---

LIB_AFTweibull	<i>Library of the Super Learner for an Accelerated Failure Time (AFT) Model with a Weibull Distribution</i>
----------------	-------------------------------------------------------------------------------------------------------------

---

**Description**

Fit an AFT parametric model with a Weibull distribution.

**Usage**

```
LIB_AFTweibull(times, failures, group=NULL, cov.quant=NULL, cov.quali=NULL, data)
```

**Arguments**

times	The name of the variable related the numeric vector with the follow-up times.
failures	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
group	The name of the variable related to the exposure/treatment. This variable shall have only two modalities encoded 0 for the untreated/unexposed patients and 1 for the treated/exposed ones. The default value is NULL: no specific exposure/treatment is considered. When a specific exposure/treatment is considered, it will be forced in the algorithm or related interactions will be tested when possible.
cov.quant	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
cov.quali	The name(s) of the variable(s) related to the possible qualitative covariates. These variables must be numeric with two levels: 0 and 1. A complete disjunctive form must be used for covariates with more levels.
data	A data frame for training the model in which to look for the variables related to the status of the follow-up time (times), the event (failures), the optional treatment/exposure (group) and the covariables included in the previous model (cov.quant and cov.quali).

**Details**

The model is obtained by using the `dist="weibull"` in the `flexsurvreg` package.

**Value**

model	The estimated model.
group	The name of the variable related to the exposure/treatment.
cov.quant	The name(s) of the variable(s) related to the possible quantitative covariates.
cov.quali	The name(s) of the variable(s) related to the possible qualitative covariates.
data	The data frame used for learning. The first column is entitled <code>times</code> and corresponds to the observed follow-up times. The second column is entitled <code>failures</code> and corresponds to the event indicators. The other columns correspond to the predictors.
times	A vector of numeric values with the times of the predictions.
hazard	A vector of numeric values with the values of the cumulative baseline hazard function at the prediction times.
predictions	A matrix with the predictions of survivals of each subject (lines) for each observed time (columns).

**References**

Jackson, C. (2016). `flexsurv`: A Platform for Parametric Survival Modeling in R. *Journal of Statistical Software*, 70(8), 1-33. doi:10.18637/jss.v070.i08

**Examples**

```
data(dataDIVAT2)

# The estimation of the model from the first 200 lignes
model <- LIB_AFTweibull(times="times", failures="failures", data=dataDIVAT2[1:200,],
  cov.quant=c("age"), cov.quali=c("hla", "retransplant", "ecd"))

# The predicted survival of the first subject of the training sample
plot(y=model$predictions[1,], x=model$times, xlab="Time (years)",
  ylab="Predicted survival", col=1, type="l", lty=1, lwd=2, ylim=c(0,1))
```

LIB\_COXaic

*Library of the Super Learner for a Cox Model with Selected Covariates*

**Description**

Fit a Cox regression for a selection of covariate.

**Usage**

```
LIB_COXaic(times, failures, group=NULL, cov.quant=NULL, cov.quali=NULL,
  data, final.model)
```

**Arguments**

<code>times</code>	The name of the variable related the numeric vector with the follow-up times.
<code>failures</code>	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
<code>group</code>	The name of the variable related to the exposure/treatment. This variable shall have only two modalities encoded 0 for the untreated/unexposed patients and 1 for the treated/exposed ones. The default value is NULL: no specific exposure/treatment is considered. When a specific exposure/treatment is considered, it will be forced in the algorithm or related interactions will be tested when possible.
<code>cov.quant</code>	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
<code>cov.qual</code>	The name(s) of the variable(s) related to the possible qualitative covariates. These variables must be numeric with two levels: 0 and 1. A complete disjunctive form must be used for covariates with more levels.
<code>data</code>	A data frame for training the model in which to look for the variables related to the status of the follow-up time ( <code>times</code> ), the event ( <code>failures</code> ), the optional treatment/exposure ( <code>group</code> ) and the covariates included in the previous model ( <code>cov.quant</code> and <code>cov.qual</code> )
<code>final.model</code>	The covariates to consider

**Value**

<code>model</code>	The estimated model.
<code>group</code>	The name of the variable related to the exposure/treatment.
<code>cov.quant</code>	The name(s) of the variable(s) related to the possible quantitative covariates.
<code>cov.qual</code>	The name(s) of the variable(s) related to the possible qualitative covariates.
<code>data</code>	The data frame used for learning. The first column is entitled <code>times</code> and corresponds to the observed follow-up times. The second column is entitled <code>failures</code> and corresponds to the event indicators. The other columns correspond to the predictors.
<code>times</code>	A vector of numeric values with the times of the predictions.
<code>hazard</code>	A vector of numeric values with the values of the cumulative baseline hazard function at the prediction times.
<code>predictions</code>	A matrix with the predictions of survivals of each subject (lines) for each observed time (columns).

**References**

Simon, N., Friedman, J., Hastie, T. and Tibshirani, R. (2011) Regularization Paths for Cox's Proportional Hazards Model via Coordinate Descent, *Journal of Statistical Software*, Vol. 39(5), 1-13, <https://www.jstatsoft.org/v39/i05/>

**Examples**

```

data(dataDIVAT2)

# The estimation of the model
model <- LIB_COXaic(times="times", failures="failures", data=dataDIVAT2,
  final.model=c("age"), cov.quantitative=c("age"), cov.qualitative=c("hla", "retransplant", "ecd"))

# The predicted survival of the first subject of the training sample
plot(y=model$predictions[1,], x=model$times, xlab="Time (years)", ylab="Predicted survival",
  col=1, type="l", lty=1, lwd=2, ylim=c(0,1))

```

LIB\_COXall

*Library of the Super Learner for Cox Regression***Description**

Fit a Cox regression for all covariates to be used in the super learner.

**Usage**

```
LIB_COXall(times, failures, group=NULL, cov.quantitative=NULL, cov.qualitative=NULL, data)
```

**Arguments**

<code>times</code>	The name of the variable related the numeric vector with the follow-up times.
<code>failures</code>	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
<code>group</code>	The name of the variable related to the exposure/treatment. This variable shall have only two modalities encoded 0 for the untreated/unexposed patients and 1 for the treated/exposed ones. The default value is NULL: no specific exposure/treatment is considered. When a specific exposure/treatment is considered, it will be forced in the algorithm or related interactions will be tested when possible.
<code>cov.quantitative</code>	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
<code>cov.qualitative</code>	The name(s) of the variable(s) related to the possible qualitative covariates. These variables must be numeric with two levels: 0 and 1. A complete disjunctive form must be used for covariates with more levels.
<code>data</code>	A data frame for training the model in which to look for the variables related to the status of the follow-up time ( <code>times</code> ), the event ( <code>failures</code> ), the optional treatment/exposure ( <code>group</code> ) and the covariables included in the previous model ( <code>cov.quantitative</code> and <code>cov.qualitative</code> ).

**Details**

The Cox regression is obtained by using the survival package.

**Value**

model	The estimated model.
group	The name of the variable related to the exposure/treatment.
cov.quant	The name(s) of the variable(s) related to the possible quantitative covariates.
cov.qual	The name(s) of the variable(s) related to the possible qualitative covariates.
data	The data frame used for learning. The first column is entitled <code>times</code> and corresponds to the observed follow-up times. The second column is entitled <code>failures</code> and corresponds to the event indicators. The other columns correspond to the predictors.
times	A vector of numeric values with the times of the predictions.
hazard	A vector of numeric values with the values of the cumulative baseline hazard function at the prediction times.
predictions	A matrix with the predictions of survivals of each subject (lines) for each observed time (columns).

**References**

Terry M. Therneau (2021). A Package for Survival Analysis in R. R package version 3.2-13, <https://CRAN.R-project.org/package=survival>.

**Examples**

```
data(dataDIVAT2)

# The estimation of the model
model <- LIB_COXall(times="times", failures="failures", data=dataDIVAT2,
  cov.quant=c("age"), cov.qual=c("hla", "retransplant", "ecd"))

# The predicted survival of the first subject of the training sample
plot(y=model$predictions[1,], x=model$times, xlab="Time (years)",
  ylab="Predicted survival", col=1, type="l", lty=1, lwd=2, ylim=c(0,1))
```

LIB\_COXen

*Library of the Super Learner for Elastic Net Cox Regression***Description**

Fit an elastic net Cox regression for fixed values of the regularization parameters.

**Usage**

```
LIB_COXen(times, failures, group=NULL, cov.quant=NULL, cov.qual=NULL,
  data, alpha, lambda)
```

**Arguments**

<code>times</code>	The name of the variable related the numeric vector with the follow-up times.
<code>failures</code>	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
<code>group</code>	The name of the variable related to the exposure/treatment. This variable shall have only two modalities encoded 0 for the untreated/unexposed patients and 1 for the treated/exposed ones. The default value is NULL: no specific exposure/treatment is considered. When a specific exposure/treatment is considered, it will be forced in the algorithm or related interactions will be tested when possible.
<code>cov.quant</code>	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
<code>cov.qual</code>	The name(s) of the variable(s) related to the possible qualitative covariates. These variables must be numeric with two levels: 0 and 1. A complete disjunctive form must be used for covariates with more levels.
<code>data</code>	A data frame for training the model in which to look for the variables related to the status of the follow-up time ( <code>times</code> ), the event ( <code>failures</code> ), the optional treatment/exposure ( <code>group</code> ) and the covariables included in the previous model ( <code>cov.quant</code> and <code>cov.qual</code> ).
<code>alpha</code>	The value of the regularization parameter alpha for penalizing the partial likelihood.
<code>lambda</code>	The value of the regularization parameter lambda for penalizing the partial likelihood.

**Details**

The elastic net Cox regression is obtained by using the `glmnet` package.

**Value**

<code>model</code>	The estimated model.
<code>group</code>	The name of the variable related to the exposure/treatment.
<code>cov.quant</code>	The name(s) of the variable(s) related to the possible quantitative covariates.
<code>cov.qual</code>	The name(s) of the variable(s) related to the possible qualitative covariates.
<code>data</code>	The data frame used for learning. The first column is entitled <code>times</code> and corresponds to the observed follow-up times. The second column is entitled <code>failures</code> and corresponds to the event indicators. The other columns correspond to the predictors.
<code>times</code>	A vector of numeric values with the times of the predictions.
<code>hazard</code>	A vector of numeric values with the values of the cumulative baseline hazard function at the prediction times.
<code>predictions</code>	A matrix with the predictions of survivals of each subject (lines) for each observed time (columns).

## References

Simon, N., Friedman, J., Hastie, T. and Tibshirani, R. (2011) Regularization Paths for Cox's Proportional Hazards Model via Coordinate Descent, Journal of Statistical Software, Vol. 39(5), 1-13, <https://www.jstatsoft.org/v39/i05/>

## Examples

```
data(dataDIVAT2)

# The estimation of the model
model <- LIB_COXen(times="times", failures="failures", data=dataDIVAT2,
  cov.quanti=c("age"), cov.quali=c("hla", "retransplant", "ecd"), lambda=.1, alpha=.1)

# The predicted survival of the first subject of the training sample
plot(y=model$predictions[1,], x=model$times, xlab="Time (years)",
  ylab="Predicted survival", col=1, type="l", lty=1, lwd=2, ylim=c(0,1))
```

---

 LIB\_COXlasso

---

*Library of the Super Learner for Lasso Cox Regression*


---

## Description

Fit a Lasso Cox regression for a fixed value of the regularization parameter.

## Usage

```
LIB_COXlasso(times, failures, group=NULL, cov.quanti=NULL, cov.quali=NULL,
  data, lambda)
```

## Arguments

times	The name of the variable related the numeric vector with the follow-up times.
failures	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
group	The name of the variable related to the exposure/treatment. This variable shall have only two modalities encoded 0 for the untreated/unexposed patients and 1 for the treated/exposed ones. The default value is NULL: no specific exposure/treatment is considered. When a specific exposure/treatment is considered, it will be forced in the algorithm or related interactions will be tested when possible.
cov.quanti	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
cov.quali	The name(s) of the variable(s) related to the possible qualitative covariates. These variables must be numeric with two levels: 0 and 1. A complete disjunctive form must be used for covariates with more levels.



data	A data frame for training the model in which to look for the variables related to the status of the follow-up time ( <code>times</code> ), the event ( <code>failures</code> ), the optional treatment/exposure ( <code>group</code> ) and the covariables included in the previous model ( <code>cov.quant</code> and <code>cov.qual</code> ).
lambda	The value of the regularization parameter lambda for penalizing the partial likelihood.

### Details

The Lasso Cox regression is obtained by using the `glmnet` package.

### Value

model	The estimated model.
group	The name of the variable related to the exposure/treatment.
cov.quant	The name(s) of the variable(s) related to the possible quantitative covariates.
cov.qual	The name(s) of the variable(s) related to the possible qualitative covariates.
data	The data frame used for learning. The first column is entitled <code>times</code> and corresponds to the observed follow-up times. The second column is entitled <code>failures</code> and corresponds to the event indicators. The other columns correspond to the predictors.
times	A vector of numeric values with the times of the predictions.
hazard	A vector of numeric values with the values of the cumulative baseline hazard function at the prediction times.
predictions	A matrix with the predictions of survivals of each subject (lines) for each observed time (columns).

### References

Simon, N., Friedman, J., Hastie, T. and Tibshirani, R. (2011) Regularization Paths for Cox's Proportional Hazards Model via Coordinate Descent, *Journal of Statistical Software*, Vol. 39(5), 1-13, <https://www.jstatsoft.org/v39/i05/>

### Examples

```
data(dataDIVAT2)

# The estimation of the model
model <- LIB_COXlasso(times="times", failures="failures", data=dataDIVAT2,
  cov.quant=c("age"), cov.qual=c("hla", "retransplant", "ecd"), lambda=1)

# The predicted survival of the first subject of the training sample
plot(y=model$predictions[1,], x=model$times, xlab="Time (years)",
  ylab="Predicted survival", col=1, type="l", lty=1, lwd=2, ylim=c(0,1))
```

LIB\_COXridge

*Library of the Super Learner for Ridge Cox Regression***Description**

Fit a ridge Cox regression for a fixed value of the regularization parameter.

**Usage**

```
LIB_COXridge(times, failures, group=NULL, cov.quantitative=NULL, cov.qualitative=NULL,
data, lambda)
```

**Arguments**

<code>times</code>	The name of the variable related the numeric vector with the follow-up times.
<code>failures</code>	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
<code>group</code>	The name of the variable related to the exposure/treatment. This variable shall have only two modalities encoded 0 for the untreated/unexposed patients and 1 for the treated/exposed ones. The default value is NULL: no specific exposure/treatment is considered. When a specific exposure/treatment is considered, it will be forced in the algorithm or related interactions will be tested when possible.
<code>cov.quantitative</code>	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
<code>cov.qualitative</code>	The name(s) of the variable(s) related to the possible qualitative covariates. These variables must be numeric with two levels: 0 and 1. A complete disjunctive form must be used for covariates with more levels.
<code>data</code>	A data frame for training the model in which to look for the variables related to the status of the follow-up time ( <code>times</code> ), the event ( <code>failures</code> ), the optional treatment/exposure ( <code>group</code> ) and the covariables included in the previous model ( <code>cov.quantitative</code> and <code>cov.qualitative</code> ).
<code>lambda</code>	The value of the regularization parameter <code>lambda</code> for penalizing the partial likelihood.

**Details**

The ridge Cox regression is obtained by using the `glmnet` package.

**Value**

<code>model</code>	The estimated model.
<code>group</code>	The name of the variable related to the exposure/treatment.
<code>cov.quantitative</code>	The name(s) of the variable(s) related to the possible quantitative covariates.
<code>cov.qualitative</code>	The name(s) of the variable(s) related to the possible qualitative covariates.

data	The data frame used for learning. The first column is entitled <code>times</code> and corresponds to the observed follow-up times. The second column is entitled <code>failures</code> and corresponds to the event indicators. The other columns correspond to the predictors.
times	A vector of numeric values with the times of the predictions.
hazard	A vector of numeric values with the values of the cumulative baseline hazard function at the prediction times.
predictions	A matrix with the predictions of survivals of each subject (lines) for each observed time (columns).

## References

Simon, N., Friedman, J., Hastie, T. and Tibshirani, R. (2011) Regularization Paths for Cox's Proportional Hazards Model via Coordinate Descent, *Journal of Statistical Software*, Vol. 39(5), 1-13, <https://www.jstatsoft.org/v39/i05/>

## Examples

```
data(dataDIVAT2)

# The estimation of the model
model <- LIB_COXridge(times="times", failures="failures", data=dataDIVAT2,
  cov.quant=c("age"), cov.quali=c("hla", "retransplant", "ecd"), lambda=1)

# The predicted survival of the first subject of the training sample
plot(y=model$predictions[1,], x=model$times, xlab="Time (years)", ylab="Predicted survival",
  col=1, type="l", lty=1, lwd=2, ylim=c(0,1))
```

---

LIB\_PHexponential      *Library of the Super Learner for a Proportional Hazards (PH) Model with an Exponential Distribution*

---

## Description

Fit a PH model with an Exponential distribution.

## Usage

```
LIB_PHexponential(times, failures, group=NULL, cov.quant=NULL,
  cov.quali=NULL, data)
```

## Arguments

times	The name of the variable related the numeric vector with the follow-up times.
failures	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).

<code>group</code>	The name of the variable related to the exposure/treatment. This variable shall have only two modalities encoded 0 for the untreated/unexposed patients and 1 for the treated/exposed ones. The default value is NULL: no specific exposure/treatment is considered. When a specific exposure/treatment is considered, it will be forced in the algorithm or related interactions will be tested when possible.
<code>cov.quant</code>	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
<code>cov.qual</code>	The name(s) of the variable(s) related to the possible qualitative covariates. These variables must be numeric with two levels: 0 and 1. A complete disjunctive form must be used for covariates with more levels.
<code>data</code>	A data frame for training the model in which to look for the variables related to the status of the follow-up time ( <code>times</code> ), the event ( <code>failures</code> ), the optional treatment/exposure ( <code>group</code> ) and the covariables included in the previous model ( <code>cov.quant</code> and <code>cov.qual</code> ).

### Details

The model is obtained by using the `dist="exp"` in the `flexsurvreg` package.

### Value

<code>model</code>	The estimated model.
<code>group</code>	The name of the variable related to the exposure/treatment.
<code>cov.quant</code>	The name(s) of the variable(s) related to the possible quantitative covariates.
<code>cov.qual</code>	The name(s) of the variable(s) related to the possible qualitative covariates.
<code>data</code>	The data frame used for learning. The first column is entitled <code>times</code> and corresponds to the observed follow-up times. The second column is entitled <code>failures</code> and corresponds to the event indicators. The other columns correspond to the predictors.
<code>times</code>	A vector of numeric values with the times of the predictions.
<code>hazard</code>	A vector of numeric values with the values of the cumulative baseline hazard function at the prediction times.
<code>predictions</code>	A matrix with the predictions of survivals of each subject (lines) for each observed time (columns).

### References

Jackson, C. (2016). `flexsurv`: A Platform for Parametric Survival Modeling in R. *Journal of Statistical Software*, 70(8), 1-33. doi:10.18637/jss.v070.i08

### Examples

```
data(dataDIVAT2)

# The estimation of the model from the first 200 lignes
model <- LIB_PHexponential(times="times", failures="failures", data=dataDIVAT2[1:200,],
```

```

cov.quanti=c("age"), cov.quali=c("hla", "retransplant", "ecd"))

# The predicted survival of the first subject of the training sample
plot(y=model$predictions[1,], x=model$times, xlab="Time (years)",
ylab="Predicted survival", col=1, type="l", lty=1, lwd=2, ylim=c(0,1))

```

---

LIB_PHgompertz	<i>Library of the Super Learner for an Proportional Hazards (PH) Model with a Gompertz Distribution</i>
----------------	---------------------------------------------------------------------------------------------------------

---

### Description

Fit a PH parametric model with a Gompertz distribution.

### Usage

```
LIB_PHgompertz(times, failures, group=NULL, cov.quanti=NULL,
cov.quali=NULL, data)
```

### Arguments

times	The name of the variable related the numeric vector with the follow-up times.
failures	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
group	The name of the variable related to the exposure/treatment. This variable shall have only two modalities encoded 0 for the untreated/unexposed patients and 1 for the treated/exposed ones. The default value is NULL: no specific exposure/treatment is considered. When a specific exposure/treatment is considered, it will be forced in the algorithm or related interactions will be tested when possible.
cov.quanti	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
cov.quali	The name(s) of the variable(s) related to the possible qualitative covariates. These variables must be numeric with two levels: 0 and 1. A complete disjunctive form must be used for covariates with more levels.
data	A data frame for training the model in which to look for the variables related to the status of the follow-up time (times), the event (failures), the optional treatment/exposure (group) and the covariables included in the previous model (cov.quanti and cov.quali).

### Details

The model is obtained by using the `dist="gompertz"` in the `flexsurvreg` package.

**Value**

model	The estimated model.
group	The name of the variable related to the exposure/treatment.
cov.quant	The name(s) of the variable(s) related to the possible quantitative covariates.
cov.quali	The name(s) of the variable(s) related to the possible qualitative covariates.
data	The data frame used for learning. The first column is entitled <code>times</code> and corresponds to the observed follow-up times. The second column is entitled <code>failures</code> and corresponds to the event indicators. The other columns correspond to the predictors.
times	A vector of numeric values with the times of the predictions.
hazard	A vector of numeric values with the values of the cumulative baseline hazard function at the prediction times.
predictions	A matrix with the predictions of survivals of each subject (lines) for each observed time (columns).

**References**

Jackson, C. (2016). `flexsurv`: A Platform for Parametric Survival Modeling in R. *Journal of Statistical Software*, 70(8), 1-33. doi:10.18637/jss.v070.i08

**Examples**

```
data(dataDIVAT2)

# The estimation of the model from the first 200 lignes
model <- LIB_PHgompertz(times="times", failures="failures", data=dataDIVAT2[1:200,],
  cov.quant=c("age"), cov.quali=c("hla", "retransplant", "ecd"))

# The predicted survival of the first subject of the training sample
plot(y=model$predictions[1,], x=model$times, xlab="Time (years)",
  ylab="Predicted survival", col=1, type="l", lty=1, lwd=2, ylim=c(0,1))
```

---

LIB_PHspline	<i>Library of the Super Learner for an Survival Regression using the Royston/Parmar Spline Model</i>
--------------	------------------------------------------------------------------------------------------------------

---

**Description**

Fit an PH model with a survival function is modelled as a natural cubic spline function.

**Usage**

```
LIB_PHspline(times, failures, group=NULL, cov.quant=NULL, cov.quali=NULL,
  data, k)
```

**Arguments**

<code>times</code>	The name of the variable related the numeric vector with the follow-up times.
<code>failures</code>	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
<code>group</code>	The name of the variable related to the exposure/treatment. This variable shall have only two modalities encoded 0 for the untreated/unexposed patients and 1 for the treated/exposed ones. The default value is NULL: no specific exposure/treatment is considered. When a specific exposure/treatment is considered, it will be forced in the algorithm or related interactions will be tested when possible.
<code>cov.quant</code>	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
<code>cov.qual</code>	The name(s) of the variable(s) related to the possible qualitative covariates. These variables must be numeric with two levels: 0 and 1. A complete disjunctive form must be used for covariates with more levels.
<code>data</code>	A data frame for training the model in which to look for the variables related to the status of the follow-up time ( <code>times</code> ), the event ( <code>failures</code> ), the optional treatment/exposure ( <code>group</code> ) and the covariables included in the previous model ( <code>cov.quant</code> and <code>cov.qual</code> ).
<code>k</code>	Number of knots.

**Details**

The model is obtained by using the `scale="hazard"` in the `flexsurvreg` package.

**Value**

<code>model</code>	The estimated model.
<code>group</code>	The name of the variable related to the exposure/treatment.
<code>cov.quant</code>	The name(s) of the variable(s) related to the possible quantitative covariates.
<code>cov.qual</code>	The name(s) of the variable(s) related to the possible qualitative covariates.
<code>data</code>	The data frame used for learning. The first column is entitled <code>times</code> and corresponds to the observed follow-up times. The second column is entitled <code>failures</code> and corresponds to the event indicators. The other columns correspond to the predictors.
<code>times</code>	A vector of numeric values with the times of the predictions.
<code>hazard</code>	A vector of numeric values with the values of the cumulative baseline hazard function at the prediction times.
<code>predictions</code>	A matrix with the predictions of survivals of each subject (lines) for each observed time (columns).

**References**

Jackson, C. (2016). `flexsurv`: A Platform for Parametric Survival Modeling in R. *Journal of Statistical Software*, 70(8), 1-33. doi:10.18637/jss.v070.i08

**Examples**

```

data(dataDIVAT2)

# The estimation of the model from the first 200 lignes with two knots
model <- LIB_PHspline(times="times", failures="failures", data=dataDIVAT2[1:200,],
  cov.quanti=c("age"), cov.quali=c("hla", "retransplant", "ecd"), k=2)

# The predicted survival of the first subject of the training sample
plot(y=model$predictions[1,], x=model$times, xlab="Time (years)",
  ylab="Predicted survival", col=1, type="l", lty=1, lwd=2, ylim=c(0,1))

```

LIB\_PLANN

*Library of the Super Learner for Survival Neural Network Based on the PLANN Method*

**Description**

Fit a neural network based on the partial logistic regression.

**Usage**

```
LIB_PLANN(times, failures, group=NULL, cov.quanti=NULL, cov.quali=NULL,
  data, inter, size, decay, maxit, MaxNWts)
```

**Arguments**

<code>times</code>	The name of the variable related the numeric vector with the follow-up times.
<code>failures</code>	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
<code>group</code>	The name of the variable related to the exposure/treatment. This variable shall have only two modalities encoded 0 for the untreated/unexposed patients and 1 for the treated/exposed ones. The default value is NULL: no specific exposure/treatment is considered. When a specific exposure/treatment is considered, it will be forced in the algorithm or related interactions will be tested when possible.
<code>cov.quanti</code>	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
<code>cov.quali</code>	The name(s) of the variable(s) related to the possible qualitative covariates. These variables must be numeric with two levels: 0 and 1. A complete disjunctive form must be used for covariates with more levels.
<code>data</code>	A data frame for training the model in which to look for the variables related to the status of the follow-up time ( <code>times</code> ), the event ( <code>failures</code> ), the optional treatment/exposure ( <code>group</code> ) and the covariables included in the previous model ( <code>cov.quanti</code> and <code>cov.quali</code> ).
<code>inter</code>	The length of the intervals.



size	The number of units in the hidden layer.
decay	The parameter for weight decay.
maxit	The maximum number of iterations.
MaxNWts	The maximum allowable number of weights.

### Details

This function is based is based on the survivalPLANN from the related package.

### Value

model	The estimated model.
group	The name of the variable related to the exposure/treatment.
cov.quant	The name(s) of the variable(s) related to the possible quantitative covariates.
cov.quali	The name(s) of the variable(s) related to the possible qualitative covariates.
data	The data frame used for learning. The first column is entitled times and corresponds to the observed follow-up times. The second column is entitled failures and corresponds to the event indicators. The other columns correspond to the predictors.
times	A vector of numeric values with the times of the predictions.
predictions	A matrix with the predictions of survivals of each subject (lines) for each observed time (columns).

### References

Biganzoli E, Boracchi P, Mariani L, and et al. Feed forward neural networks for the analysis of censored survival data: a partial logistic regression approach. Stat Med, 17:1169-86, 1998.

### Examples

```
data(dataDIVAT2)

# The neural network based from the first 300 individuals of the data base

model <- LIB_PLANN(times="times", failures="failures", data=dataDIVAT2[1:300,],
  cov.quant=c("age"), cov.quali=c("hla", "retransplant", "ecd"),
  inter=0.5, size=32, decay=0.01, maxit=100, MaxNWts=10000)

# The predicted survival of the first subject of the training sample
plot(y=model$predictions[1,], x=model$times, xlab="Time (years)",
  ylab="Predicted survival", col=1, type="l", lty=1, lwd=2, ylim=c(0,1))
```

**Description**

Fit survival random forest tree for given values of the regularization parameters.

**Usage**

```
LIB_RSF(times, failures, group=NULL, cov.quant=NULL, cov.quali=NULL,
data, nodesize, mtry, ntree)
```

**Arguments**

<code>times</code>	The name of the variable related the numeric vector with the follow-up times.
<code>failures</code>	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
<code>group</code>	The name of the variable related to the exposure/treatment. This variable shall have only two modalities encoded 0 for the untreated/unexposed patients and 1 for the treated/exposed ones. The default value is NULL: no specific exposure/treatment is considered. When a specific exposure/treatment is considered, it will be forced in the algorithm or related interactions will be tested when possible.
<code>cov.quant</code>	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
<code>cov.quali</code>	The name(s) of the variable(s) related to the possible qualitative covariates. These variables must be numeric with two levels: 0 and 1. A complete disjunctive form must be used for covariates with more levels.
<code>data</code>	A data frame for training the model in which to look for the variables related to the status of the follow-up time ( <code>times</code> ), the event ( <code>failures</code> ), the optional treatment/exposure ( <code>group</code> ) and the covariables included in the previous model ( <code>cov.quant</code> and <code>cov.quali</code> ).
<code>nodesize</code>	The value of the node size.
<code>mtry</code>	The number of variables randomly sampled as candidates at each split.
<code>ntree</code>	The number of trees.

**Details**

The survival random forest tree is obtained by using the `randomForestSRC` package.

**Value**

model	The estimated model.
group	The name of the variable related to the exposure/treatment.
cov.quant	The name(s) of the variable(s) related to the possible quantitative covariates.
cov.qual	The name(s) of the variable(s) related to the possible qualitative covariates.
data	The data frame used for learning. The first column is entitled <code>times</code> and corresponds to the observed follow-up times. The second column is entitled <code>failures</code> and corresponds to the event indicators. The other columns correspond to the predictors.
times	A vector of numeric values with the times of the predictions.
predictions	A matrix with the predictions of survivals of each subject (lines) for each observed time (columns).

**References**

Simon, N., Friedman, J., Hastie, T. and Tibshirani, R. (2011) Regularization Paths for Cox's Proportional Hazards Model via Coordinate Descent, *Journal of Statistical Software*, Vol. 39(5), 1-13, <https://www.jstatsoft.org/v39/i05/>

**Examples**

```
data(dataDIVAT2)

# The estimation of the model
model <- LIB_RSF(times="times", failures="failures", data=dataDIVAT2,
  cov.quant=c("age"), cov.qual=c("hla", "retransplant", "ecd"), nodesize=10,
  mtry=2, ntree=100)

# The predicted survival of the first subject of the training sample
plot(y=model$predictions[1,], x=model$times, xlab="Time (years)",
  ylab="Predicted survival", col=1, type="l", lty=1, lwd=2, ylim=c(0,1))
```

LIB\_SNN

*Library of the Super Learner for Survival Neural Network***Description**

Fit a 1-layer neural network based on the partial likelihood from a Cox proportional hazards model.

**Usage**

```
LIB_SNN(times, failures, group=NULL, cov.quant=NULL, cov.qual=NULL,
  data, n.nodes, decay, batch.size, epochs)
```

**Arguments**

<code>times</code>	The name of the variable related the numeric vector with the follow-up times.
<code>failures</code>	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
<code>group</code>	The name of the variable related to the exposure/treatment. This variable shall have only two modalities encoded 0 for the untreated/unexposed patients and 1 for the treated/exposed ones. The default value is NULL: no specific exposure/treatment is considered. When a specific exposure/treatment is considered, it will be forced in the algorithm or related interactions will be tested when possible.
<code>cov.quant</code>	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
<code>cov.qual</code>	The name(s) of the variable(s) related to the possible qualitative covariates. These variables must be numeric with two levels: 0 and 1. A complete disjunctive form must be used for covariates with more levels.
<code>data</code>	A data frame for training the model in which to look for the variables related to the status of the follow-up time ( <code>times</code> ), the event ( <code>failures</code> ), the optional treatment/exposure ( <code>group</code> ) and the covariables included in the previous model ( <code>cov.quant</code> and <code>cov.qual</code> ).
<code>n.nodes</code>	The number of hidden nodes.
<code>decay</code>	The value of the weight decay.
<code>batch.size</code>	The value of batch size.
<code>epochs</code>	The value of epochs.

**Details**

This function is based on the `deepsurv` from the `survivalmodels` package. You need to call Python using `reticulate`. In order to use it, the required Python packages must be installed with `reticulate::py_install`. Therefore, before running the present `LIB_SNN` function, you must install and call for the `reticulate` and `survivalmodels` packages, and install `pycox` by using the following command: `install_pycox(pip = TRUE, install_torch = FALSE)`. The `survivalSL` package functions without these supplementary installations if this learner is not included in the library.

**Value**

<code>model</code>	The estimated model.
<code>group</code>	The name of the variable related to the exposure/treatment.
<code>cov.quant</code>	The name(s) of the variable(s) related to the possible quantitative covariates.
<code>cov.qual</code>	The name(s) of the variable(s) related to the possible qualitative covariates.
<code>data</code>	The data frame used for learning. The first column is entitled <code>times</code> and corresponds to the observed follow-up times. The second column is entitled <code>failures</code> and corresponds to the event indicators. The other columns correspond to the predictors.

times	A vector of numeric values with the times of the predictions.
predictions	A matrix with the predictions of survivals of each subject (lines) for each observed time (columns).

## References

Katzman, J. L., Shaham, U., Cloninger, A., Bates, J., Jiang, T., & Kluger, Y. (2018). DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network. *BMC Medical Research Methodology*, 18(1), 24. <https://doi.org/10.1186/s12874-018-0482-1>

---

metrics *Metrics to Evaluate the Prognostic Capacities*

---

## Description

Compute several metrics to evaluate the prognostic capacities with time-to-event data.

## Usage

```
metrics(times, failures, data, prediction.matrix, prediction.times, metric,
pro.time=NULL, ROC.precision=seq(.01, .99, by=.01))
```

## Arguments

times	The name of the variable related the numeric vector with the follow-up times.
failures	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
data	A data frame for in which to look for the variables related to the status of the follow-up time (times) and the event (failures).
prediction.matrix	A matrix with the predictions of survivals of each subject (lines) for each prognostic times (columns).
prediction.times	A vector of numeric values with the times of the predictions (same length than the number of columns of prediction.matrix).
metric	The metric to compute. See details.
pro.time	This optional value of prognostic time represents the maximum delay for which the capacity of the variable is evaluated. The same unit than the one used in the argument times. Not used for the following metrics: "loglik", "ibs", "bll", and "ibll". Default value is the time at which half of the subjects are still at risk.
ROC.precision	An optional argument with the percentiles (between 0 and 1) of the prognostic variable used for computing each point of the time dependent ROC curve. Only used when metric="auc". 0 (min) and 1 (max) are not allowed. By default, the precision is seq(.01, .99, .01).

## Details

The following metrics can be used: "bs" for the Brier score at the prognostic time `pro.time`, "ci" for the concordance index at the prognostic time `pro.time`, "loglik" for the log-likelihood, "ibs" for the integrated Brier score up to the last observed time of event, "ibll" for the integrated binomial log-likelihood up to the last observed time of event, "bll" for the binomial log-likelihood, "ribs" for the restricted integrated Brier score up to the prognostic time `pro.time`, "ribll" for the restricted integrated binomial log-likelihood up to the last observed time of event, "bll" for the binomial log-likelihood, "auc" for the area under the time-dependent ROC curve up to the prognostic time `pro.time`.

## Value

A numeric value with the metric estimation.

## Examples

```
data(dataDIVAT2)

# The estimation of the model
model <- LIB_COXridge(times="times", failures="failures", data=dataDIVAT2,
  cov.quant=c("age"), cov.quali=c("hla", "retransplant", "ecd"), lambda=1)

# The apparent AUC at 10-year post-transplantation
metrics(times="times", failures="failures", data=dataDIVAT2,
  prediction.matrix=model$predictions, prediction.times=model$times,
  metric="auc", pro.time=10)

# The integrated Brier score up to 10 years post-transplantation
metrics(times="times", failures="failures", data=dataDIVAT2,
  prediction.matrix=model$predictions, prediction.times=model$times,
  metric="ribs", pro.time=10)
```

---

plot.libsl

*Calibration Plot for a Cox-like Model*

---

## Description

A calibration plot of an object of the class `libsl` (library of survival super learner).

## Usage

```
## S3 method for class 'libsl'
plot(x, n.groups=5, pro.time=NULL,
  newdata=NULL, times=NULL, failures=NULL, ...)
```

**Arguments**

<code>x</code>	An object returned by a library of survival super learner.
<code>n.groups</code>	A numeric value with the number of groups by their class probabilities. The default is 5.
<code>pro.time</code>	The prognostic time at which the calibration plot of the survival probabilities.
<code>newdata</code>	An optional data frame containing the new sample for validation with covariate values, follow-up times, and event status. The default value is NULL, the calibration plot is performed from the same subjects of the training sample.
<code>times</code>	The name of the variable related the numeric vector with the follow-up times in <code>newdata</code> (optional argument only necessary when <code>newdata</code> is not NULL).
<code>failures</code>	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event) in <code>newdata</code> (optional argument only necessary when <code>newdata</code> is not NULL).
<code>...</code>	Additional arguments affecting the plot.

**Details**

The plot represents the observed survival and the related 95% confidence intervals, which are respectively estimated by the Kaplan and Meier estimator and the Greenwood formula, against the mean of the predictive values for individuals stratified into groups of the same size according to the percentiles. The identity line is usually included for reference.

**Value**

No return value for this S3 method.

**See Also**

[plot.default](#)

**Examples**

```
data(dataDIVAT2)

# The estimation of the model from the first 200 lignes
model <- LIB_COXall(times="times", failures="failures", data=dataDIVAT2,
  cov.quant=c("age"), cov.quali=c("hla", "retransplant", "ecd"))

# The calibration plot from the validation sample of 150 patients
plot(model, n.groups=5, pro.time=12, col=3,
  xlab="Predicted 12-year survival", ylab="Observed 12-year survival",
  newdata=dataDIVAT2[151:300,], times="times", failures="failures")
```

---

plot.rocrisca                      *Plot Method for 'rocrisca' Objects*

---

### Description

A plot of ROC curves is produced.

### Usage

```
## S3 method for class 'rocrisca'  
plot(x, ..., information=TRUE)
```

### Arguments

x	An object of class rocrisca, returned by the functions roc.binary, roc.net, roc.summary, and roc.time.
...	Additional arguments affecting the plot.
information	A logical value indicating whether the non-information line is plotted. The default value is TRUE.

### Value

No return value for this S3 method.

### See Also

[plot.default](#)

### Examples

```
data(dataDIVAT3)  
  
# A subgroup analysis to reduce the time needed for this example  
  
dataDIVAT3 <- dataDIVAT3[1:400,]  
  
# The time-dependent ROC curve to evaluate the  
# capacities of the recipient age for the prognosis of post-kidney  
# transplant mortality up to 2000 days.  
  
# Compute the raw sensitivity and specificity  
roc1 <- roc(times="death.time", failures="death", variable="ageR",  
confounders=~1, data=dataDIVAT3, pro.time=2000,  
precision=seq(0.1,0.9, by=0.2))  
  
plot(roc1, type="b", col=1, pch=2, lty=2, xlab="1-specificity", ylab="sensitivity")
```



plot.slttime

*Calibration Plot for Super Learner***Description**

A calibration plot of a Super Learner obtained by the function `survivalSL`.

**Usage**

```
## S3 method for class 'slttime'
plot(x, method, n.groups, pro.time, newdata,
     times, failures, ...)
```

**Arguments**

<code>x</code>	An object returned by the function <code>survivalSL</code> .
<code>method</code>	A character string with the name of the algorithm included in the SL for which the calibration plot is performed. The default is "sl" for the Super Learner.
<code>n.groups</code>	A numeric value with the number of groups by their class probabilities. The default is 5.
<code>pro.time</code>	The prognostic time at which the calibration plot of the survival probabilities.
<code>newdata</code>	An optional data frame containing the new sample for validation with covariate values, follow-up times, and event status. The default value is <code>NULL</code> , the calibration plot is performed from the same subjects of the training sample.
<code>times</code>	The name of the variable related the numeric vector with the follow-up times in <code>newdata</code> (optional argument only necessary when <code>newdata</code> is not <code>NULL</code> ).
<code>failures</code>	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event) in <code>newdata</code> (optional argument only necessary when <code>newdata</code> is not <code>NULL</code> ).
<code>...</code>	Additional arguments affecting the plot.

**Details**

The plot represents the observed survival and the related 95% confidence intervals, which are respectively estimated by the Kaplan and Meier estimator and the Greenwood formula, against the mean of the predictive values for individuals stratified into groups of the same size according to the percentiles. The identity line is usually included for reference.

**Value**

No return value for this S3 method.

**See Also**

[plot.default](#)

**Examples**

```

data(dataDIVAT2)

#The outcome model base on a Super Learner from the first 150 individuals of the data base
s11 <- survivalSL( methods=c("LIB_AFTgamma", "LIB_PHgompertz"), metric="ci",
  data=dataDIVAT2[1:150,], times="times", failures="failures", group="ecd",
  cov.quant=c("age"), cov.quali=c("hla", "retransplant"), cv=3)

# The calibration plot from the validation sample of 150 patients
plot(s11, method="sl", n.groups=5, pro.time=12, col=2,
  xlab="Predicted 12-year survival", ylab="Observed 12-year survival",
  newdata=dataDIVAT2[151:300,], times="times", failures="failures")

```

---

predict.libsl

*Prediction from an Flexible Parametric Model*


---

**Description**

Predict the survival based on a model or algorithm from an object of the class libsl.

**Usage**

```

## S3 method for class 'libsl'
predict(object, newdata, newtimes, ...)

```

**Arguments**

object	An object returned by the function LIB_AFTllogis, LIB_AFTgamma, LIB_AFTgamma, LIB_AFTweibull, LIB_PHexponential, LIB_PHspline or LIB_PHgompertz.
newdata	An optional data frame containing covariate values at which to produce predicted values. There must be a column for every covariate included in cov.quant and cov.quali included in the training sample. The default value is NULL, the predicted values are computed for the subjects of the training sample.
newtimes	The times at which to produce predicted values. The default value is NULL, the predicted values are computed for the observed times in the training data frame.
...	For future methods.

**Details**

The model object is obtained from the flexsurvreg package.

**Value**

times	A vector of numeric values with the times of the predictions.
predictions	A matrix with the predictions of survivals of each subject (lines) for each observed time (columns).

**Examples**

```

data(dataDIVAT2)

# The estimation of the model from the first 200 lines
model <- LIB_PHgompertz(times="times", failures="failures", data=dataDIVAT2[1:200,],
  cov.quant=c("age"), cov.qual=c("hla", "retransplant", "ecd"))

# Predicted survival for 2 new subjects
pred <- predict(model,
  newdata=data.frame(age=c(52,52), hla=c(0,1), retransplant=c(1,1), ecd=c(0,1)))

plot(y=pred$predictions[1,], x=pred$times, xlab="Time (years)", ylab="Predicted survival",
  col=1, type="l", lty=1, lwd=2, ylim=c(0,1))

lines(y=pred$predictions[2,], x=pred$times, col=2, type="l", lty=1, lwd=2)

legend("bottomright", col=c(1,2), lty=1, lwd=2, c("Subject #1", "Subject #2"))

```

---

predict.slttime

*Prediction from a Super Learner for Censored Outcomes*


---

**Description**

Predict the survival of new observations based on an SL by using the survivalSL function.

**Usage**

```

## S3 method for class 'slttime'
predict(object, newdata, newtimes, ...)

```

**Arguments**

object	An object returned by the function survivalSL.
newdata	An optional data frame containing covariate values at which to produce predicted values. There must be a column for every covariate included in cov.quant and cov.qual included in the training sample. The default value is NULL, the predicted values are computed for the subjects of the training sample.
newtimes	The times at which to produce predicted values. The default value is NULL, the predicted values are computed for the observed times in the training data frame.
...	For future methods.

**Value**

times	A vector of numeric values with the times of the predictions.
predictions	A matrix with the predictions of survivals of each subject (lines) for each observed time (columns).

**See Also**[survivalSL](#).**Examples**

```

data(dataDIVAT2)

# The training of the super learner from the first 150 individuals of the data base
s11 <- survivalSL(method=c("LIB_COXridge", "LIB_AFTgamma"), metric="ci",
  data=dataDIVAT2[1:150,], times="times", failures="failures", pro.time = 12,
  cov.quantile=c("age"), cov.quali=c("hla", "retransplant", "ecd"), cv=3)

# Individual prediction for 2 new subjects
pred <- predict(s11,
  newdata=data.frame(age=c(52,52), hla=c(0,1), retransplant=c(1,1), ecd=c(0,1)))

plot(y=pred$predictions$sl[1,], x=pred$times, xlab="Time (years)",
  ylab="Predicted survival", col=1, type="l", lty=1, lwd=2, ylim=c(0,1))

lines(y=pred$predictions$sl[2,], x=pred$times, col=2, type="l", lty=1, lwd=2)

legend("bottomright", col=c(1,2), lty=1, lwd=2, c("Subject #1", "Subject #2"))

```

---

`print.libsl`*S3 Method for Printing an 'libsl' Object*

---

**Description**

Print the model or algorithm.

**Usage**

```

## S3 method for class 'libsl'
print(x, ...)

```

**Arguments**

<code>x</code>	An object returned by the function <code>flexsurv</code> .
<code>...</code>	For future methods.

**Value**

No return value for this S3 method.

**See Also**[LIB\\_AFTgamma](#), [LIB\\_AFTgamma](#), [LIB\\_AFTllogis](#), [LIB\\_AFTweibull](#), [LIB\\_PHexponential](#), [LIB\\_PHgompertz](#).

**Examples**

```
data(dataDIVAT2)

model <- LIB_AFTgamma(times="times", failures="failures", data=dataDIVAT2[1:100,],
  cov.quanti=c("age"), cov.quali=c("hla", "retransplant", "ecd"))

print(model)
```

---

print.sitime	<i>S3 Method for Printing an 'sitime' Object</i>
--------------	--------------------------------------------------

---

**Description**

Print the contribution of learners included in the super learner.

**Usage**

```
## S3 method for class 'sitime'
print(x, digits=7, ...)
```

**Arguments**

x	An object returned by the function survivalSL.
digits	An optional integer for the number of digits to print when printing numeric values.
...	For future methods.

**Value**

No return value for this S3 method.

**Examples**

```
data(dataDIVAT2)

s11 <- survivalSL(method=c("LIB_COXridge", "LIB_AFTgamma"), metric="ci",
  data=dataDIVAT2[1:150,], times="times", failures="failures", pro.time = 12,
  cov.quanti=c("age"), cov.quali=c("hla", "retransplant", "ecd"), cv=3)

print(s11, digits=4)
```

roc

*Time-Dependent ROC Curves With Right Censored Data***Description**

This function allows for the estimation of time-dependent ROC curve by considering possible confounding factors. This method is implemented by standardizing and weighting based on an IPW estimator.

**Usage**

```
roc(times, failures, variable, confounders, data,
    pro.time, precision=seq(.01, .99, by=.01))
```

**Arguments**

times	A character string with the name of the variable in data which represents the follow up times.
failures	A character string with the name of the variable in data which represents the event indicator (0=right censored, 1=event).
variable	A character string with the name of the variable in data which represents the prognostic variable under interest. This variable is collected at the baseline. The variable must be previously standardized according to the covariates among the controls as proposed by Le Borgne et al. (2017).
confounders	An object of class "formula". More precisely only the right part with an expression of the form $\sim$ model, where model is the linear predictor of the logistic regressions performed for each cut-off value. The user can use $\sim 1$ to obtain the crude estimation.
data	An object of the class data.frame containing the variables previously detailed.
pro.time	The value of prognostic time represents the maximum delay for which the capacity of the variable is evaluated. The same unit than the one used in the argument times.
precision	The quintiles (between 0 and 1) of the prognostic variable used for computing each point of the time dependent ROC curve. 0 (min) and 1 (max) are not allowed.

**Details**

This function computes confounder-adjusted time-dependent ROC curve with right-censored data. We adapted the naive IPCW estimator as explained by Blanche, Dartigues and Jacqmin-Gadda (2013) by considering the probability of experiencing the event of interest before the fixed prognostic time, given the possible confounding factors.

**Value**

table	This data frame presents the sensitivities and specificities associated with the cut-off values. J represents the Youden index.
auc	The area under the time-dependent ROC curve for a prognostic up to pro. time.

**References**

Blanche et al. (2013) Review and comparison of roc curve estimators for a time-dependent outcome with marker-dependent censoring. *Biometrical Journal*, 55, 687-704. <doi:10.1002/bimj.201200045>

Le Borgne et al. Standardized and weighted time-dependent ROC curves to evaluate the intrinsic prognostic capacities of a marker by taking into account confounding factors. *Stat Methods Med Res.* 27(11):3397-3410, 2018. <doi: 10.1177/0962280217702416>.

**Examples**

```
# import and attach the data example
data(dataDIVAT3)

# A subgroup analysis to reduce the time needed for this example

dataDIVAT3 <- dataDIVAT3[1:400,]

# The standardized and weighted time-dependent ROC curve to evaluate the
# capacities of the recipient age for the prognosis of post kidney
# transplant mortality up to 2000 days by taking into account the
# donor age and the recipient gender.

# 1. Standardize the marker according to the covariates among the controls
lm1 <- lm(ageR ~ ageD + sexeR, data=dataDIVAT3[dataDIVAT3$death.time >= 2500,])
dataDIVAT3$ageR_std <- (dataDIVAT3$ageR - (lm1$coef[1] + lm1$coef[2] * dataDIVAT3$ageD +
  lm1$coef[3] * dataDIVAT3$sexeR)) / sd(lm1$residuals)

# 2. Compute the sensitivity and specificity from the proposed IPW estimators
roc2 <- roc(times="death.time", failures="death", variable="ageR_std",
  confounders=~bs(ageD, df=3) + sexeR, data=dataDIVAT3, pro.time=2000,
  precision=seq(0.1,0.9, by=0.2))

# The corresponding ROC graph
plot(roc2, col=2, pch=2, lty=1, type="b", xlab="1-specificity", ylab="sensibility")

# The corresponding AUC
roc2$auc
```

**Description**

Return predictive performances of a model or algorithm obtained by a library of the class libsl.

**Usage**

```
## S3 method for class 'libsl'
summary(object, newdata=NULL, ROC.precision=seq(.01,.99,.01), digits=7, ...)
```

**Arguments**

object	An object returned by a library of the class libsl.
newdata	An optional data frame containing the new sample for validation with covariate values, follow-up times, and event status. The default value is NULL, the calibration plot is performed from the same subjects of the training sample.
ROC.precision	An optional argument with the percentiles (between 0 and 1) of the prognostic variable used for computing each point of the time dependent ROC curve. 0 (min) and 1 (max) are not allowed. By default, the precision is seq(.01, .99, .01).
digits	An optional integer for the number of digits to print when printing numeric values.
...	Additional arguments affecting the summary which are passed from libsl by default. It concerns the argument times, failures, and pro.time.

**Details**

The following metrics are returned: "brier" for the Brier score at the prognostic time `pro.time`, "ibs" for the Integrated Brier score up to the last observed time of event, "ibll" for the Integrated Binomial Log-likelihood up to the last observed time of event, "bll" for the binomial Log-likelihood, "ribs" for the restricted Integrated Brier score up to the prognostic time `pro.time`, "ribll" for the restricted Integrated Binomial Log-likelihood up to the last observed time of event, "bll" for the binomial Log-likelihood, "auc" for the area under the time-dependent ROC curve up to the prognostic time `pro.time`.

**Value**

No return value for this S3 method.

**See Also**

[LIB\\_AFTgamma](#), [LIB\\_AFTgamma](#), [LIB\\_AFTllogis](#), [LIB\\_AFTweibull](#), [LIB\\_PHexponential](#), [LIB\\_PHgompertz](#).

**Examples**

```
data(dataDIVAT2)

# The training of the Weibull model with the first 400 patients
model <- LIB_PHgompertz(times="times", failures="failures", data=dataDIVAT2[1:400,],
  cov.quant=c("age"), cov.quali=c("hla", "retransplant", "ecd"))

# The prognostic capacities from the same training sample
# (up to 4 years for several indicators)
summary(model, pro.time=4)

# The prognostic capacities from a validation of the next 150 patients
```



```
# (up to 4 years for several indicators)
summary(model, pro.time=4, newdata=dataDIVAT2[401:550,], times="times",
failures="failures")
```

summary.sltime

*Summaries of a Super Learner***Description**

Return goodness-of-fit indicators of a Super Learner obtained by the function `survivalSL`.

**Usage**

```
## S3 method for class 'sltime'
summary(object, method="sl", newdata=NULL,
ROC.precision=seq(.01,.99,.01), digits=7, ...)
```

**Arguments**

<code>object</code>	An object returned by the function <code>survivalSL</code> .
<code>method</code>	A character string with the name of the algorithm included in the SL for which the calibration plot is performed. The default is "sl" for the Super Learner.
<code>newdata</code>	An optional data frame containing the new sample for validation with covariate values, follow-up times, and event status. The default value is <code>NULL</code> , the calibration plot is performed from the same subjects of the training sample.
<code>ROC.precision</code>	An optional argument with the percentiles (between 0 and 1) of the prognostic variable used for computing each point of the time dependent ROC curve. 0 (min) and 1 (max) are not allowed. By default, the precision is <code>seq(.01, .99, .01)</code> .
<code>digits</code>	An optional integer for the number of digits to print when printing numeric values.
<code>...</code>	Additional arguments affecting the summary which are passed from <code>libs1</code> by default. It concerns the argument <code>times</code> , <code>failures</code> , and <code>pro.time</code> .

**Details**

The following metrics are returned: "ci" for the concordance index at the prognostic time `pro.time`, "bs" for the Brier score at the prognostic time `pro.time`, "ibs" for the integrated Brier score up to the last observed time of event, "ibll" for the integrated binomial log-likelihood up to the last observed time of event, "bll" for the binomial Log-likelihood, "ribs" for the restricted Integrated Brier score up to the prognostic time `pro.time`, "ribll" for the restricted integrated binomial log-likelihood up to the last observed time of event, "bll" for the binomial log-likelihood, and "auc" for the area under the time-dependent ROC curve up to the prognostic time `pro.time`.

**Value**

No return value for this S3 method.

**See Also**

[survivalSL](#).

**Examples**

```
data(dataDIVAT2)

dataDIVAT2$train <- 1*rbinom(n=dim(dataDIVAT2)[1], size = 1, prob=1/2)

# The training of the super learner with 2 algorithms from the
# first 100 patients of the training sample
s11 <- survivalSL(method=c("LIB_AFTgamma", "LIB_PHgompertz"), metric="auc",
  data=dataDIVAT2[dataDIVAT2$train==1,][1:100,], times="times", failures="failures",
  pro.time = 12, cov.quant=c("age"), cov.quali=c("hla", "retransplant", "ecd"),
  cv=3)

# The prognostic capacities from the same training sample
summary(s11)
```

---

survivalSL

*Super Learner for Censored Outcomes*

---

**Description**

This function allows to compute a Super Learner (SL) to predict survival outcomes.

**Usage**

```
survivalSL(methods, metric="ci", data, times, failures, group=NULL,
  cov.quant=NULL, cov.quali=NULL, cv=10, param.tune=NULL, pro.time=NULL,
  optim.local.min=FALSE, ROC.precision=seq(.01,.99,.01),
  param.weights.fix=NULL, param.weights.init=NULL,
  keep.predictions=TRUE, progress=TRUE)
```

**Arguments**

methods	A vector of characters with the names of the algorithms included in the SL. At least two algorithms have to be included.
metric	The loss function used to estimate the weights of the algorithms in the SL. See details.
data	A data frame in which to look for the variables related to the status of the follow-up time (times), the event (failures), the optional treatment/exposure (group) and the covariables included in the previous model (cov.quant and cov.quali).
times	The name of the variable related the numeric vector with the follow-up times.
failures	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).

<code>group</code>	The name of the variable related to the exposure/treatment. This variable shall have only two modalities encoded 0 for the untreated/unexposed patients and 1 for the treated/exposed ones. The default value is NULL: no specific exposure/treatment is considered. When a specific exposure/treatment is considered, it will be forced in the algorithm or related interactions will be tested when possible.
<code>cov.quant</code>	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
<code>cov.qual</code>	The name(s) of the variable(s) related to the possible qualitative covariates. These variables must be numeric with two levels: 0 and 1. A complete disjunctive form must be used for covariates with more levels.
<code>cv</code>	The number of splits for cross-validation. The default value is 10.
<code>param.tune</code>	A list with a length equals to the number of algorithms included in methods. If NULL, the tuning parameters are estimated (see details).
<code>pro.time</code>	This optional value of prognostic time represents the maximum delay for which the capacity of the variable is evaluated. The same unit than the one used in the argument times. Not used for the following metrics: "loglik", "ibs", "bll", and "ibll". Default value is the time at which half of the subjects are still at risk.
<code>optim.local.min</code>	An optional logical value. If TRUE, the optimization is performed twice to better ensure the estimation of the weights. If FALSE (default value), the optimization is performed once.
<code>ROC.precision</code>	The percentiles (between 0 and 1) of the prognostic variable used for computing each point of the time dependent ROC curve. Only used when <code>metric="auc"</code> . 0 (min) and 1 (max) are not allowed. By default: <code>seq(.01, .99, .01)</code> .
<code>param.weights.fix</code>	A vector with the parameters of the multinomial logistic regression which generates the weights of the algorithms declared in methods. When completed, the related parameters are not estimated. The default value is NULL: the parameters are estimated by a cv-fold cross-validation. See details.
<code>param.weights.init</code>	A vector with the initial values of the parameters of the multinomial logistic regression which generates the weights of the algorithms declared in methods. The default value is NULL: the initial values are equaled to 0. See details.
<code>keep.predictions</code>	A logical value specifying if all the predictions for all the methods are saved. If FALSE, only the predictions related to the SL are saved (for space saving). The default is TRUE.
<code>progress</code>	A logical value to print a progress bar in the R console. The default is TRUE

## Details

Each object of the list declared in `param.tune` must have the same name than the names of the methods included in the SL. If `param.tune = NULL`, the tuning parameters of each algorithm are estimated by cv-fold cross-validation. Otherwise, the user can propose a tuning grid for each

method, as explained in the following table. The following metrics can be used: "ci" for the concordance index at the prognostic time `pro.time`, "bs" for the Brier score at the prognostic time `pro.time`, "loglik" for the log-likelihood, "ibs" for the integrated Brier score up to the last observed time of event, "ibll" for the Integrated binomial log-likelihood up to the last observed time of event, "bll" for the binomial log-likelihood, "ribs" for the restricted integrated Brier score up to the prognostic time `pro.time`, "ribll" for the restricted integrated binomial log-likelihood up to the last observed time of event, "bll" for the binomial log-likelihood, and "auc" for the area under the time-dependent ROC curve up to the prognostic time `pro.time`.

The following learners are available:

Names	Description	Package
"LIB_AFTgamma"	Gamma-distributed AFT model	flexsurv
"LIB_AFTgamma"	Generalized Gamma-distributed AFT model	flexsurv
"LIB_AFTweibull"	Weibull-distributed AFT model	flexsurv
"LIB_PHexponential"	Exponential-distributed PH model	flexsurv
"LIB_PHgompertz"	Gompertz-distributed PH model	flexsurv
"LIB_PHspline"	Spline-based PH model	flexsurv
"LIB_COXall"	Usual Cox model	survival
"LIB_COXaic"	Cox model with AIC-based forward selection	MASS
"LIB_COXen"	Elastic Net Cox model	glmnet
"LIB_COXlasso"	Lasso Cox model	glmnet
"LIB_COXridge"	Ridge Cox model	glmnet
"LIB_RSF"	Survival Random Forest	randomForestSRC
"LIB_SNN"	(Python-based) Survival Neural Network	survivalmodels
"LIB_PLANN"	(Python-based) Survival Neural Network	survivalPLANN

The following loss functions for the estimation of the super learner weights are available (`metric`):

- Area under the ROC curve ("`auc`")
- Concordance index ("`ci`")
- Brier score ("`bs`")
- Binomial log-likelihood ("`bll`")
- Integrated Brier score ("`ibs`")
- Integrated binomial log-likelihood ("`ibll`")
- Restricted integrated Brier score ("`ribs`")
- Restricted integrated binomial log-Likelihood ("`ribll`")

## Value

<code>times</code>	A vector of numeric values with the times of the predictions.
<code>predictions</code>	A list of matrices with the predictions of survivals of each subject (lines) for each observed time (columns). Each matrix corresponds to the included methods and the resulted SL (the last item entitled "sl"). If <code>keep.predictions=TRUE</code> , it corresponds to a matrix with predictions related to the SL.

data	The data frame used for learning. The first column is entitled <code>times</code> and corresponds to the observed follow-up times. The second column is entitled <code>failures</code> and corresponds to the event indicators. The other columns correspond to the predictors.
predictors	A list with the predictors involved in <code>group</code> , <code>cov.quant</code> and <code>cov.quali</code> .
ROC.precision	The percentiles (between 0 and 1) of the prognostic variable used for computing each point of the time dependent ROC curve.
cv	The number of splits for cross-validation.
pro.time	The maximum delay for which the capacity of the variable is evaluated.
models	A list with the estimated models/algorithms included in the SL.
weights	A list composed by two vectors: the regressions coefficients of the logistic multinomial regression and the resulting weights' values
metric	A list composed by two vectors: the loss function used to estimate the weights of the algorithms in the SL and its value.
param.tune	The estimated tuning parameters.

## References

Polley E and van der Laan M. Super Learner In Prediction. <http://biostats.bepress.com>. 2010.

## Examples

```
data(dataDIVAT2)

# The Super Learner based from the first 300 individuals of the data base
s11 <- survivalSL(methods=c("LIB_AFTgamma", "LIB_PHgompertz"), metric="ci",
  data=dataDIVAT2[1:300,], times="times", failures="failures", group="ecd",
  cov.quant=c("age"), cov.quali=c("hla", "retransplant"), cv=5)

# Individual prediction
pred <- predict(s11, newdata=data.frame(age=c(52,52), hla=c(0,1),
  retransplant=c(1,1), ecd=c(0,1)))

plot(y=pred$predictions$s1[1,], x=pred$times, xlab="Time (years)",
  ylab="Predicted survival", col=1, type="l", lty=1, lwd=2, ylim=c(0,1))

lines(y=pred$predictions$s1[2,], x=pred$times, col=2, type="l", lty=1, lwd=2)

legend("topright", col=c(1,2), lty=1, lwd=2, c("Subject #1", "Subject #2"))
```

---

tuneCOXaic

*Tune a Cox Model with a Forward Selection Based on the AIC*


---

## Description

This function finds the model which minimize the AIC of a Cox PH model.

**Usage**

```
tuneCOXaic(times, failures, group=NULL, cov.quant= NULL, cov.quali=NULL,
data, model.min=NULL, model.max=NULL)
```

**Arguments**

<code>times</code>	The name of the variable related the numeric vector with the follow-up times.
<code>failures</code>	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
<code>group</code>	The name of the variable related to the exposure/treatment. This variable shall have only two modalities encoded 0 for the untreated/unexposed patients and 1 for the treated/exposed ones. The default value is NULL: no specific exposure/treatment is considered. When a specific exposure/treatment is considered, it will be forced in the algorithm or related interactions will be tested when possible.
<code>cov.quant</code>	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
<code>cov.quali</code>	The name(s) of the variable(s) related to the possible qualitative covariates. These variables must be numeric with two levels: 0 and 1. A complete disjunctive form must be used for covariates with more levels.
<code>data</code>	A data frame for training the model in which to look for the variables related to the status of the follow-up time ( <code>times</code> ), the event ( <code>failures</code> ), the optional treatment/exposure ( <code>group</code> ) and the covariables included in the previous model ( <code>cov.quant</code> and <code>cov.quali</code> ).
<code>model.min</code>	An optional argument with the minimal set of covariates.
<code>model.max</code>	An optional argument with the maximal set of covariates.

**Details**

The function runs the `stepAIC` function of the MASS package for covariates' selection.

**Value**

<code>optimal</code>	The names of covariate to adjust the fit.
<code>results</code>	The result of the <code>stepAIC</code> process.

**References**

Venables, W. N. and Ripley, B. D. (2002) Modern Applied Statistics with S. Fourth edition. Springer.

**Examples**

```
data(dataDIVAT2)

tune.model <- tuneCOXaic(times="times", failures="failures", data=dataDIVAT2,
  cov.quant=c("age"), cov.quali=c("hla", "retransplant", "ecd"))
```

```

tune.model$optimal$final.model # the covariate in the model with the best AIC

# The estimation of the training model with the corresponding lambda value
model <- LIB_COXaic(times="times", failures="failures", data=dataDIVAT2,
  cov.quant=c("age"), cov.quali=c("hla", "retransplant", "ecd"),
  final.model=tune.model$optimal$final.model)

# The resulted predicted survival of the first subject of the training sample
plot(y=model$predictions[1,], x=model$times, xlab="Time (years)",
  ylab="Predicted survival", col=1, type="l", lty=1, lwd=2, ylim=c(0,1))

```

tuneCOXen

*Tune Elastic Net Cox Regression***Description**

This function finds the optimal lambda and alpha parameters for an elastic net Cox regression.

**Usage**

```

tuneCOXen(times, failures, group=NULL, cov.quant=NULL, cov.quali=NULL,
  data, cv=10, parallel=FALSE, alpha, lambda)

```

**Arguments**

times	The name of the variable related the numeric vector with the follow-up times.
failures	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
group	The name of the variable related to the exposure/treatment. This variable shall have only two modalities encoded 0 for the untreated/unexposed patients and 1 for the treated/exposed ones. The default value is NULL: no specific exposure/treatment is considered. When a specific exposure/treatment is considered, it will be forced in the algorithm or related interactions will be tested when possible.
cov.quant	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
cov.quali	The name(s) of the variable(s) related to the possible qualitative covariates. These variables must be numeric with two levels: 0 and 1. A complete disjunctive form must be used for covariates with more levels.
data	A data frame for training the model in which to look for the variables related to the status of the follow-up time (times), the event (failures), the optional treatment/exposure (group) and the covariables included in the previous model (cov.quant and cov.quali).
cv	The value of the number of folds. The default value is 10.
parallel	If TRUE, use parallel foreach to fit each fold. The default is FALSE.
alpha	The values of the regularization parameter alpha optimized over.
lambda	The values of the regularization parameter lambda optimized over.

**Details**

The function runs the `cv.glmnet` function of the `glmnet` package.

**Value**

<code>optimal</code>	The value of lambda that gives the minimum mean cross-validated error.
<code>results</code>	The data frame with the mean cross-validated errors for each lambda values.

**References**

Simon, N., Friedman, J., Hastie, T. and Tibshirani, R. (2011) Regularization Paths for Cox's Proportional Hazards Model via Coordinate Descent, *Journal of Statistical Software*, Vol. 39(5), 1-13, <https://www.jstatsoft.org/v39/i05/>

**Examples**

```
data(dataDIVAT2)

tune.model <- tuneCOXen(times="times", failures="failures", data=dataDIVAT2,
  cov.quanti=c("age"), cov.quali=c("hla", "retransplant", "ecd"), cv=5,
  alpha=seq(.1, 1, by=.1), lambda=seq(.1, 1, by=.1))

tune.model$optimal$lambda # the estimated lambda value

# The estimation of the training model with the corresponding lambda value
model <- LIB_COXen(times="times", failures="failures", data=dataDIVAT2,
  cov.quanti=c("age"), cov.quali=c("hla", "retransplant", "ecd"),
  alpha=tune.model$optimal$alpha,
  lambda=tune.model$optimal$lambda)

# The resulted predicted survival of the first subject of the training sample
plot(y=model$predictions[1,], x=model$times, xlab="Time (years)",
  ylab="Predicted survival", col=1, type="l", lty=1, lwd=2, ylim=c(0,1))
```

---

tuneCOXlasso

*Tune Lasso Cox Regression*


---

**Description**

This function finds the optimal lambda parameter for a Lasso Cox regression.

**Usage**

```
tuneCOXlasso(times, failures, group=NULL, cov.quanti=NULL, cov.quali=NULL,
  data, cv=10, parallel=FALSE, lambda)
```



**Arguments**

<code>times</code>	The name of the variable related the numeric vector with the follow-up times.
<code>failures</code>	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
<code>group</code>	The name of the variable related to the exposure/treatment. This variable shall have only two modalities encoded 0 for the untreated/unexposed patients and 1 for the treated/exposed ones. The default value is NULL: no specific exposure/treatment is considered. When a specific exposure/treatment is considered, it will be forced in the algorithm or related interactions will be tested when possible.
<code>cov.quant</code>	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
<code>cov.qual</code>	The name(s) of the variable(s) related to the possible qualitative covariates. These variables must be numeric with two levels: 0 and 1. A complete disjunctive form must be used for covariates with more levels.
<code>data</code>	A data frame for training the model in which to look for the variables related to the status of the follow-up time ( <code>times</code> ), the event ( <code>failures</code> ), the optional treatment/exposure ( <code>group</code> ) and the covariables included in the previous model ( <code>cov.quant</code> and <code>cov.qual</code> ).
<code>cv</code>	The value of the number of folds. The default value is 10.
<code>parallel</code>	If TRUE, use parallel foreach to fit each fold. The default is FALSE.
<code>lambda</code>	The values of the regularization parameter lambda optimized over.

**Details**

The function runs the `cv.glmnet` function of the `glmnet` package.

**Value**

<code>optimal</code>	The value of lambda that gives the minimum mean cross-validated error.
<code>results</code>	The data frame with the mean cross-validated errors for each lambda values.

**References**

Simon et al. (2011) Regularization Paths for Cox's Proportional Hazards Model via Coordinate Descent, *Journal of Statistical Software*, Vol. 39(5), 1-13, <https://www.jstatsoft.org/v39/i05/>

**Examples**

```
data(dataDIVAT2)

tune.model <- tuneCOXlasso(times="times", failures="failures", data=dataDIVAT2,
  cov.quant=c("age"), cov.qual=c("hla", "retransplant", "ecd"),
  cv=5, lambda=seq(0, 10, by=.1))

tune.model$optimal$lambda # the estimated lambda value
```

```
# The estimation of the training model with the corresponding lambda value
model <- LIB_COXlasso(times="times", failures="failures", data=dataDIVAT2,
  cov.quant=c("age"), cov.qual=c("hla", "retransplant", "ecd"),
  lambda=tune.model$optimal$lambda)

# The resulted predicted survival of the first subject of the training sample
plot(y=model$predictions[1,], x=model$times, xlab="Time (years)",
  ylab="Predicted survival", col=1, type="l", lty=1, lwd=2, ylim=c(0,1))
```

---

tuneCOXridge

*Tune Ridge Cox Regression*


---

## Description

This function finds the optimal lambda parameter for a ridge Cox regression.

## Usage

```
tuneCOXridge(times, failures, group=NULL, cov.quant=NULL,
  cov.qual=NULL, data, cv=10, parallel=FALSE, lambda)
```

## Arguments

times	The name of the variable related the numeric vector with the follow-up times.
failures	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
group	The name of the variable related to the exposure/treatment. This variable shall have only two modalities encoded 0 for the untreated/unexposed patients and 1 for the treated/exposed ones. The default value is NULL: no specific exposure/treatment is considered. When a specific exposure/treatment is considered, it will be forced in the algorithm or related interactions will be tested when possible.
cov.quant	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
cov.qual	The name(s) of the variable(s) related to the possible qualitative covariates. These variables must be numeric with two levels: 0 and 1. A complete disjunctive form must be used for covariates with more levels.
data	A data frame for training the model in which to look for the variables related to the status of the follow-up time (times), the event (failures), the optional treatment/exposure (group) and the covariables included in the previous model (cov.quant and cov.qual).
cv	The value of the number of folds. The default value is 10.
parallel	If TRUE, use parallel foreach to fit each fold. The default is FALSE.
lambda	The values of the regularization parameter lambda optimized over.

**Details**

The function runs the `cv.glmnet` function of the `glmnet` package.

**Value**

<code>optimal</code>	The value of lambda that gives the minimum mean cross-validated error.
<code>results</code>	The data frame with the mean cross-validated errors for each lambda values.

**References**

Simon, N., Friedman, J., Hastie, T. and Tibshirani, R. (2011) Regularization Paths for Cox's Proportional Hazards Model via Coordinate Descent, *Journal of Statistical Software*, Vol. 39(5), 1-13, <https://www.jstatsoft.org/v39/i05/>

**Examples**

```
data(dataDIVAT2)

tune.model <- tuneCOXridge(times="times", failures="failures", data=dataDIVAT2,
  cov.quant=c("age"), cov.quali=c("hla", "retransplant", "ecd"),
  cv=5, lambda=seq(0, 10, by=.1))

tune.model$optimal$lambda # the estimated lambda value

# The estimation of the training model with the corresponding lambda value
model <- LIB_COXridge(times="times", failures="failures", data=dataDIVAT2,
  cov.quant=c("age"), cov.quali=c("hla", "retransplant", "ecd"),
  lambda=tune.model$optimal$lambda)

# The resulted predicted survival of the first subject of the training sample
plot(y=model$predictions[1,], x=model$times, xlab="Time (years)",
  ylab="Predicted survival", col=1, type="l", lty=1, lwd=2, ylim=c(0,1))
```

---

tunePHspline

*Tune a Survival Regression using the Royston/Parma Spline Model*


---

**Description**

This function finds the optimal number of knots of the spline function.

**Usage**

```
tunePHspline(times, failures, group=NULL, cov.quant=NULL, cov.quali=NULL,
  data, cv=10, k)
```

**Arguments**

<code>times</code>	The name of the variable related the numeric vector with the follow-up times.
<code>failures</code>	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
<code>group</code>	The name of the variable related to the exposure/treatment. This variable shall have only two modalities encoded 0 for the untreated/unexposed patients and 1 for the treated/exposed ones. The default value is NULL: no specific exposure/treatment is considered. When a specific exposure/treatment is considered, it will be forced in the algorithm or related interactions will be tested when possible.
<code>cov.quant</code>	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
<code>cov.qual</code>	The name(s) of the variable(s) related to the possible qualitative covariates. These variables must be numeric with two levels: 0 and 1. A complete disjunctive form must be used for covariates with more levels.
<code>data</code>	A data frame for training the model in which to look for the variables related to the status of the follow-up time ( <code>times</code> ), the event ( <code>failures</code> ), the optional treatment/exposure ( <code>group</code> ) and the covariables included in the previous model ( <code>cov.quant</code> and <code>cov.qual</code> ).
<code>cv</code>	The value of the number of folds. The default value is 10.
<code>k</code>	The number of knots optimized over.

**Details**

The function runs the `flexsurvspline` function of the `flexsurv` package. The metric used in the cross-validation is the C-index.

**Value**

<code>optimal</code>	The value of <code>k</code> that gives the maximum mean cross-validated C-index.
<code>results</code>	The data frame with the mean cross-validated C-index according to <code>k</code> .

**References**

Royston, P. and Parmar, M. (2002). Flexible parametric proportional-hazards and proportional odds models for censored survival data, with application to prognostic modelling and estimation of treatment effects. *Statistics in Medicine* 21(1):2175-2197. doi: 10.1002/sim.1203

**Examples**

```
data(dataDIVAT2)

# The estimation of the hyperparameters

tune.model <- tunePHspline(times="times", failures="failures", data=dataDIVAT2[1:200,],
  cov.quant=c("age"), cov.qual=c("hla", "retransplant", "ecd"),
  cv=3, k=1:2)
```

```

# the estimated nodesize value

tune.model$optimal
tune.model$results

# The estimation of the training model with the corresponding lambda value

model <- LIB_PHSpline(times="times", failures="failures", data=dataDIVAT2[1:200,],
  cov.quant=c("age"), cov.quali=c("hla", "retransplant", "ecd"),
  k=tune.model$optimal$k)

# The resulted predicted survival of the first subject of the training sample

plot(y=model$predictions[1,], x=model$times, xlab="Time (years)",
  ylab="Predicted survival", col=1, type="l", lty=1, lwd=2, ylim=c(0,1))

```

tunePLANN

*Tune a Survival Neural Network Based on the PLANN Method***Description**

This function finds the optimal inter, size, decay, maxit, and MaxNWts parameters for the survival neural network by using cross-validation and the concordance index.

**Usage**

```
tunePLANN(times, failures, group=NULL, cov.quant=NULL, cov.quali=NULL,
  data, cv=10, inter, size, decay, maxit, MaxNWts)
```

**Arguments**

times	The name of the variable related the numeric vector with the follow-up times.
failures	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
group	The name of the variable related to the exposure/treatment. This variable shall have only two modalities encoded 0 for the untreated/unexposed patients and 1 for the treated/exposed ones. The default value is NULL: no specific exposure/treatment is considered. When a specific exposure/treatment is considered, it will be forced in the algorithm or related interactions will be tested when possible.
cov.quant	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
cov.quali	The name(s) of the variable(s) related to the possible qualitative covariates. These variables must be numeric with two levels: 0 and 1. A complete disjunctive form must be used for covariates with more levels.

data	A data frame for training the model in which to look for the variables related to the status of the follow-up time (times), the event (failures), the optional treatment/exposure (group) and the covariables included in the previous model (cov.quant and cov.qual).
cv	The value of the number of folds. The default value is 10.
inter	The length of the intervals.
size	The number of units in the hidden layer.
decay	The parameter for weight decay.
maxit	The maximum number of iterations.
MaxNWts	The maximum allowable number of weights.

### Details

This function is based on the survivalPLANN package.

### Value

optimal	The value of inter, size, decay, maxit, and MaxNWts that gives the maximum mean cross-validated C-index.
results	The data frame with the mean cross-validated C-index according to inter, size, decay, maxit, and MaxNWts.

### References

Biganzoli E, Boracchi P, Mariani L, and et al. Feed forward neural networks for the analysis of censored survival data: a partial logistic regression approach. Stat Med, 17:1169-86, 1998.

### Examples

```
data(dataDIVAT2)

# The hyper-parameter grid needs to be more precise and the maximum number
# of iterations > 1000. We have reduced the arguments to respect examples requiring
# less than 5 seconds for packages on the CRAN.

tune.model <- tunePLANN(times="times", failures="failures", data=dataDIVAT2[1:300,],
  cov.quant=c("age"), cov.qual=c("hla", "retransplant", "ecd"), cv=3,
  inter=1, size=c(16, 32), decay=0.01, maxit=50, MaxNWts=10000)

tune.model$optimal # the optimal hyperparameters

tune.model$results # the C-index for the tested grid
```

---

tuneRSF	<i>Tune a Survival Random Forest</i>
---------	--------------------------------------

---

**Description**

This function finds the optimal nodesize, mtry, and ntree parameters for a survival random forest tree.

**Usage**

```
tuneRSF(times, failures, group=NULL, cov.quant= NULL,
cov.qual= NULL, data, nodesize, mtry, ntree)
```

**Arguments**

times	The name of the variable related the numeric vector with the follow-up times.
failures	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
group	The name of the variable related to the exposure/treatment. This variable shall have only two modalities encoded 0 for the untreated/unexposed patients and 1 for the treated/exposed ones. The default value is NULL: no specific exposure/treatment is considered. When a specific exposure/treatment is considered, it will be forced in the algorithm or related interactions will be tested when possible.
cov.quant	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
cov.qual	The name(s) of the variable(s) related to the possible qualitative covariates. These variables must be numeric with two levels: 0 and 1. A complete disjunctive form must be used for covariates with more levels.
data	A data frame for training the model in which to look for the variables related to the status of the follow-up time (times), the event (failures), the optional treatment/exposure (group) and the covariables included in the previous model (cov.quant and cov.qual).
nodesize	The values of the node size optimized over.
mtry	The numbers of variables randomly sampled as candidates at each split optimized over.
ntree	The numbers of trees optimized over.

**Details**

The function runs the `tune.rfsrc` function of the `randomForestSRC` package.

**Value**

optimal	The value of lambda that gives the minimum mean cross-validated error.
results	The data frame with the mean cross-validated errors for each lambda values.

## References

Ishwaran H. and Kogalur U.B. (2007). Random survival forests for R, *Rnews*, 7(2):25-31.

## Examples

```
data(dataDIVAT2)

tune.model <- tuneRSF(times="times", failures="failures", data=dataDIVAT2,
  cov.quanti=c("age"), cov.quali=c("hla", "retransplant", "ecd"),
  nodesize=c(100, 250, 500), mtry=1, ntree=100)

tune.model$optimal # the estimated nodesize value

# The estimation of the training model with the corresponding lambda value
model <- LIB_RSF(times="times", failures="failures", data=dataDIVAT2,
  cov.quanti=c("age"), cov.quali=c("hla", "retransplant", "ecd"),
  nodesize=tune.model$optimal$nodesize, mtry=1, ntree=100)

# The resulted predicted survival of the first subject of the training sample
plot(y=model$predictions[1,], x=model$times, xlab="Time (years)",
  ylab="Predicted survival", col=1, type="l", lty=1, lwd=2, ylim=c(0,1))
```

---

tuneSNN

*Tune a 1-Layer Survival Neural Network*

---

## Description

This function finds the optimal `n.nodes`, `decay`, `batch.size`, and `epochs` parameters for a survival neural network.

## Usage

```
tuneSNN(times, failures, group=NULL, cov.quanti=NULL, cov.quali=NULL,
  data, cv=10, n.nodes, decay, batch.size, epochs)
```

## Arguments

<code>times</code>	The name of the variable related the numeric vector with the follow-up times.
<code>failures</code>	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
<code>group</code>	The name of the variable related to the exposure/treatment. This variable shall have only two modalities encoded 0 for the untreated/unexposed patients and 1 for the treated/exposed ones. The default value is NULL: no specific exposure/treatment is considered. When a specific exposure/treatment is considered, it will be forced in the algorithm or related interactions will be tested when possible.
<code>cov.quanti</code>	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.



<code>cov.quali</code>	The name(s) of the variable(s) related to the possible qualitative covariates. These variables must be numeric with two levels: 0 and 1. A complete disjunctive form must be used for covariates with more levels.
<code>data</code>	A data frame for training the model in which to look for the variables related to the status of the follow-up time ( <code>times</code> ), the event ( <code>failures</code> ), the optional treatment/exposure ( <code>group</code> ) and the covariables included in the previous model ( <code>cov.quant</code> and <code>cov.quali</code> ).
<code>cv</code>	The value of the number of folds. The default value is 10.
<code>n.nodes</code>	The number of hidden nodes optimized over.
<code>decay</code>	The value of the weight decay optimized over.
<code>batch.size</code>	The value of batch size
<code>epochs</code>	The value of epochs

### Details

This function is based on the `deepsurv` from the `survivalmodels` package. You need to call Python using `reticulate`. In order to use it, the required Python packages must be installed with `reticulate::py_install`. Therefore, before running the present `LIB_SNN` function, you must install and call for the `reticulate` and `survivalmodels` packages, and install `pycox` by using the following command: `install_pycox(pip = TRUE, install_torch = FALSE)`. The `survivalSL` package functions without these supplementary installations if this learner is not included in the library.

### Value

<code>optimal</code>	The value of <code>n.nodes</code> , <code>decay</code> , <code>batch.size</code> , and <code>epochs</code> that gives the maximum mean cross-validated C-index.
<code>results</code>	The data frame with the mean cross-validated C-index according to <code>n.nodes</code> , <code>decay</code> , <code>batch.size</code> , and <code>epochs</code> .

### References

Katzman et al. DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network. *BMC Medical Research Methodology*, 18(1), 24. 1018.  
<https://doi.org/10.1186/s12874-018-0482-1>

# Index

- \* **Dataset**
  - dataDIVAT2, 2
  - dataDIVAT3, 3
  - dataOFSEP, 4
- \* **Metrics**
  - metrics, 29
  - roc, 38
- \* **Plotting**
  - plot.libsl, 30
  - plot.rocrisca, 32
  - plot.slttime, 33
- \* **Predicting**
  - predict.libsl, 34
  - predict.slttime, 35
- \* **Printing**
  - print.libsl, 36
  - print.slttime, 37
- \* **Summarizing**
  - summary.libsl, 39
  - summary.slttime, 41
- \* **Super Learner Library**
  - LIB\_AFTgamma, 5
  - LIB\_AFTgamma, 7
  - LIB\_AFTllogis, 8
  - LIB\_AFTweibull, 10
  - LIB\_COXaic, 11
  - LIB\_COXall, 13
  - LIB\_COXen, 14
  - LIB\_COXlasso, 16
  - LIB\_COXridge, 18
  - LIB\_PHexponential, 19
  - LIB\_PHgompertz, 21
  - LIB\_PHSpline, 22
  - LIB\_PLANN, 24
  - LIB\_RSf, 26
  - LIB\_SNN, 27
- \* **Super Learner**
  - survivalSL, 42
- \* **Tuning**
  - tuneCOXaic, 45
  - tuneCOXen, 47
  - tuneCOXlasso, 48
  - tuneCOXridge, 50
  - tunePHspline, 51
  - tunePLANN, 53
  - tuneRSF, 55
  - tuneSNN, 56
- dataDIVAT2, 2
- dataDIVAT3, 3
- dataOFSEP, 4
- LIB\_AFTgamma, 5, 36, 40
- LIB\_AFTgamma, 7, 36, 40
- LIB\_AFTllogis, 8, 36, 40
- LIB\_AFTweibull, 10, 36, 40
- LIB\_COXaic, 11
- LIB\_COXall, 13
- LIB\_COXen, 14
- LIB\_COXlasso, 16
- LIB\_COXridge, 18
- LIB\_PHexponential, 19, 36, 40
- LIB\_PHgompertz, 21, 36, 40
- LIB\_PHSpline, 22
- LIB\_PLANN, 24
- LIB\_RSf, 26
- LIB\_SNN, 27
- metrics, 29
- plot.default, 31–33
- plot.libsl, 30
- plot.rocrisca, 32
- plot.slttime, 33
- predict.libsl, 34
- predict.slttime, 35
- print.libsl, 36
- print.slttime, 37
- roc, 38

summary.libsl, [39](#)  
summary.sltime, [41](#)  
survivalSL, [36](#), [42](#), [42](#)

tuneCOXaic, [45](#)  
tuneCOXen, [47](#)  
tuneCOXlasso, [48](#)  
tuneCOXridge, [50](#)  
tunePHspline, [51](#)  
tunePLANN, [53](#)  
tuneRSF, [55](#)  
tuneSNN, [56](#)