Package 'RCASPAR'

October 8, 2014

Type Package

Title A package for survival time prediction based on a piecewise baseline hazard Cox regression model.

Version 1.10.0

Date 2010-08-23

Author Douaa Mugahid

Maintainer Douaa Mugahid <douaa.mugahid@gmail.com>, Lars Kaderali <lars.kaderali@bioquant.uni-heidelberg.de>

Description The package is the R-version of the C-based software \{}bold{CASPAR} (Kaderali,2006: \{}url{http://bioinformatics.oxfordjournals.org/content/22/12/1495}). It is meant to help predict survival times in the presence of high-dimensional explanatory covariates. The model is a piecewise baseline hazard Cox regression model with an Lq-norm based prior that selects for the most important regression coefficients, and in turn the most relevant covariates for survival analysis. It was primarily tried on gene expression and aCGH data, but can be used on any other type of high-dimensional data and in disciplines other than biology and medicine.

biocViews aCGH, GeneExpression, Genetics, Proteomics, Visualization

License GPL (>=3)

LazyLoad yes

R topics documented:

RCASPAR-package *A package for survival time prediction based on a piecewise baseline hazard Cox regression model.*

Description

The package is the R-version of the C-based software **CASPAR** (Kaderali,2006). It is meant to help predict survival times in the presence of high-dimensional explanatory co-variates. The model is a piecewise baseline hazard Cox regression model with an Lq-norm based prior that selects for the most important regression coefficients, and in turn the most relevant co-variates for survival analysis. It was primarily tried on gene expression and aCGH data, but can be used on any other type of high-dimensional data and in disciplines other than biology and medicine.

Details

Package:	RCASPAR
Type:	Package
Version:	1.0
Date:	2010-08-23
License: GPL(>=3) LazyLoad:	yes

Author(s)

Douaa Mugahid

Maintainer: Douaa Mugahid <mugahid@stud.uni-heidelberg.de>, Lars Kaderali <lars.kaderali@bioquant.uni-heidelberg.de>

References

The basic model is based on the Cox regression model as first introduced by Sir David Cox in: Cox,D.(1972).Regression models & life tables. *Journal of the Royal Society of Statistics*, 34(2), 187-220. The extension of the Cox model to its stepwise form was adapted from: Ibrahim, J.G, Chen, M.-H. & Sinha, D. (2005). *Bayesian Survival Analysis (second ed.)*. NY: Springer. as well as Kaderali, Lars.(2006) A Heirarchial Bayesian Approach to Regression and its Application to Predicting Survival Times in Cancer Patients. Aachen: Shaker The prior on the regression coefficients was adopted from: Mazur, J., Ritter,D.,Reinelt, G. & Kaderali, L. (2009). Reconstructing Non-Linear dynamic Models of Gene Regulation using Stochastic Sampling. *BMC Bioinformatics*, 10(448).

Examples

```
## Eg.(1): A simple example performed with a training and validation set:
data(Bergamaschi)
data(survData)
  ## Generate prediction:
result <- STpredictor_BLH(geDataS=Bergamaschi[1:27, 1:2], survDataS=survData[1:27, 9:10], geDataT=Bergamaschi[28
, cut.off=15, groups = 3, method = "CG", noprior = 1, extras = list(reltol=1))
 ## Plot a KM plot with both long and short survivors:
kmplt_svrl(long=result$long_survivors, short=result$short_survivors,title="KM plot of long and short survivors")
 ## Determine the area under the curve of AUROC curves vs. time to see the performance of the predictor given the cho
  ## and validation sets:
survivAURC(Stime=result$predicted_STs$True_STs,status=result$predicted_STs$censored, marker=result$predicted_ST
 ## Perform a log-rank test to see if the difference between the long and short survivors is significant:
logrnk(dataL=result$long_survivors, dataS=result$short_survivors)

## Eg.(2): A simple example performed with cross validation:
data(Bergamaschi)
data(survData)
  ## Generate prediction:
STpredictor_xvBLH(geData=Bergamaschi[1:40,1:2], survData=survData[1:40,9:10], k = 10, cut.off = 10, q = 1, s = 1, a
 ## Plot a KM plot with both long and short survivors:
kmplt_svrl(long=result$long_survivors, short=result$short_survivors,title="KM plot of long and short survivors")
 ## Determine the area under the curve of AUROC curves vs. time to see the performance of the predictor given the cho
  ## and validation sets:
survivAURC(Stime=result$predicted_STs$True_STs,status=result$predicted_STs$censored, marker=result$predicted_ST
 ## Perform a log-rank test to see if the difference between the long and short survivors is significant:
logrnk(dataL=result$long_survivors, dataS=result$short_survivors)
```

`Bergamaschi` *Gene expression data of 82 patients with 10 genes as covariates*

Description

A dataframe:\ "Bergamaschi:" A dataframe of 10 covariates (in the columns) for 82 patients (in the rows)\

Usage

```
data(Bergamaschi)
```

Format

The format is: num [1:82, 1:10] 0.654 0.701 0.126 0.899 0.267 ... - attr(*, "dimnames")=List of 2 ..$: chr [1:82] "1" "2" "3" "4" ... ..$: chr [1:10] "IMAGE:753234" "IMAGE:50794" "IMAGE:302190" "IMAGE:51408" ...

Details

A subset of the data set used in Bergamaschi, A., & al., e. (2006). Distinct Patterns of DNA Copy NumberAlteration are associated with different clinicopathological features and gene expression subtypes of breast cancer. Genes, Chromosomes and cancer , 45, 1033-1040.

Source

Bergamaschi, A., & al., e. (2006). Distinct Patterns of DNA Copy NumberAlteration are associated with different clinicopathological features and gene expression subtypes of breast cancer. Genes, Chromosomes and cancer , 45, 1033-1040.

References

Bergamaschi, A., & al., e. (2006). Distinct Patterns of DNA Copy NumberAlteration are associated with different clinicopathological features and gene expression subtypes of breast cancer. Genes, Chromosomes and cancer , 45, 1033-1040.

Examples

```
data(Bergamaschi)
colnames(Bergamaschi)
```

---

deriv_weight_estimator_BLH

A function that gives the derivative of the objective function of the model for gradient-based optimization algorithms.

Description

Given the necessary data, this function calculates the derivative of the objective function without a w.r.t. the baseline hazards and weights(regression coefficients) in the model to be used in gradient-based optimization algorithms.

Usage

```
deriv_weight_estimator_BLH(geDataT, survDataT, weights_baselineH, q, s, a, b, groups)
```

Arguments

geDataT	The co-variate data (gene expression or aCGH, etc...) of the patient set passed on by the user. It is a matrix with the co-variates in the columns and the subjects in the rows. Each cell corresponds to that row*th* subject's column*th* co-variate's value.
survDataT	The survival data of the patient set passed on by the user. It takes on the form of a data frame with at least have the following columns "True_STs" and "censored", corresponding to the observed survival times and the censoring status of the subjects consecutively. Censored patients are assigned a "1" while patients who experience an event are assigned "1".
weights_baselineH	A single vector with the initial values of the baseline hazards followed by the weights(regression coefficients) for the co-variates.
q	One of the two parameters on the prior distribution used on the weights (regression coefficients) in the model.
s	The second of the two parameters on the prior distribution used on the weights (regression coefficients) in the model.
a	The shape parameter for the gamma distribution used as a prior on the baseline hazards.
b	The scale parameter for the gamma distribution used as a prior on the baseline hazards.
groups	The number of partitions along the time axis for which a different baseline hazard is to be assigned. This number should be the same as the number of initial values passed for the baseline hazards in the beginning of the "weights_baselineH" argument.

Value

A vector of the same length as the "weights_baselineH" argument corresponding to the calculated derivatives of the objective with respect to every component of "weights_baselineH".

Note

This function is in itself not ver useful to the user, but is used within the function `weights_BLH`

Author(s)

Douaa Mugahid

References

The basic model is based on the Cox regression model as first introduced by Sir David Cox in: Cox,D.(1972).Regression models & life tables. *Journal of the Royal Society of Statistics*, 34(2), 187-220. The extension of the Cox model to its stepwise form was adapted from: Ibrahim, J.G, Chen, M.-H. & Sinha, D. (2005). *Bayesian Survival Analysis (second ed.)*. NY: Springer. as well as Kaderali, Lars.(2006) A Hierarchial Bayesian Approach to Regression and its Application to

Predicting Survival Times in Cancer Patients. Aachen: Shaker The prior on the regression coefficients was adopted from: Mazur, J., Ritter,D.,Reinelt, G. & Kaderali, L. (2009). Reconstructing Non-Linear dynamic Models of Gene Regulation using Stochastic Sampling. *BMC Bioinformatics*, 10(448).

See Also

weight_estimator_BLH, codederiv_weight_estimator_BLH_noprior

Examples

```
data(Bergamaschi)
data(survData)
deriv_weight_estimator_BLH(survDataT=survData[1:10, 9:10], geDataT=Bergamaschi[1:10, 1:2], weights_baselineH=c(0
```

`deriv_weight_estimator_BLH_noprior`

A function that gives the derivative of the objective function of the model for gradient-based optimization algorithms without including the prior on the regression coefficients.

Description

Given the necessary data, this function calculates the derivative of the objective function without a prior on the regression coefficients w.r.t. the baseline hazards and weights(regression coefficients) in the model to be used in gradient-based optimization algorithms. This is sometimes necessary to get the optimization algorithm out of a peaked origin where it could start.

Usage

```
deriv_weight_estimator_BLH_noprior(survDataT, geDataT, weights_baselineH, a, b, groups)
```

Arguments

`survDataT`	The survival data of the patient set passed on by the user. It takes on the form of a data frame with at least have the following columns "True_STs" and "censored", corresponding to the observed survival times and the censoring status of the subjects consecutively. Censored patients are assigned a "1" while patients who experience an event are assigned "1".
`geDataT`	The co-variate data (gene expression or aCGH, etc...) of the patient set passed on by the user. It is a matrix with the co-variates in the columns and the subjects in the rows. Each cell corresponds to that row*th* subject's column*th* co-variate's value.
`weights_baselineH`	A single vector with the initial values of the baseline hazards followed by the weights(regression coefficients) for the co-variates.

a	The shape parameter for the gamma distribution used as a prior on the baseline hazards.
b	The scale parameter for the gamma distribution used as a prior on the baseline hazards.
groups	The number of partitions along the time axis for which a different baseline hazard is to be assigned. This number should be the same as the number of initial values passed for the baseline hazards in the beginning of the "weights_baselineH" argument.

Value

A vector of the same length as the "weights_baselineH" argument corresponding to the calculated derivatives of the objective with respect to every component of "weights_baselineH".

Note

This function is in itself not very useful to the user, but is used within the function `weights_BLH`.

Author(s)

Douaa Mugahid

References

The basic model is based on the Cox regression model as first introduced by Sir David Cox in: Cox,D.(1972).Regression models & life tables. *Journal of the Royal Society of Statistics*, 34(2), 187-220. The extension of the Cox model to its stepwise form was adapted from: Ibrahim, J.G, Chen, M.-H. & Sinha, D. (2005). *Bayesian Survival Analysis (second ed.)*. NY: Springer. as well as Kaderali, Lars.(2006) A Hierarchial Bayesian Approach to Regression and its Application to Predicting Survival Times in Cancer Patients. Aachen: Shaker

See Also

`weight_estimator_BLH_noprior`, `deriv_weight_estimator_BLH`

Examples

```
data(Bergamaschi)
data(survData)
deriv_weight_estimator_BLH_noprior(survDataT=survData[1:10, 9:10], geDataT=Bergamaschi[1:10, 1:2], weights_basel
```

kmplt *Plot Kaplan Meier curve*

Description

This function plots the survival curve for the provided data set as a Kaplan Meier plot. It can only be used for visualization and returns no numeric values.

Usage

```
kmplt(data, title)
```

Arguments

`data`	A data frame containing *at least* the two columns "censored" and "True_STs". Where "censored" contains the censorship status of the subject as either "0/F" for 'uncensored subjects' or "1/T" for 'censored subjects'. This information is essential to be able to plot the KM curve.
`title`	A string of characters denoting the title of the plot produced.

Details

Note that this function was intended only for visualization and does not return any numerical values *as such*.

Value

A plot of the survival function "S(t)" against the survival times (unit-less) as calculated for the provided data set.

Author(s)

Douaa Mugahid

References

Bland JM, Altman DG. Survival probabilities (the Kaplan-Meier method). *BMJ* 1998;317;1572\
http://www.bmj.com/statsbk/12.dtl

Examples

```
censored <- c(1, 0, 1, 1, 1, 0, 1, 0, 0, 0)
True_STs <- c(1, 4, 5, 4, 6, 3, 2, 1, 3, 4)
dat <- as.data.frame(cbind(True_STs, censored))
kmplt(dat, "test")
```

`kmplt_svrl` *A function that plots the KM curves of $2-3$ patient sets in one graph.*

Description

This function can plot the KM curves estimated for $2-3$ patients simultaneously for sake of easier comparison.

Usage

```
kmplt_svrl(all, long, short, title)
```

Arguments

`all`	Data for the first set of patients; usually the complete set of patients, but could be any other.It is a data frame containing *at least* the two columns "censored" and "True_STs". Where "censored" contains the censorship status of the subject as either "0/F" for **uncensored subjects** or "1/T" for **censored subjects**. This information is essential to be able to plot the KM curve.
`long`	Data for the second set of patients; in our case the group of patients who survived at least up to the cut off value passed to the predictor. It has essentially the same structure as "all"
`short`	Data for the third and last set of patients; in our case the group of patients who survived less than the cut off value passed on to the predictor. It essentially has the same structure as the two other arguments.
`title`	The main title for the plot.

Details

This function essentially is the same as `kmplt` but does the same for up to 3 plots simultaneously.

Value

A plot with all $2-3$ KM curves.

Author(s)

Douaa Mugahid

References

http://www.bmj.com/statsbk/12.dtl

See Also

`kmplt`

Examples

```
censored <- c(1, 0, 1, 1, 1, 0, 1, 0, 0, 0)
True_STs <- c(1, 4, 5, 4, 6, 3, 2, 1, 3, 4)
dat1 <- as.data.frame(cbind(True_STs, censored))
censored <- c(1, 0, 1, 0, 1, 0, 1, 0, 1, 1)
True_STs <- c(7, 7, 8, 5, 9, 11, 8, 11, 10, 6)
dat2 <- as.data.frame(cbind(True_STs, censored))
censored <- c(1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1)
True_STs <- c(1, 4, 5, 4, 6, 3, 2, 1, 3, 4, 7, 7, 8, 5, 9, 11, 8, 11, 10, 6)
dat3 <- as.data.frame(cbind(True_STs, censored))
kmplt_svrl(all=dat3, long=dat2, short=dat1, title="KM of predictions")
```

---

logrnk *Performs Log Rank test on the long and short patient sets*

Description

This function performs a Chi-square test on the long and short subject sets to determine if their is a significant difference between the survival times in both sets. It returns the p-value.

Usage

```
logrnk(dataL, dataS)
```

Arguments

dataL	The set of subjects predicted to fall into the long-survivor set. A data frame containing at least the following columns: "PatientOrderValidation" (the number/order of the subject); "group" (the group into which the patient falls L (for long) or S(for short)); "censored" (the censorship status of the patient 1 for uncensored and 0 for censored).
dataS	Same as "dataL" but for the set of short survivors.

Details

Note that the typical arguments to be passed are the results of the "STpredict" functions "long_survivors" and "long_survivors"

Value

The estimated p-value is returned

Author(s)

Douaa AS Mugahid

References

Bland JM, Altman DG. Survival probabilities (the Kaplan-Meier method). *BMJ* 2004;328;1073 http://www.bmj.com/statsbk/12.dtl

See Also

survivAURC

Examples

```
PatientOrderValidation_L <- c(1, 2, 3, 5, 7)
PatientOrderValidation_S <- c(4, 6, 8)
group_L <- rep("L", 5)
group_S <- rep("S", 3)
censored_L <- c(0, 0, 1, 1, 0)
censored_S <- c(0, 0, 1)
True_STs_L <- c(5, 6, 6 ,7, 8)
True_STs_S <- c(2, 3, 2)
short <- as.data.frame(cbind(PatientOrderValidation_S, group_S, censored_S, True_STs_S))
long <- as.data.frame(cbind(PatientOrderValidation_L, group_L, censored_L, True_STs_L))
names(short) <- c("PatientOrderValidation", "group", "censored", "True_STs")
names(long) <- c("PatientOrderValidation", "group", "censored", "True_STs")
logrnk(dataL=long, dataS=short)
```

---

pltgamma *Plotting the gamma distribution of shape parameter*

Description

This function allows the user to visualize the shape of the gamma distribution used as a prior on the baseline hazards in the functions ending with "_BLH"

Usage

```
pltgamma(a, b)
```

Arguments

a	Is the shape parameter for the gamma distribution.
b	Is the scale parameter for the gamma distribution

Value

A plot of the gamma distribution with the given parameters.

Author(s)

Douaa Mugahid

See Also

pltprior

Examples

```
pltgamma(a=2,b=2)
```

`pltprior` *A function to visualize the shape of the prior on the weights with the chosen q and s parameters.*

Description

This function helps visualize the effect of the parameters q and s on the prior distribution used on the weights.

Usage

```
pltprior(q, s)
```

Arguments

q	One of the two parameters used to determine the prior on the weights.
s	The second of two parameters used to determine the prior on the weights.

Details

The prior assumed on the weights in the objective function takes on the following formulation:

$$[L_q(\beta; q, s) = \frac{\frac{q^{(q-1)}q}{2s\Gamma(\frac{1q}{)}}}{e} xp(\frac{-1qs^q}{|} weights|^q)$$

Value

A 3D plot of the value of the prior taking into consideration only two weights

Author(s)

Douaa Mugahid

References

Mazur J., et al. Reconstructing nonlinear dynamic models of gene regulation using stochastic sampling. *BMC Bioinformatics* 2009;10;448

See Also

pltgamma

Examples

```
pltprior(q=1,s=1)
```

`simpson` *A function that calculates the area under a curve based on the Simpson algorithm*

Description

A function that calculates the approximate value of the definite integral of a continuous function. In other words, it can help plot the area under the curve of the plotted function between two limits.

Usage

```
simpson(x, y)
```

Arguments

x	A vector of the values at which the function is to be plotted.
y	A vector with the values of the function at the corresponding x-values.

Value

A single numerical value of the approximate area under the curve generated with the x and y values.

Note

Compared to the trapezoidal algorithm, this is usually more accurate.

Author(s)

Douaa Mugahid

References

Hennion, P.E.(1962). Algorithm 84: Simpson's integration. Communications of ACM. 5(4), 208

See Also

`trapezoid`

Examples

```
x <- seq(0:20)
y <- seq(0, 100, 1)
simpson(x,y)
```

`STpredictor_BLH` *Predicts the survival times of the validation set based on the regression coefficients and baseline hazards determined according to the Piecewise baseline hazard Cox regression model.*

Description

This function uses the training set to estimated the best regression coefficients, and baseline hazards describing the data according to the piecewise baseline hazard Cox regression model. It then takes them and uses them to predict the survival times of the validation set, which are determined as the mean value of the p.d.f. of the survival time, as a continuous random variable, given the co-variate values of that subject.

Usage

```
STpredictor_BLH(geDataS, survDataS, cut.off, file = paste(getwd(), "STpredictor_results", sep = "/"), g
NULL, geneweights = NULL, method = "BFGS", noprior = 1, extras = list())
```

Arguments

`geDataS`	The co-variate data of the validation set passed on by the user. It is a matrix with the co-variates in the columns and the subjects in the rows. Each cell corresponds to that row*th* subject's column*th* co-variate's value.
`survDataS`	The survival data of the validation set passed on by the user. It takes on the form of a data frame with at least have the following columns "True_STs" and "censored", corresponding to the observed survival times and the censoring status of the subjects consecutively. Censored patients are assigned a "1" while patients who experience an event are assigned "1".
`cut.off`	The value of the separator around which the patients are grouped according to their predicted survival times.
`file`	The path of the file to which the log file of this session is saved.
`geDataT`	The co-variate data of the kth training set passed on by the user.
`survDataT`	The survival data of the kth training set passed on by the user.
`groups`	The number of partitions along the time axis for which a different baseline hazard is to be assigned. This number should be the same as the number of initial values passed for the baseline hazards in the beginning of the "weights_baselineH" argument.
`a`	The shape parameter for the gamma distribution used as a prior on the baseline hazards.
`b`	The scale parameter for the gamma distribution used as a prior on the baseline hazards.
`q`	One of the two parameters on the prior distribution used on the weights (regression coefficients) in the model.

`s` The second of the two parameters on the prior distribution used on the weights (regression coefficients) in the model.

`BLHs` A vector with the initial values for the baseline hazards. Should be of length *groups*. The default is NULL, in which case a vector of length `groups` with values corresponding to the maximum of the gamma distributions with the given parameters is created.

`geneweights` A vector with the initial values of the weights(regression coefficients) for the covariates. The default is NULL, in which case a vector of zeros the same length as `ncol(geData)` is created as the initial starting value.

`method` The preferred optimization method. It can be one of the following:\ `"Nelder-Mead"`: for the Nelder-Mead simplex algorithm.\ `"L-BFGS-B"`: for the L-BFGS-B quasi-Newtonian method.\ `"BFGS"`: for the BFGS quasi-Newtonian method.\ `"CG"`: for the Conjugate Gradient decent method\ `"SANN"`: for the simulated annealing algorithm.\

`noprior` An integer indicating the number of iterations to be done without assuming a prior on the regression coefficients.

`extras` The extra arguments to passed to the optimization function optim. For further details on them, see the documentation for the `optim` function.

Value

`log_optimization`
The result of the optimization performed on the training set as is described in the documentation for the `optim` function

`short_survivors`
A data frame of results for the patients living less than the cut off value; with the columns True_STs (the observed survival times), Predicted_STs (the predicted survival times), censored(the censoring status of the patient, absolute_error(the sign-less difference between the predicted and observed survival times), PatientOrderValidation (The patient's number)

`long_survivors` A data frame with the results for the patients living at least as long as the cut off value; with columns True_STs (the observed survival times), Predicted_STs (the predicted survival times), censored(the censoring status of the patient,absolute_error(the sign-less difference between the predicted and observed survival times), PatientOrderValidation (The patient's number)

Author(s)

Douaa Mugahid

References

The basic model is based on the Cox regression model as first introduced by Sir David Cox in: Cox,D.(1972).Regression models & life tables. *Journal of the Royal Society of Statistics*, 34(2), 187-220. The extension of the Cox model to its stepwise form was adapted from: Ibrahim, J.G, Chen, M.-H. & Sinha, D. (2005). *Bayesian Survival Analysis (second ed.)*. NY: Springer. as well as Kaderali, Lars.(2006) A Hierarchical Bayesian Approach to Regression and its Application to

Predicting Survival Times in Cancer Patients. Aachen: Shaker The prior on the regression coefficients was adopted from: Mazur, J., Ritter,D.,Reinelt, G. & Kaderali, L. (2009). Reconstructing Non-Linear dynamic Models of Gene Regulation using Stochastic Sampling. *BMC Bioinformatics*, 10(448).

See Also

STpredictor_xvBLH

Examples

```
data(Bergamaschi)
data(survData)
result <- STpredictor_BLH(geDataS=Bergamaschi[1:27, 1:2], survDataS=survData[1:27, 9:10], geDataT=Bergamaschi[28
s = 1, a = 1.558, b = 0.179, cut.off=3, groups = 3, method = "CG", noprior = 1, extras = list(reltol=1))
```

STpredictor_xvBLH *This function performs a cross validation on the full data set to help predict the survival times of the patients using the piecewise baseline hazard PH Cox model.*

Description

Using the full data provided by the user, this function splits the data set k times, into a smaller validation set, and a much bigger training set. The regression coefficients of the model are estimated from the training set and used to predict the survival times of the validation set. The patients can then be split into patients two groups around a cut off value also specified by the user.

Usage

```
STpredictor_xvBLH(geData, survData, k = 10, cut.off, file = paste(getwd(), "STpredictor.xv.BLH_results
, BLHs = NULL, method = "BFGS", noprior = 1, extras = list())
```

Arguments

geData	A matrix with the co-variate data of the full set of subjects. It is constructed with the co-variate in the columns and the subjects in the rows.Each cell corresponds to that row*th* subject's column *th* co-variate's value.
survData	The survival data of the entire set of subjects. It takes on the form of a data frame with at least have the following columns "True_STs" and "censored", corresponding to the observed survival times and the censoring status of the subjects consecutively. Censored patients are assigned a "1" while patients who experience an event are assigned "1".
k	The number of times the cross-validation is.
cut.off	The value of the separator around which the patients are grouped according to their predicted survival times.
file	The path of the file to which the log file of this session is saved.

q	One of the two parameters on the prior distribution used on the weights (regression coefficients) in the model.
s	The second of the two parameters on the prior distribution used on the weights (regression coefficients) in the model.
a	The shape parameter for the gamma distribution used as a prior on the baseline hazards.
b	The scale parameter for the gamma distribution used as a prior on the baseline hazards.
groups	The number of partitions along the time axis for which a different baseline hazard is to be assigned. This number should be the same as the number of initial values passed for the baseline hazards in the beginning of the "weights_baselineH" argument
geneweights	A vector with the initial values of the weights(regression coefficients) for the covariates. The default is NULL, in which case a vector of zeros the same length as `ncol(geData)` is created as the initial starting value.
BLHs	A vector with the initial values for the baseline hazards. Should be of length *groups*. The default is NULL, in which case a vector of length `groups` with values corresponding to the maximum of the gamma distributions with the given parameters is created.
method	The preferred optimization method. It can be one of the following: `"Nelder-Mead"`: for the Nelder-Mead simplex algorithm. `"L-BFGS-B"`: for the L-BFGS-B quasi-Newtonian method. `"BFGS"`: for the BFGS quasi-Newtonian method. `"CG"`: for the Conjugate Gradient decent method `"SANN"`: for the simulated annealing algorithm.
noprior	An integer indicating the number of iterations to be done without assuming a prior on the regression coefficients.
extras	The extra arguments to passed to the optimization function optim. For further details on them, see the documentation for the `optim` function.

Value

predicted_STs	A data frame of the results for all patients, with the columns True_STs (the observed survival times), Predicted_STs (the predicted survival times), censored(the censoring status of the patient,absolute_error(the signless difference between the predicted and oberved survival times), PatientOrderValidation (The patient's number)
short_survivors	A data frame of results for the patients living less than the cut off value; with the columns True_STs (the observed survival times), Predicted_STs (the predicted survival times), censored(the censoring status of the patient,absolute_error(the signless difference between the predicted and oberved survival times), PatientOrderValidation (The patient's number)
long_survivors	A data frame with the results for the patients living at least as long as the cut off value; with columns True_STs (the observed survival times), Predicted_STs (the predicted survival times), censored(the censoring status of the patient,absolute_error(the sign-less difference between the predicted and observed survival times), PatientOrderValidation (The patient's number)

weights	A vector with the mean value of the regression coefficients obtained from the k training sets
baselineHs	A vector with the mean value of the baseline hazards returned from the k training sets

Author(s)

Douaa Mugahid

References

The basic model is based on the Cox regression model as first introduced by Sir David Cox in: Cox,D.(1972).Regression models & life tables. *Journal of the Royal Society of Statistics*, 34(2), 187-220. The extension of the Cox model to its stepwise form was adapted from: Ibrahim, J.G, Chen, M.-H. & Sinha, D. (2005). *Bayesian Survival Analysis (second ed.)*. NY: Springer.// as well as Kaderali, Lars.(2006) A Hierarchial Bayesian Approach to Regression and its Application to Predicting Survival Times in Cancer Patients. Aachen: Shaker The prior on the regression coefficients was adopted from: Mazur, J., Ritter,D.,Reinelt, G. & Kaderali, L. (2009). Reconstructing Non-Linear dynamic Models of Gene Regulation using Stochastic Sampling. *BMC Bioinformatics*, 10(448).

See Also

STpredictor_BLH

Examples

```
data(Bergamaschi)
data(survData)
STpredictor_xvBLH(geData=Bergamaschi[1:20, 1:2], survData=survData[1:20, 9:10], k = 10, cut.off=3, file = paste(ge
groups = 3, geneweights = NULL, BLHs = NULL, method = "CG", noprior = 1, extras = list(reltol=1))
```

---

survData *Survial data of 82 patients*

Description

A dataframe "survData:" A dataframe containing the survival information of the 82 patients the relevant columns of which are "True_STs", "censored" indicating the recorded survival times of the patients, and their censorship status simultaneously. In the "censored" column "0" indicates a non-censored patient, and "1" indicates a censored patient\

Usage

```
data(survData)
```

Format

A data frame with 82 observations on the following 10 variables.

`Array.ID` The array IDs of the patients

`Sample.ID.Bauke.` The array IDs of the patients as annotated in one source

`Experiment.Jon.` The array IDs of the patients as annotated in a third source (in essence the same as the previous two)

`ExptID` The Experiment ID

`Order` The number/order of the patient

`Overall.survival..mons.undivided` the overall survival time of the patients in months

`Relapse.free.survival..mons.` the relapse free survival time of the patients in months

`Status.0.A..1.AWD..2.DOD..3.DOC` the status of the tumour (as abbreviated in the publication)

`censored` the status of censorship with "0" corresponding to a "non-censored" patient and "1" to a "censored patient"

`True_STs` The survival/censorship times of the patient in years

Details

A subset of the data set used in Bergamaschi, A., & al., e. (2006). Distinct Patterns of DNA Copy NumberAlteration are associated with different clinicopathological features and gene expression subtypes of breast cancer. Genes, Chromosomes and cancer , 45, 1033-1040.

Source

Bergamaschi, A., & al., e. (2006). Distinct Patterns of DNA Copy NumberAlteration are associated with different clinicopathological features and gene expression subtypes of breast cancer. Genes, Chromosomes and cancer , 45, 1033-1040.

References

Bergamaschi, A., & al., e. (2006). Distinct Patterns of DNA Copy NumberAlteration are associated with different clinicopathological features and gene expression subtypes of breast cancer. Genes, Chromosomes and cancer , 45, 1033-1040.

Examples

```
data(survData)
colnames(survData)
```

survivAURC *A function that calculates the area under a curve constructed from plotting the area under a ROC curve at the corresponding time point at which it was generated.*

Description

This function plots the individual area under the ROC curves of different time points against the time at which they were evaluated, and calculates the area under that curve.

Usage

```
survivAURC(Stime, status, marker, entry = NULL, cut.values = NULL, time.max = 20, by = 1)
```

Arguments

Stime	The observed survival times of the patients.
status	The censoring status of the patient. 1 for a censored patient, and 0 for a patient who has an event.
marker	The predicted survival time of the patients.
entry	The time of entry of the patients, set to NULL by default.
cut.values	The cut off values for which the ROC curves are to be constructed.
time.max	The maximum time point for which the area under a ROC curve is to be plotted.
by	The step size between every ROC curve estimated and the next.

Details

The calculations for the Area under the ROC curves at each time point is done according to Patrick Heagerty's survivalROC function in the survivalROC package in R. It is a value indicating the performance of the predictor when used on the results of both functions STpredictor.BLH and STpredictor.xv.BLH.

Value

AUC	The value of the area under the curve generated
AUeachROC	A vector with the values of the area under the individual ROC curves

A plot of the Area under the ROC curves against their corresponding time points.

Author(s)

Douaa Mugahid

References

Heagerty,P., Lumely T. & Pepe M.(2000). Time-dependent ROC curves for censored survival data & a diagnostic marker. Biometrics, 56(2), 337-344.

See Also

survivROC

Examples

```
True_STs <- c(1.416667,2.75,2.416667,2.583333,2.166667,2.5,2.5,1.833333,1.25,0.6666667,1,6.583333,6.5,6.666667,
3.416667,6.083333,1.833333,5.583333,0.75,5.75,5.5,0.5833333,7.666667,5,2.833333,1.333333,5.083333,0.8333333,1.
1.083333,3.75,3.25,3,2.416667,2.75,2.5,2.666667,4.5,4.416667,1.5,0.8333333,3.166667,3.833333,3.833333,0.416666
2.333333,3.416667,3.416667,3,0.6666667,0.75,2.166667,1,1.416667,1.333333,1.166667,1.166667,0.4166667,1.25,1.16
Predicted_STs <- c(6.030591,6.014457,3.545584,5.414229,6.41576,9.393992,5.542331,6.890859,8.090213,4.98545,2.77
11.12364,3.982502,5.441881,12.61404,12.21851,17.05850,12.78141,16.22795,21.48544,6.281354,13.83925,8.859929,6.
10.97276,15.63089,8.01967,5.817267,5.59897,4.340784,32.40319,33.74123,27.45024,26.31024,26.88833,24.34707,32.0
7.79996,5.601459,2.802408,35.77047,24.34717,30.65796,25.93927,20.64544,22.04807,19.15037,23.83430,1.876557,3.9
2.860931,2.541947,4.543111,4.525553,4.148272,3.986912,6.246755,6.89523)
censored <- c(0,1,1,1,1,0,1,1,0,1,0,1,1,1,1,0,0,0,0,1,1,1,1,0,1,0,1,1,0,1,1,0,0,1,0,0,1,0,1,0,1,1,0,0,1,1,1,0,1
1)
survivAUC(Stime=True_STs,status=censored, marker=Predicted_STs,time.max=5)
```

---

survivROC — *Generates the ROC curve at a given time point given the observed and predicted survival data in the presence of censored subjects.*

Description

The function generates the Receiver-Operator Curve (ROC) at the specified time point using predicted and observed data in the presence of censored subjects. It so plots (1 - specificity) against the (specificity) at the designated cut off points. It is based on Patrick Heagerty's survivalROC function in the survivalROC package.

Usage

```
survivROC(Stime, status, marker, entry = NULL, predict.time, cut.values = NULL, plot = TRUE)
```

Arguments

Stime	The observed survival times of the patients.
status	The censoring status of the patient. 1 for a censored patient, and 0 for a patient who has an event.
marker	The predicted survival time of the patients.
entry	The time of entry of the patients, set to NULL by default.
predict.time	The time point for which the ROC curve is to be plotted.
cut.values	The cut off values for which the ROC curves are to be constructed.
plot	A logical argument that specifies whether a plot is to be generated (TRUE) or not (FALSE). The argument is set to TRUE by default.

Details

This function is basically the survivROC function in Patrick Heagerty's survivROC package, with slight modifications to it to better suit our purpose. Unlike Heagerty's function it only performs the calculations using the KM estimator and does not provide any other methods as options.

Value

`cut.values`	the cut off values for which the sensitivity and (1-specificity) were calculated
`Comp.Specificity`	(1 - Specificity) as was calculated for every cut off value.
`Sensitivity`	The Sensitivity as was calculated for every cut off value
`predict.time`	The time point for which the ROC curve is calculated
`Survival`	The value of the estimated survivor function (using the KM estimator) at "predict.time"
`AUC`	The value of the area under the estimated ROC curve

Note

It is important to note that using the Kaplan-Meier estimator as the method for estimating the survival function results in non-monotonous curves in some instances.

Author(s)

Douaa Mugahid

References

Heagerty,P., Lumely T. & Pepe M.(2000). Time-dependent ROC curves for censored survival data & a diagnostic marker. Biometrics, 56(2), 337-344.

See Also

survivAURC

Examples

```
True_STs <- c(1.416667,2.75,2.416667,2.583333,2.166667,2.5,2.5,1.833333,1.25,0.6666667,1,6.583333,6.5,6.666667,
3.416667,6.083333,1.833333,5.583333,0.75,5.75,5.5,0.5833333,7.666667,5,2.833333,1.333333,5.083333,0.8333333,1.
1.083333,3.75,3.25,3,2.416667,2.75,2.5,2.666667,4.5,4.416667,1.5,0.8333333,3.166667,3.833333,3.833333,0.416666
2.333333,3.416667,3.416667,3,0.6666667,0.75,2.166667,1,1.416667,1.333333,1.166667,1.166667,0.4166667,1.25,1.16
Predicted_STs <- c(6.030591,6.014457,3.545584,5.414229,6.41576,9.393992,5.542331,6.890859,8.090213,4.98545,2.77
11.12364,3.982502,5.441881,12.61404,12.21851,17.05850,12.78141,16.22795,21.48544,6.281354,13.83925,8.859929,6.
10.97276,15.63089,8.01967,5.817267,5.59897,4.340784,32.40319,33.74123,27.45024,26.31024,26.88833,24.34707,32.0
7.79996,5.601459,2.802408,35.77047,24.34717,30.65796,25.93927,20.64544,22.04807,19.15037,23.83430,1.876557,3.9
2.860931,2.541947,4.543111,4.525553,4.148272,3.986912,6.246755,6.89523)
censored <- c(0,1,1,1,1,0,1,1,0,1,0,1,1,1,1,0,0,0,0,1,1,1,1,0,1,0,1,1,0,1,1,0,0,1,0,0,1,0,1,0,1,1,0,0,1,1,1,0,1
1)
survivROC(Stime=True_STs,status=censored, marker=Predicted_STs,predict.time=5)
```

`trapezoid` *A function that calculates the area under a curve based on the Simpson algorithm*

Description

A function that calculates the approximate value of the definite integral of a continuous function. In other words, it can help plot the area under the curve of the plotted function between two limits.

Usage

```
trapezoid(x, y)
```

Arguments

`x`	The values to be used along the x-axis while plotting the curve of the function. The *x* in f(x)=y
`y`	The values to be used along the y-axis while plotting the curve of the function. The *y* in the f(x)=y.

Value

The area under the curve plotted with the x and y values provided as arguments.

Note

Using this method is slightly less accurate than using the simpson integration method

Author(s)

Douaa Mugahid

References

Weisstein, Eric W. "Trapezoidal Rule." From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/Trapezo

See Also

`simpson`

Examples

```
x <- seq(0:20)
y <- seq(0, 100, 1)
trapezoid(x,y)
```

`weights_BLH` *Optimization for the regression coefficients and baseline hazards that maximize the partial likelihood in our PW Cox PH regression model.*

Description

This function is a wrapper around the optimization function `optim` to allow the optimization for the regression coefficients and baseline hazards appropriate for the data set at hand. It is where the functions `weight_estimator_BLH`, `weight_estimator_BLH_noprior`,`deriv_weight_estimator_BLH`, `deriv_weight_esti` are required.

Usage

```
weights_BLH(geDataT, survDataT, q, s, a, b, groups, par, method = c("Nelder-Mead", "L-BFGS-B", "CG", "BF
dist = NULL)
```

Arguments

`geDataT`	A matrix with the co-variate in the columns and the subjects in the rows.Each cell corresponds to that row*th* subject's column*th* co-variate's value.
`survDataT`	A data frame with the survival data of the set of subjects at hand. It should at least have the following columns "True_STs" and "censored", corresponding to the observed survival times and the censoring status of the subjects consecutively. Censored patients are assigned a "1" while patients who experience an event are assigned "1".
`q`	One of the two parameters on the prior distribution used on the weights (regression coefficients) in the model.
`s`	The second of the two parameters on the prior distribution used on the weights (regression coefficients) in the model.
`a`	The shape parameter for the gamma distribution used as a prior on the baseline hazards.
`b`	The scale parameter for the gamma distribution used as a prior on the baseline hazards.
`groups`	The number of partitions along the time axis for which a different baseline hazard is to be assigned. This number should be the same as the number of initial values passed for the baseline hazards in the beginning of the "weights_baselineH" argument.
`par`	A single vector with the initial values of the baseline hazards followed by the weights(regression coefficients) for the co-variates.
`method`	The preferred optimization method. It can be one of the following: `"Nelder-Mead"`: for the Nelder-Mead simplex algorithm. `"L-BFGS-B"`: for the L-BFGS-B quasi-Newtonian method. `"BFGS"`: for the BFGS quasi-Newtonian method. `"CG"`: for the Conjugate Gradient decent method `"SANN"`: for the simulated annealing algorithm.

noprior	An integer indicating the number of iterations to be done without assuming a prior on the regression coefficients.
extras	The extra arguments to passed to the optimization function optim. For further details on them, see the documentation for the optim function.
dist	The distribution function to be passed to the optimization algorithm in case of using SANN to generate a new candidate point.

Value

The same value as the optim function. See it's documentation for details.

Note

Note that this function is just a wrapper around the optim function to serve our purpose, and it's main purpose is to be called within the main functions of this package STpredictor_BLH and weights_xvBLH

Author(s)

Douaa Mugahid

References

http://sekhon.berkeley.edu/stats/html/optim.html

Examples

```
data(Bergamaschi)
data(survData)
weights_BLH(geDataT=Bergamaschi[1:10,1:2], survDataT=survData[1:10, 9:10], q=1, s=1, a=1.56, b=0.17, groups=3, pa
list(reltol=1), dist = NULL)
```

---

weights_xvBLH *A special version of STpredictor.BLH used within k-xv to predict the survival times of the kth validation group in the cross validation step.*

Description

This function is an "incomplete" version of STpredictor.BLH used within the cross validation function STpredictor_xvBLH to predicted the survival times of the subset of patients in the kth partitioning. It is not meant for use outside that function.

Usage

```
weights_xvBLH(geDataS, survDataS, geDataT, survDataT, q = 1, s = 1, a = 2, b = 2, groups = 3, par, method
```

Arguments

geDataS	The co-variate data of the kth validation set passed on by STpredictor.xv.BLH. It is a matrix with the co-variates in the columns and the subjects in the rows. Each cell corresponds to that row*th* subject's column*th* co-variate's value.
survDataS	The survival data of the kth validation set passed on by STpredictor_xvBLH. It takes on the form of a data frame with at least have the following columns "True_STs" and "censored", corresponding to the observed survival times and the censoring status of the subjects consecutively. Censored patients are assigned a "1" while patients who experience an event are assigned "1".
geDataT	The co-variate data of the kth training set passed on by STpredictor_xvBLH.
survDataT	The survival data of the kth training set passed on by STpredictor_xvBLH.
q	One of the two parameters on the prior distribution used on the weights (regression coefficients) in the model.
s	The second of the two parameters on the prior distribution used on the weights (regression coefficients) in the model.
a	The shape parameter for the gamma distribution used as a prior on the baseline hazards.
b	The scale parameter for the gamma distribution used as a prior on the baseline hazards.
groups	The number of partitions along the time axis for which a different baseline hazard is to be assigned. This number should be the same as the number of initial values passed for the baseline hazards in the beginning of the "weights_baselineH" argument.
par	A single vector with the initial values of the baseline hazards followed by the weights(regression coefficients) for the co-variates.
method	The preferred optimization method. It can be one of the following: "Nelder-Mead": for the Nelder-Mead simplex algorithm. "L-BFGS-B" for the L-BFGS-B quasi-Newtonian method. "BFGS" for the BFGS quasi-Newtonian method. "CG" for the Conjugate Gradient decent method. "SANN": for the simulated annealing algorithm.
noprior	An integer indicating the number of iterations to be done without assuming a prior on the regression coefficients.
extras	The extra arguments to passed to the optimization function optim. For further details on them, see the documentation for the optim function.

Value

prediction	A data frame with the columns True_STs (the observed survival times), Predicted_STs (the predicted survival times), censored(the censoring status of the patient,absolute_error(the sign-less difference between the predicted and observed survival times), PatientOrderValidation (The patient's number)
est.geneweight	The estimated regression coefficients from the kth training set (geDataT,survDataT)
est.baselineH	The estimated baseline hazards from the kth training set (geDataT, survDataT)

Note

This function is not meant to be used outside its wrapper.

Author(s)

Douaa Mugahid

See Also

STpredictor_BLH

Examples

```
data(Bergamaschi)
data(survData)
weights_xvBLH(geDataS=Bergamaschi[21:31, 1:2], survDataS=survData[21:31, 9:10],geDataT=Bergamaschi[1:20, 1:2],
survDataT=survData[1:20, 9:10], q = 1, s = 1, a = 2, b = 2, groups = 3, par = c(0.1, 0.1, 0.1,rep(0,2)),
method = "CG", noprior = 1, extras = list(reltol=1))
```

---

weight_estimator_BLH *Returns the value of the objective function used for optimizing for the regression parameters and baseline hazards in the model.*

Description

Given the arguments, it can evaluate the value of the objective function used by the optimization algorithms for determining the optimal regression parameters and baseline hazard values.

Usage

```
weight_estimator_BLH(survDataT, geDataT, weights_baselineH, q, s, a, b, groups)
```

Arguments

survDataT	The survival data of the patient set passed on by the user. It takes on the form of a data frame with at least have the following columns "True_STs" and "censored", corresponding to the observed survival times and the censoring status of the subjects consecutively. Censored patients are assigned a "1" while patients who experience an event are assigned "1".
geDataT	The co-variate data (gene expression or aCGH, etc...) of the patient set passed on by the user. It is a matrix with the co-variates in the columns and the subjects in the rows. Each cell corresponds to that row*th* subject's column*th* co-variate's value.
weights_baselineH	A single vector with the initial values of the baseline hazards followed by the weights(regression coefficients) for the co-variates.

q	One of the two parameters on the prior distribution used on the weights (regression coefficients) in the model.
s	The second of the two parameters on the prior distribution used on the weights (regression coefficients) in the model.
a	The shape parameter for the gamma distribution used as a prior on the baseline hazards.
b	The scale parameter for the gamma distribution used as a prior on the baseline hazards.
groups	The number of partitions along the time axis for which a different baseline hazard is to be assigned. This number should be the same as the number of initial values passed for the baseline hazards in the beginning of the "weights_baselineH" argument.

Value

A single numerical value corresponding to the value of the objective function with the given regression coefficients and baseline hazard values.

Note

This function is in itself not useful to the user, but is used within the function `weights.BLH`

Author(s)

Douaa Mugahid

References

The basic model is based on the Cox regression model as first introduced by Sir David Cox in: Cox,D.(1972).Regression models & life tables. *Journal of the Royal Society of Statistics*, 34(2), 187-220. The extension of the Cox model to its stepwise form was adapted from: Ibrahim, J.G, Chen, M.-H. & Sinha, D. (2005). *Bayesian Survival Analysis (second ed.)*. NY: Springer. as well as Kaderali, Lars.(2006) A Hierarchial Bayesian Approach to Regression and its Application to Predicting Survival Times in Cancer Patients. Aachen: Shaker The prior on the regression coefficients was adopted from: Mazur, J., Ritter,D.,Reinelt, G. & Kaderali, L. (2009). Reconstructing Non-Linear dynamic Models of Gene Regulation using Stochastic Sampling. *BMC Bioinformatics*, 10(448).

See Also

`weight_estimator_BLH_noprior`, `deriv_weight_estimator_BLH_noprior`

Examples

```
data(Bergamaschi)
data(survData)
weight_estimator_BLH(survDataT=survData[1:10, 9:10], geDataT=Bergamaschi[1:10 , 1:2],weights_baselineH=c(0.1,0.2
```

`weight_estimator_BLH_noprior` *Returns the value of the objective function used for optimizing for the regression parameters and baseline hazards in the model, without including the prior on the regression coefficients.*

Description

Given the arguments, it can evaluate the value of the objective function used by the optimization algorithms for determining the optimal regression parameters and baseline hazard values without including the prior on the regression coefficients, which can be necessary if the starting conditions are set to the origin, which is very peaked in case of inclusion of the prior distribution.

Usage

```
weight_estimator_BLH_noprior(geDataT, survDataT, weights_baselineH, a, b, groups)
```

Arguments

`survDataT`	The survival data of the patient set passed on by the user. It takes on the form of a data frame with at least have the following columns "True_STs" and "censored", corresponding to the observed survival times and the censoring status of the subjects consecutively. Censored patients are assigned a "1" while patients who experience an event are assigned "1".
`geDataT`	The co-variate data (gene expression or aCGH, etc...) of the patient set passed on by the user. It is a matrix with the co-variates in the columns and the subjects in the rows. Each cell corresponds to that row*th* subject's column*th* co-variate's value.
`weights_baselineH`	A single vector with the initial values of the baseline hazards followed by the weights(regression coefficients) for the co-variates.
`a`	The shape parameter for the gamma distribution used as a prior on the baseline hazards.
`b`	The scale parameter for the gamma distribution used as a prior on the baseline hazards.
`groups`	The number of partitions along the time axis for which a different baseline hazard is to be assigned. This number should be the same as the number of initial values passed for the baseline hazards in the beginning of the "weights_baselineH" argument.

Value

A vector of the same length as the "weights_baselineH" argument corresponding to the calculated derivatives of the objective with respect to every component of "weights_baselineH".

Note

This function is in itself not useful to the user, but is used within the function `weights.BLH`

Author(s)

Douaa Mugahid

References

The basic model is based on the Cox regression model as first introduced by Sir David Cox in: Cox,D.(1972).Regression models & life tables. *Journal of the Royal Society of Statistics*, 34(2), 187-220. The extension of the Cox model to its stepwise form was adapted from: Ibrahim, J.G, Chen, M.-H. & Sinha, D. (2005). *Bayesian Survival Analysis (second ed.)*. NY: Springer. as well as Kaderali, Lars.(2006) A Hierarchial Bayesian Approach to Regression and its Application to Predicting Survival Times in Cancer Patients. Aachen: Shaker

See Also

`weight_estimator_BLH`, `deriv_weight_estimator_BLH_noprior`

Examples

```
data(Bergamaschi)
data(survData)
weight_estimator_BLH_noprior(geDataT=Bergamaschi[1:10, 1:2], survDataT=survData[1:10, 9:10], weights_baselineH=c
```

Index