

Package: MicrobiomeSurv (via r-universe)

September 1, 2024

Type Package

Title A Biomarker Validation Approach for Classification and Predicting Survival Using Microbiome Data

Version 0.1.0

Maintainer Thi Huyen Nguyen <thihuyen.nguyen@uhasselt.be>

Description An approach to identify microbiome biomarker for time to event data by discovering microbiome for predicting survival and classifying subjects into risk groups. Classifiers are constructed as a linear combination of important microbiome and treatment effects if necessary. Several methods were implemented to estimate the microbiome risk score such as majority voting technique, LASSO, Elastic net, supervised principle component analysis (SPCA), and supervised partial least squares analysis (SPLS). Sensitivity analysis on the quantile used for the classification can also be accessed to check the deviation of the classification group based on the quantile specified. Large scale cross validation can be performed in order to investigate the mostly selected microbiome and for internal validation. During the evaluation process, validation is accessed using the hazard ratios (HR) distribution of the test set and inference is mainly based on resampling and permutations technique.

URL <https://github.com/N-T-Huyen/MicrobiomeSurv>

BugReports <https://github.com/N-T-Huyen/MicrobiomeSurv/issues/new>

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

Imports graphics, stats, ggplot2, survival, survminer, glmnet, methods, superpc, lmtest, gplots, tidyr, dplyr, microbiome, pls, grDevices

Suggests knitr, rmarkdown

VignetteBuilder knitr
Depends R (>= 2.10)
Repository <https://n-t-huyen.r-universe.dev>
RemoteUrl <https://github.com/n-t-huyen/microbiomesurv>
RemoteRef HEAD
RemoteSha ae3e5454878c291fe2cb39e44e6e65a65d961e42

Contents

CoxPHUni	3
CVLasoelascox	4
cvle-class	7
CVMajorityvotes	8
cvmm-class	10
CVMSpecificCoxPh	12
cvmv-class	14
CVPcaPls	15
cvpp-class	17
cvsit-class	18
CVSITaxa	19
data_zero_per_group_otu_w3	21
DistHR	22
EstimateHR	24
fam_info_w3	26
fam_shan_trim_w3	27
FirstFilter	28
GetRA	29
hello	30
Lasoelascox	30
Majorityvotes	33
metadata_taxonomy	34
MiFreq	35
ms-class	36
MSpecificCoxPh	38
perm-class	40
QuantileAnalysis	41
SecondFilter	43
SITaxa	44
SummaryData	46
SurvPcaClass	47
SurvPlsClass	49
Top1Uni	52
Week3_otu	53
Week3_response	54
ZerosPerGroup	55

CoxPHUni	<i>This function will fit the full and reduced models and calculate LRT raw p-value and adjusted p-value based on BH Method</i>
----------	---

Description

This function will fit the full and reduced models and calculate LRT raw p-value and adjusted p-value based on BH Method

Usage

```
CoxPHUni(Survival, Censor, Prognostic, Micro.mat, Method = "BH")
```

Arguments

Survival	The time to event outcome.
Censor	An indicator variable indicate the subject is censored or not.
Prognostic	A dataframe containing possible prognostic(s) factor and/or treatment effect to be used in the model.
Micro.mat	a microbiome matrix, can be at otu, family or any level of the ecosystem. Rows are taxa, columns are subjectsc.
Method	A multiplicity adjustment Method that user can choose. The default is BH Method.

Value

A relative abundance matrix of OTUs

coef	coefficient of one microbiome (OTU or family, ...)
exp.coef	exponential of the coefficient
p.value.LRT	raw LRT p-value
p.value	adjusted p-value based on chosen Method

Author(s)

Thi Huyen Nguyen, <thihuyen.nguyen@uhasselt.be>

Olajumoke Evangelina Owokotomo, <olajumoke.x.owokotomo@gsk.com>

Ziv Shkedy

See Also

[CoxPHUni](#)

Examples

```
# Prepare data
data(Week3_response)
Week3_response = data.frame(Week3_response)
surv_fam_shan_w3 = data.frame(cbind(as.numeric(Week3_response$T1Dweek),
as.numeric(Week3_response$T1D)))
colnames(surv_fam_shan_w3) = c("Survival", "Censor")
prog_fam_shan_w3 = data.frame(factor(Week3_response$Treatment_new))
colnames(prog_fam_shan_w3) = c("Treatment")
data(fam_shan_trim_w3)
names_fam_shan_trim_w3 =
c("Unknown", "Lachnospiraceae", "S24.7", "Lactobacillaceae", "Enterobacteriaceae", "Rikenellaceae")
fam_shan_trim_w3 = data.matrix(fam_shan_trim_w3[,2:82])
rownames(fam_shan_trim_w3) = names_fam_shan_trim_w3
# Using the funtion
summary_fam_shan_w3 = CoxPHUni(Survival = surv_fam_shan_w3$Survival,
                               Censor = surv_fam_shan_w3$Censor,
                               Prognostic = prog_fam_shan_w3,
                               Micro.mat = fam_shan_trim_w3,
                               Method = "BH")
```

CVLasoeLascox

Cross Validations for Lasso Elastic Net Survival predictive models and Classification

Description

The function does cross validation for Lasso, Elastic net and Ridge regressions models before the survival analysis and classification. The survival analysis is based on the selected taxa in the presence or absence of prognostic factors.

Usage

```
CVLasoeLascox(
  Survival,
  Censor,
  Micro.mat,
  Prognostic,
  Standardize = TRUE,
  Alpha = 1,
  Fold = 4,
  Ncv = 10,
  nlambda = 100,
  Mean = TRUE,
  Quantile = 0.5
)
```

Arguments

Survival	A vector of survival time with length equals to number of subjects.
Censor	A vector of censoring indicator.
Micro.mat	A large or small microbiome profile matrix. A matrix with microbiome profiles where the number of rows is equal to the number of taxa and number of columns is equal to number of patients.
Prognostic	A dataframe containing possible prognostic(s) factor and/or treatment effect to be used in the model.
Standardize	A Logical flag for the standardization of the microbiome matrix, prior to fitting the model sequence. The coefficients are always returned on the original scale. Default is standardize=TRUE.
Alpha	The mixing parameter for glmnet (see glmnet). The range is $0 \leq \text{Alpha} \leq 1$. The Default is 1.
Fold	Number of folds to be used for the cross validation. Its value ranges between 3 and the number of subjects in the dataset.
Ncv	Number of validations to be carried out. The default is 10.
nlambda	The number of lambda values - default is 100 as in glmnet.
Mean	The cut off value for the classifier, default is the mean cutoff.
Quantile	If users want to use quantile as cutoff point. They need to specify Mean = FALSE and a quantile that they wish to use. The default is the median cutoff.

Details

The function performs the cross validations for Lasso, Elastic net and Ridge regressions models for Cox proportional hazard model. Taxa are selected at each iteration and then use for the classifier. Which implies that predictive taxa is varied from one cross validation to the other depending on selection. The underline idea is to investigate the Hazard Ratio for the train and test data based on the optimal lambda selected for the non-zero shrinkage coefficients, the nonzero selected taxa will thus be used in the survival analysis and in calculation of the risk scores for each sets of data.

Value

A object of class `cvle` is returned with the following values

Coef.mat	A matrix of coefficients with rows equals to number of cross validations and columns equals to number of taxa.
lambda	A vector of estimated optimum lambda for each iterations.
n	A vector of the number of selected taxa.
HRTrain	A matrix of survival information for the training dataset. It has three columns representing the estimated HR, the 95% lower confidence interval and the 95% upper confidence interval.
HRTTest	A matrix of survival information for the test dataset. It has three columns representing the estimated HR, the 95% lower confidence interval and the 95% upper confidence interval.

p1d	A vector of partial likelihood deviance at each cross validations.
Mi.mat	A matrix with 0 and 1. Number of rows equals to number of iterations and number of columns equals to number of 1 taxon indicates that the particular taxon was selected or had nonzero coefficient and otherwise it is zero.
Micro.mat	The Microbiome data matrix that was used for the analysis either same as Mdata or a reduced version.

Author(s)

Thi Huyen Nguyen, <thihuyen.nguyen@uhasselt.be>

Olajumoke Evangelina Owokotomo, <olajumoke.x.owokotomo@gsk.com>

Ziv Shkedy

See Also

[coxph](#), [EstimateHR](#), [glmnet](#), [LasoeLascox](#)

Examples

```
# Prepare data
data(Week3_response)
Week3_response = data.frame(Week3_response)
surv_fam_shan_w3 = data.frame(cbind(as.numeric(Week3_response$T1Dweek),
as.numeric(Week3_response$T1D)))
colnames(surv_fam_shan_w3) = c("Survival", "Censor")
prog_fam_shan_w3 = data.frame(factor(Week3_response$Treatment_new))
colnames(prog_fam_shan_w3) = c("Treatment")
data(fam_shan_trim_w3)
names_fam_shan_trim_w3 =
c("Unknown", "Lachnospiraceae", "S24.7", "Lactobacillaceae", "Enterobacteriaceae", "Rikenellaceae")
fam_shan_trim_w3 = data.matrix(fam_shan_trim_w3[,2:82])
rownames(fam_shan_trim_w3) = names_fam_shan_trim_w3

# Using the function
CV_lasso_fam_shan_w3 = CVLasoeLascox(Survival = surv_fam_shan_w3$Survival,
                                     Censor = surv_fam_shan_w3$Censor,
                                     Micro.mat = fam_shan_trim_w3,
                                     Prognostic = prog_fam_shan_w3,
                                     Standardize = TRUE,
                                     Alpha = 1,
                                     Fold = 4,
                                     Ncv = 10,
                                     nlambda = 100)

# Number of selected taxa per CV
CV_lasso_fam_shan_w3@n

# Get the matrix of coefficients
CV_lasso_fam_shan_w3@Coef.mat

# Survival information of the train dataset
```

```

CV_lasso_fam_shan_w3@HRTrain

# Survival information of the test dataset
CV_lasso_fam_shan_w3@HRTTest

```

cvle-class *The cvle Class.*

Description

Class of object returned by function [CVLasoelascox](#).

Usage

```

## S4 method for signature 'cvle'
show(object)

## S4 method for signature 'cvle'
summary(object)

## S4 method for signature 'cvle,missing'
plot(x, y, type = 1, ...)

```

Arguments

object	A cvle class object
x	A cvle class object
y	missing
type	Plot type. 1 distribution of the HR under training and test set. 2 HR vs number selected taxa.
...	The usual extra arguments to generic functions — see plot , plot.default

Slots

Coef.mat A matrix of coefficients with rows equals to number of cross validations and columns equals to number of taxa,

lambda A vector of estimated optimum lambda for each iterations.

n A vector of the number of selected taxa.

mi.mat A matrix with 0 and 1. Number of rows equals to number of iterations and number of columns equals to number of taxa. 1 indicates that the particular taxon was selected or had nonzero coefficient and otherwise it is zero.

HRTrain A matrix of survival information for the training dataset. It has three columns representing the estimated HR, the 95% lower confidence interval and the 95% upper confidence interval.

HRTest A matrix of survival information for the test dataset. It has three columns representing the estimated HR, the 95% lower confidence interval and the 95% upper confidence interval.

pId A vector of partial likelihood deviance at each cross validations.

Micro.mat The microbiome matrix that was used for the analysis which can either be the full the full data or a reduced supervised PCA version.

Author(s)

Thi Huyen Nguyen, <thihuyen.nguyen@uhasselt.be>

Olajumoke Evangelina Owokotomo, <olajumoke.x.owokotomo@gsk.com>

Ziv Shkedy

See Also

[EstimateHR](#), [glmnet](#), [Lasoelascox](#)

CVMajorityvotes

Cross validation for majority votes

Description

This function does cross validation for the Majority votes based classification which is a cross validated approach to [Majorityvotes](#).

Usage

```
CVMajorityvotes(
  Survival,
  Censor,
  Prognostic = NULL,
  Micro.mat,
  Reduce = TRUE,
  Select = 5,
  Fold = 3,
  Ncv = 100,
  Mean = TRUE,
  Quantile = 0.5
)
```

Arguments

Survival A vector of survival time with length equals to number of subjects.

Censor A vector of censoring indicator.

Prognostic A dataframe containing possible prognostic(s) factor and/or treatment effect to be used in the model.

Micro.mat	A large or small microbiome profile matrix. A matrix with microbiome profiles where the number of rows should be equal to the number of taxa and number of columns should be equal to number of patients.
Reduce	A boolean parameter indicating if the microbiome profile matrix should be reduced, default is TRUE and larger microbiome profile matrix is reduced by supervised pca approach.
Select	Number of taxa (default is 5) to be selected from supervised PCA. This is valid only if the argument Reduce=TRUE.
Fold	Number of times in which the dataset is divided. Default is 3 which implies dataset will be divided into three groups and 2/3 of the dataset will be the train dataset and 1/3 will be to train the results.
Ncv	The Number of cross validation loop. Default is 100.
Mean	The cut off value for the classifier, default is the mean cutoff.
Quantile	If users want to use quantile as cutoff point. They need to specify Mean = FALSE and a quantile that they wish to use. The default is the median cutoff.

Value

A object of class `cvmv` is returned with the following values

HRTrain	A matrix of survival information for the training dataset. It has three columns representing the estimated HR, the 95% lower confidence interval and the 95% upper confidence interval.
HRTest	A matrix of survival information for the test dataset. It has three columns representing the estimated HR, the 95% lower confidence interval and the 95% upper confidence interval.
Ncv	The number of cross validation used.
Micro.mat	The microbiome data matrix that was used for the analysis either same as Micro.mat or a reduced version.
Progfact	The names of prognostic factors used.

Author(s)

Thi Huyen Nguyen, <thihuyen.nguyen@uhasselt.be>

Olajumoke Evangelina Owokotomo, <olajumoke.x.owokotomo@gsk.com>

Ziv Shkedy

See Also

[Majorityvotes](#)

Examples

```

# Prepare data
data(Week3_response)
Week3_response = data.frame(Week3_response)
surv_fam_shan_w3 = data.frame(cbind(as.numeric(Week3_response$T1Dweek),
as.numeric(Week3_response$T1D)))
colnames(surv_fam_shan_w3) = c("Survival", "Censor")
prog_fam_shan_w3 = data.frame(factor(Week3_response$Treatment_new))
colnames(prog_fam_shan_w3) = c("Treatment")
data(fam_shan_trim_w3)
names_fam_shan_trim_w3 =
c("Unknown", "Lachnospiraceae", "S24.7", "Lactobacillaceae", "Enterobacteriaceae", "Rikenellaceae")
fam_shan_trim_w3 = data.matrix(fam_shan_trim_w3[,2:82])
rownames(fam_shan_trim_w3) = names_fam_shan_trim_w3
# Using the function
CVMajority_fam_shan_w3 = CVMajorityvotes(Survival = surv_fam_shan_w3$Survival,
                                         Micro.mat = fam_shan_trim_w3,
                                         Censor = surv_fam_shan_w3$Censor,
                                         Reduce=TRUE,
                                         Select=5,
                                         Mean = TRUE,
                                         Prognostic = prog_fam_shan_w3,
                                         Fold=3,
                                         Ncv=10)

# Get the class of the object
class(CVMajority_fam_shan_w3)      # An "cvmv" Class

# Method that can be used for the result
show(CVMajority_fam_shan_w3)
summary(CVMajority_fam_shan_w3)
plot(CVMajority_fam_shan_w3)

```

cvmm-class

The cvmm Class.

Description

Class of object returned by function [CVMspecificCoxPh](#).

Usage

```

## S4 method for signature 'cvmm'
show(object)

## S4 method for signature 'cvmm'
summary(object, which = 1)

```

```
## S4 method for signature 'cvmm,ANY'
plot(x, y, which = 1, ...)
```

Arguments

object	A CVMSpecificCoxPh class object
which	This specify which taxon for which estimated HR information need to be visualized. By default results of the first taxon is used.
x	A CVMSpecificCoxPh class object CVMSpecificCoxPh
y	missing
...	The usual extra arguments to generic functions — see plot , plot.default

Details

plot signature(x = "cvmm"): Plots for [CVMSpecificCoxPh](#) class analysis results.
Any parameters of [plot.default](#) may be passed on to this particular plot method.

Slots

HRTrain A 3-way array, The first dimension is the number of taxa, the second dimension is the HR statistics for the low risk group in the train dataset (HR,1/HR LCI, UCI) while the third dimension is the number of cross validation performed.

HRTTest A 3-way array, The first dimension is the number of taxa, the second dimension is the HR statistics for the low risk group in the test dataset (HR,1/HR LCI, UCI) while the third dimension is the number of cross validation performed.

train The selected subjects for each CV in the train dataset.

test The selected subjects for each CV in the test dataset.

n.mi The number of taxa used in the analysis.

Ncv The number of cross validation performed.

Rdata The microbiome data matrix that was used for the analysis either same as Micro.mat or a reduced version

Author(s)

Thi Huyen Nguyen, <thihuyen.nguyen@uhasselt.be>
Olajumoke Evangelina Owokotomo, <olajumoke.x.owokotomo@gsk.com>
Ziv Shkedy

See Also

[CVMSpecificCoxPh](#)

CVMSpecificCoxPh

*Cross validation for the Taxon specific analysis***Description**

The function performs cross validation for each taxon depending the number of fold which guides the division into the train and testing dataset. The classifier is then obtained on the training dataset to be validated on the test dataset.

Usage

```
CVMSpecificCoxPh(
  Fold = 3,
  Survival,
  Micro.mat,
  Censor,
  Reduce = TRUE,
  Select = 5,
  Prognostic = NULL,
  Mean = TRUE,
  Quantile = 0.5,
  Ncv = 100
)
```

Arguments

Fold	Number of times in which the dataset is divided. Default is 3 which implies dataset will be divided into three groups and 2/3 of the dataset will be the train dataset and 1/3 will be to test the results.
Survival	A vector of survival time with length equals to number of subjects
Micro.mat	A large or small microbiome profile matrix. A matrix with microbiome profiles where the number of rows should be equal to the number of taxa and number of columns should be equal to number of patients.
Censor	A vector of censoring indicator.
Reduce	A boolean parameter indicating if the microbiome profile matrix should be reduced, default is TRUE and larger microbiome profile matrix is reduced by supervised pca approach and first pca is extracted from the reduced matrix to be used in the classifier.
Select	Number of taxa (default is 5) to be selected from supervised PCA. This is valid only if th argument Reduce=TRUE.
Prognostic	A dataframe containing possible prognostic(s) factor and/or treatment effect to be used in the model.
Mean	The cut off value for the classifier, default is the mean cutoff.
Quantile	If users want to use quantile as cutoff point. They need to specify Mean = FALSE and a quantile that they wish to use. The default is the median cutoff.
Ncv	The Number of cross validation loop. Default is 100.

Details

This function performs the cross validation for taxon by taxon analysis. The data will firstly be divided into data train dataset and test dataset. Furthermore, a taxon-specific model is fitted on train data and a classifier is built. In addition, the classifier is then evaluated on test dataset for each particular taxon. The Process is repeated for all the full or reduced taxa to obtain the HR statistics of the low risk group. The following steps depends on the number of cross validation specified.

Value

A object of class `cvmm` is returned with the following values.

HRTrain	The Train dataset HR statistics for each taxon by the number of CV.
HRTTest	The Test dataset HR statistics for each taxon by the number of CV.
train	The selected subjects for each CV in the train dataset.
test	The selected subjects for each CV in the test dataset.
n.mi	The number of taxa used in the analysis.
Ncv	The number of cross validation performed.
Rdata	The Microbiome data matrix that was used for the analysis either same as <code>Micro.mat</code> or a reduced version.

Author(s)

Thi Huyen Nguyen, <thihuyen.nguyen@uhasselt.be>

Olajumoke Evangelina Owokotomo, <olajumoke.x.owokotomo@gsk.com>

Ziv Shkedy

See Also

[coxph](#), [EstimateHR](#), [MSpecificCoxPh](#),

Examples

```
# Prepare data
data(Week3_response)
Week3_response = data.frame(Week3_response)
surv_fam_shan_w3 = data.frame(cbind(as.numeric(Week3_response$T1Dweek),
as.numeric(Week3_response$T1D)))
colnames(surv_fam_shan_w3) = c("Survival", "Censor")
prog_fam_shan_w3 = data.frame(factor(Week3_response$Treatment_new))
colnames(prog_fam_shan_w3) = c("Treatment")
data(fam_shan_trim_w3)
names_fam_shan_trim_w3 =
c("Unknown", "Lachnospiraceae", "S24.7", "Lactobacillaceae", "Enterobacteriaceae", "Rikenellaceae")
fam_shan_trim_w3 = data.matrix(fam_shan_trim_w3[,2:82])
rownames(fam_shan_trim_w3) = names_fam_shan_trim_w3
# Using the function
CVCox_taxon_fam_shan_w3 = CVMSpecificCoxPh(Fold=3,
Survival = surv_fam_shan_w3$Survival,
```

```

Micro.mat = fam_shan_trim_w3,
Censor = surv_fam_shan_w3$Censor,
Reduce=TRUE,
Select=5,
Prognostic=prog_fam_shan_w3,
Mean = TRUE,
Ncv=10)

# Get the class of the object
class(CVCox_taxon_fam_shan_w3) # An "cvmv" Class

# Method that can be used for the result
show(CVCox_taxon_fam_shan_w3)
summary(CVCox_taxon_fam_shan_w3)
plot(CVCox_taxon_fam_shan_w3)

```

cvmv-class

The cvmv Class.

Description

Class of object returned by function [CVMajorityvotes](#).

Usage

```

## S4 method for signature 'cvmv'
show(object)

## S4 method for signature 'cvmv'
summary(object)

## S4 method for signature 'cvmv,ANY'
plot(x, y, ...)

```

Arguments

object	A cvmv class object
x	A cvmv class object
y	missing
...	The usual extra arguments to generic functions — see plot , plot.default

Slots

HRTrain A matrix of survival information for the training dataset. It has three columns representing the estimated HR, the 95% lower confidence interval and the 95% upper confidence interval.

HRTest A matrix of survival information for the test dataset. It has three columns representing the estimated HR, the 95% lower confidence interval and the 95% upper confidence interval.

Ncv The number of cross validation used.

Micro.mat The microbiome data matrix that was used for the analysis either same as Micro.mat or a reduced version.

Progfact The names of prognostic factors used.

Author(s)

Thi Huyen Nguyen, <thihuyen.nguyen@uhasselt.be>

Olajumoke Evangelina Owokotomo, <olajumoke.x.owokotomo@gsk.com>

Ziv Shkedy

See Also

[Majorityvotes](#), [CVPcaPls](#), [SurvPcaClass](#), [SurvPlsClass](#)

CVPcaPls

Cross Validations for PCA and PLS based methods

Description

This function does cross validation for the analysis performs by [SurvPcaClass](#) and [SurvPlsClass](#) functions where the dimension reduction methods can either be PCA and PLS.

Usage

```
CVPcaPls(
  Fold = 3,
  Survival,
  Micro.mat,
  Censor,
  Reduce = TRUE,
  Select = 15,
  Prognostic = NULL,
  Ncv = 5,
  DR = "PCA"
)
```

Arguments

Fold	Number of times in which the dataset is divided. Default is 3 which implies dataset will be divided into three groups and 2/3 of the dataset will be the train dataset and 1/3 will be to test the results.
Survival	A vector of survival time with length equals to number of subjects.
Micro.mat	A large or small microbiome profile matrix. A matrix with microbiome profiles where the number of rows should be equal to the number of taxa and number of columns should be equal to number of patients.

Censor	A vector of censoring indicator.
Reduce	A boolean parameter indicating if the microbiome profile matrix should be reduced, default is TRUE and larger microbiome profile matrix is reduced by supervised pca approach and first pca is extracted from the reduced matrix to be used in the classifier.
Select	Number of taxa (default is 5) to be selected from supervised PCA. This is valid only if the argument Reduce=TRUE.
Prognostic	A dataframe containing possible prognostic(s) factor and/or treatment effect to be used in the model.
Ncv	The Number of cross validation loop. Default is 100.
DR	The dimension reduction method. It can be either "PCA" for Principle components analysis or "PLS" for Partial least squares.

Details

This function does cross validation for the analysis using two reduction method. The reduction method can be PCA or PLS. If it is PCA then the [SurvPcaClass](#) is internally used for the cross validation and [SurvPlsClass](#) otherwise.

Value

A object of class [cvpp](#) is returned with the following values

Result	A dataframe containg the estimated Hazard ratio of the test dataset and the training dataset.
Ncv	The number of cross validation performed.
Method	The dimesion reduction method used.
CVtrain	The training dataset indices matrix used for the cross validation.
CVtest	The test dataset indices matrix used for the cross validation.
Select	The number of taxa used for the dimesion reduction method used.

Author(s)

Thi Huyen Nguyen, <thihuyen.nguyen@uhasselt.be>

Olajumoke Evangelina Owokotomo, <olajumoke.x.owokotomo@gsk.com>

Ziv Shkedy

See Also

[SurvPlsClass](#), [SurvPcaClass](#)

Examples

```

# Prepare data
data(Week3_response)
Week3_response = data.frame(Week3_response)
surv_fam_shan_w3 = data.frame(cbind(as.numeric(Week3_response$T1Dweek),
as.numeric(Week3_response$T1D)))
colnames(surv_fam_shan_w3) = c("Survival", "Censor")
prog_fam_shan_w3 = data.frame(factor(Week3_response$Treatment_new))
colnames(prog_fam_shan_w3) = c("Treatment")
data(fam_shan_trim_w3)
names_fam_shan_trim_w3 =
c("Unknown", "Lachnospiraceae", "S24.7", "Lactobacillaceae", "Enterobacteriaceae", "Rikenellaceae")
fam_shan_trim_w3 = data.matrix(fam_shan_trim_w3[,2:82])
rownames(fam_shan_trim_w3) = names_fam_shan_trim_w3
# Using the function
CVPls_fam_shan_w3 = CVPcaPls(Fold = 3,
                             Survival = surv_fam_shan_w3$Survival,
                             Micro.mat = fam_shan_trim_w3,
                             Censor = surv_fam_shan_w3$Censor,
                             Reduce=TRUE,
                             Select=5,
                             Prognostic = prog_fam_shan_w3,
                             Ncv=10,
                             DR = "PLS")

# Get the class of the object
class(CVPls_fam_shan_w3) # An "cvpp" Class

# Method that can be used for the result
show(CVPls_fam_shan_w3)
summary(CVPls_fam_shan_w3)
plot(CVPls_fam_shan_w3)

```

cvpp-class

The cvpp Class.

Description

Class of object returned by function [CVPcaPls](#).

Usage

```

## S4 method for signature 'cvpp'
show(object)

## S4 method for signature 'cvpp'
summary(object)

```

```
## S4 method for signature 'cvpp,missing'
plot(x, y, ...)
```

Arguments

object	A cvpp class object
x	A cvpp class object
y	missing
...	The usual extra arguments to generic functions — see plot , plot.default

Slots

Results A dataframe containing the estimated Hazard ratio of the test dataset and the training dataset

Ncv The number of cross validation performed

Method The dimension reduction method used

CVtrain The training dataset indices matrix used for the cross validation

CVtest The test dataset indices matrix used for the cross validation

Select The number of taxa used for the dimension reduction method used

Author(s)

Thi Huyen Nguyen, <thihuyen.nguyen@uhasselt.be>
 Olajumoke Evangelina Owokotomo, <olajumoke.x.owokotomo@gsk.com>
 Ziv Shkedy

See Also

[CVPcaPls](#), [SurvPcaClass](#), [SurvPlsClass](#)

cvsit-class

The cvsit Class.

Description

Class of object returned by function [cvsit](#).

Usage

```
## S4 method for signature 'cvsit'
show(object)

## S4 method for signature 'cvsit'
summary(object)

## S4 method for signature 'cvsit,missing'
plot(x, y, type = 1, ...)
```

Arguments

object	A cvsit class object
x	A cvsit class object
y	missing
type	Plot type. 1 distribution of the HR under test For the Top K taxa using PCA. 2 distribution of the HR under test For the Top K taxa using PLS.
...	The usual extra arguments to generic functions — see plot , plot.default

Slots

HRpca A 3-way array in which first, second, and third dimensions correspond to number of taxa, Hazard ratio information (Estimated HR, LowerCI and UpperCI), and number of cross validation respectively. This contains the estimated HR on test data and dimension reduction method is PCA.

HRpls A 3-way array in which first, second, and third dimensions correspond to number of taxa, Hazard ratio information (Estimated HR, LowerCI and UpperCI), and number of cross validation respectively. This contains the estimated HR on test data and dimension reduction method is PLS.

Ntaxa The number of taxa in the reduced matrix.

Ncv The number of cross validation done.

Top A sequence of top k taxa considered. Default is Top=seq(5,100,by=5).

Author(s)

Thi Huyen Nguyen, <thihuyen.nguyen@uhasselt.be>

Olajumoke Evangelina Owokotomo, <olajumoke.x.owokotomo@gsk.com>

Ziv Shkedy

See Also

[CVPcaPls](#), [SurvPcaClass](#), [SurvPlsClass](#)

CVSITaxa

Cross validation for sequentially increases taxa

Description

This function does cross validation for the taxon by taxon analysis while sequentially increasing the number of taxa as specified.

Usage

```
CVSITaxa(
  Object,
  Top = seq(5, 100, by = 5),
  Survival,
  Censor,
  Prognostic = NULL
)
```

Arguments

Object	An object of class <code>cvmm</code> .
Top	The Top k number of taxa to be used.
Survival	A vector of survival time with length equals to number of subjects.
Censor	A vector of censoring indicator.
Prognostic	A dataframe containing possible prognostic(s) factor and/or treatment effect to be used in the model.

Details

The function is a cross validation version of the function `SITaxa`. This function firstly processes the cross validation for the taxon by taxon analysis results, and then sequentially considers top k taxa. The function recompute first PCA or PLS on train data and estimate risk scores on both test and train data only on the microbiome matrix with top k taxa. Patients are then classified as having low or high risk based on the test data where the cutoff used is mean of the risk score. The process is repeated for each top K taxa sets.

Value

A object of class `cvsit` is returned with the following values

HRpca	A 3-way array in which first, second, and third dimensions correspond to number of taxa, Hazard ratio information (Estimated HR, LowerCI and UpperCI), and number of cross validation respectively. This contains the estimated HR on test data and dimension reduction method is PCA.
HRpls	A 3-way array in which first, second, and third dimensions correspond to number of taxa, Hazard ratio information (Estimated HR, LowerCI and UpperCI), and number of cross validation respectively. This contains the estimated HR on test data and dimension reduction method is PLS.
Ntaxa	The number of taxa in the reduced matrix.
Ncv	The number of cross validation done.
Top	A sequence of top k taxa considered. Default is <code>Top = seq(5, 100, by=5)</code>

Author(s)

Thi Huyen Nguyen, <thihuyen.nguyen@uhasselt.be>
 Olajumoke Evangelina Owokotomo, <olajumoke.x.owokotomo@gsk.com>
 Ziv Shkedy

See Also

[MSpecificCoxPh](#), [SITaxa](#)

Examples

```
# Prepare data
data(Week3_response)
Week3_response = data.frame(Week3_response)
surv_fam_shan_w3 = data.frame(cbind(as.numeric(Week3_response$T1Dweek),
as.numeric(Week3_response$T1D)))
colnames(surv_fam_shan_w3) = c("Survival", "Censor")
prog_fam_shan_w3 = data.frame(factor(Week3_response$Treatment_new))
colnames(prog_fam_shan_w3) = c("Treatment")
data(fam_shan_trim_w3)
names_fam_shan_trim_w3 =
c("Unknown", "Lachnospiraceae", "S24.7", "Lactobacillaceae", "Enterobacteriaceae", "Rikenellaceae")
fam_shan_trim_w3 = data.matrix(fam_shan_trim_w3[,2:82])
rownames(fam_shan_trim_w3) = names_fam_shan_trim_w3
# Getting the cvmm object
CVCox_taxon_fam_shan_w3 = CVMSpecificCoxPh(Fold=3,
                                           Survival = surv_fam_shan_w3$Survival,
                                           Micro.mat = fam_shan_trim_w3,
                                           Censor = surv_fam_shan_w3$Censor,
                                           Reduce=TRUE,
                                           Select=5,
                                           Prognostic=prog_fam_shan_w3,
                                           Mean = TRUE,
                                           Ncv=10)

# Using the function
CVSITaxa_fam_shan_w3 = CVSITaxa(Object = CVCox_taxon_fam_shan_w3,
                                Top=seq(1, 6, by=2),
                                Survival = surv_fam_shan_w3$Survival,
                                Censor = surv_fam_shan_w3$Censor,
                                Prognostic=prog_fam_shan_w3)

# Get the class of the object
class(CVSITaxa_fam_shan_w3) # An "cvsit" Class
```

data_zero_per_group_otu_w3

Zero per treatment groups.

Description

A dataset containing the information of zeros per treatment groups at OTU level.

Usage

```
data(data_zero_per_group_otu_w3)
```

Format

A data frame with 2720 rows and 10 variables:

OTU Name of OTUs

zero.ctrl Number of zeros in control group

propzero.ctrl Percentage of zeros in the control group

nCtrl Number of subjects in the control group

zero.Treated Number of zeros in treated group

propzero.Treated Percentage of zeros in the treated group

nTreated Number of subjects in the treated group

zero.total Number of zeros in total

propzero.total Percentage of zeros in total

nTotal Number of subjects in the experiment

Source

<https://github.com/N-T-Huyen>

DistHR

Null Distribution of the Estimated HR

Description

This function generates the null distribution of the HR by permutation approach either using a large microbiome matrix or a reduced version by supervised pca approach. Several ways of permutation setting can be implemented. That is, the function can be used to generate null distributions for four different validation schemes which are PLS based, PCA based, Majority votes based and Lasso based. Note this function internally calls function [SurvPcaClass](#), [SurvPlsClass](#), [Majorityvotes](#), and [Lasoelascox](#).

Usage

```
DistHR(  
  Survival,  
  Censor,  
  Micro.mat,  
  Prognostic = NULL,  
  Mean = TRUE,  
  Quantile = 0.5,  
  Reduce = FALSE,
```

```

Select = 5,
nperm = 100,
case = 2,
Method = "BH",
Validation = c("PLSbased", "PCAbased", "L1based", "MVbased")
)

```

Arguments

Survival	A vector of survival time with length equals to number of subjects.
Censor	A vector of censoring indicator.
Micro.mat	A large or small microbiome profile matrix. A matrix with microbiome profiles where the number of rows should be equal to the number of taxa and number of columns should be equal to number of patients.
Prognostic	A dataframe containing possible prognostic(s) factor and/or treatment effect to be used in the model.
Mean	The cut off value for the classifier, default is the mean cutoff.
Quantile	If user want to use quantile as cutoff point. They need to specify Mean = FALSE and a quantile that they want to use. The default is the median cutoff.
Reduce	A boolean parameter indicating if the microbiome profile matrix should be reduced, default is TRUE and larger microbiome profile matrix is reduced by supervised pca approach and first pca is extracted from the reduced matrix to be used in the classifier.
Select	Number of taxa (default is 5) to be selected from supervised PCA. This is valid only if the argument Reduce=TRUE.
nperm	Number of permutations to be used and default 100.
case	There are seven different ways on how to call this argument: <ol style="list-style-type: none"> 1. Permute survival only. 2. Permute survival and rows of data frame of the prognostic factors. 3. Permute survival, rows of data frame of the prognostic factors, columns of microbiome matrix independently. 4. Permute microbiome matrix only.
Method	A multiplicity adjustment Method that user can choose. The default is BH Method.
Validation	There are four different validation schemes where the null distribution can be estimated. That is c("PLSbased", "PCAbased", "L1based", "MVbased").

Value

A object of class `perm` is returned with the following values

HRobs	Estimated HR for low risk group on the original data.
HRperm	Estimated HR for low risk group on the permuted data.
nperm	Number of permutations carried out.
Validation	The validation scheme that was used.

Author(s)

Thi Huyen Nguyen, <thihuyen.nguyen@uhasselt.be>

Olajumoke Evangelina Owokotomo, <olajumoke.x.owokotomo@gsk.com>

Ziv Shkedy

See Also

[coxph](#), [EstimateHR](#), [SurvPcaClass](#), [SurvPlsClass](#), [Majorityvotes](#), [Lasoelascox](#)

Examples

```
# Prepare data
data(Week3_response)
Week3_response = data.frame(Week3_response)
surv_fam_shan_w3 = data.frame(cbind(as.numeric(Week3_response$T1Dweek),
as.numeric(Week3_response$T1D)))
colnames(surv_fam_shan_w3) = c("Survival", "Censor")
prog_fam_shan_w3 = data.frame(factor(Week3_response$Treatment_new))
colnames(prog_fam_shan_w3) = c("Treatment")
data(fam_shan_trim_w3)
names_fam_shan_trim_w3 =
c("Unknown", "Lachnospiraceae", "S24.7", "Lactobacillaceae", "Enterobacteriaceae", "Rikenellaceae")
fam_shan_trim_w3 = data.matrix(fam_shan_trim_w3[,2:82])
rownames(fam_shan_trim_w3) = names_fam_shan_trim_w3
# Using the function
DistHR_fam_shan_w3 = DistHR(Survival = surv_fam_shan_w3$Survival,
                           Micro.mat = fam_shan_trim_w3,
                           Censor = surv_fam_shan_w3$Censor,
                           Prognostic=prog_fam_shan_w3,
                           Mean = TRUE,
                           Quantile=0.5,
                           Reduce= FALSE,
                           Select = 5,
                           nperm=100,
                           case=4,
                           Method = "BH",
                           Validation="PCAbased")

# Method that can be used for the result
show(DistHR_fam_shan_w3)
summary(DistHR_fam_shan_w3)
plot(DistHR_fam_shan_w3)
```

Description

The function classifies subjects into Low and High risk groups using the risk scores based on the cut-off point which is the mean of the risk score. Also visualize survival fit along with HR estimates.

Usage

```
EstimateHR(
  Risk.Scores,
  Data.Survival,
  Prognostic = NULL,
  Plots = FALSE,
  Mean = TRUE,
  Quantile = 0.5
)
```

Arguments

Risk.Scores	A vector of risk scores with size equals to number of subjects obtained from (Lasoelascx).
Data.Survival	A dataframe in which the first column is the Survival and the second column is the Censoring indicator for each subject.
Prognostic	A dataframe containing possible prognostic(s) factor and/or treatment effect
Plots	A boolean parameter indicating if plots should be shown. Default is FALSE.
Mean	The cut off value for the classifier, default is the mean cutoff
Quantile	If user want to use quantile as cutoff point. They need to specify Mean = FALSE and a quantile that they want to use. The default is the median cutoff

Details

The risk scores obtained using the taxa is then used to generate the risk group by dividing subjects into low and high risk groups. A Cox model is then fitted with the risk group as covariate in the presence or absence of prognostic factors and or treatment effect. The extent of survival in the risk groups is known

Value

An object of is returned, which is a list with the results of the cox regression and some informative plot concerning survival of the risk group.

SurvResult	The cox proportional regression result
Riskgroup	The riskgroup based on the riskscore and the cut off value and length is equal to number of subjects
KMplot	The Kaplan-Meier survival plot of the riskgroup
SurvBPlot	The distribution of the survival in the riskgroup

Author(s)

Thi Huyen Nguyen, <thihuyen.nguyen@uhasselt.be>
 Olajumoke Evangelina Owokotomo, <olajumoke.x.owokotomo@gsk.com>
 Ziv Shkedy

See Also

[coxph](#), [Lasoelascox](#)

Examples

```
# Prepare data
data(Week3_response)
Week3_response = data.frame(Week3_response)
surv_fam_shan_w3 = data.frame(cbind(as.numeric(Week3_response$T1Dweek),
as.numeric(Week3_response$T1D)))
colnames(surv_fam_shan_w3) = c("Survival", "Censor")
prog_fam_shan_w3 = data.frame(factor(Week3_response$Treatment_new))
colnames(prog_fam_shan_w3) = c("Treatment")
data(fam_shan_trim_w3)
names_fam_shan_trim_w3 =
c("Unknown", "Lachnospiraceae", "S24.7", "Lactobacillaceae", "Enterobacteriaceae", "Rikenellaceae")
fam_shan_trim_w3 = data.matrix(fam_shan_trim_w3[,2:82])
rownames(fam_shan_trim_w3) = names_fam_shan_trim_w3
# Obtaining the risk score and data survival
lasso_fam_shan_w3 = Lasoelascox(Survival = surv_fam_shan_w3$Survival,
                               Censor = surv_fam_shan_w3$Censor,
                               Micro.mat = fam_shan_trim_w3,
                               Prognostic = prog_fam_shan_w3,
                               Plots = TRUE,
                               Standardize = TRUE,
                               Alpha = 1,
                               Fold = 4,
                               nlambda = 100,
                               Mean = TRUE)

# Using the function
est_HR_fam_shan_w3 = EstimateHR(Risk.Scores = lasso_fam_shan_w3$Risk.Scores,
                                Data.Survival = lasso_fam_shan_w3$Data.Survival,
                                Prognostic = prog_fam_shan_w3, Plots = TRUE,
                                Mean = TRUE)
```

fam_info_w3

Information at family level.

Description

A dataset containing the information at family level.

Usage

```
data(fam_info_w3)
```

Format

A data frame with 2720 rows and 2 variables:

OTUID ID of OTU

Family Family name

Source

<https://github.com/N-T-Huyen>

fam_shan_trim_w3 *Dataset at family level.*

Description

A dataset containing the Shannon index of 6 families after filtering.

Usage

```
data(fam_shan_trim_w3)
```

Format

A data frame with 6 rows and 82 variables:

Rows are family names and columns are names of subjects.

Source

<https://github.com/N-T-Huyen>

FirstFilter	<i>This function is used for the first step of filtering which removes OTUs having all zeros (inactive OTUs). The input is an OTU matrix with rows are OTUs and columns are subjects.</i>
-------------	---

Description

This function is used for the first step of filtering which removes OTUs having all zeros (inactive OTUs). The input is an OTU matrix with rows are OTUs and columns are subjects.

Usage

```
FirstFilter(Micro.mat)
```

Arguments

Micro.mat	A large or small microbiome matrix. A matrix with microbiome profiles where the number of rows should be equal to the number of taxa and number of columns should be equal to number of patients.
-----------	---

Value

A smaller microbiome matrix.

Micro.mat.trim The OTU matrix after removing all inactive OTUs

Author(s)

Thi Huyen Nguyen, <thihuyen.nguyen@uhasselt.be>

Olajumoke Evangelina Owokotomo, <olajumoke.x.owokotomo@gsk.com>

Ziv Shkedy

See Also

[FirstFilter](#)

Examples

```
# Preparing data for analysis at OTU level
data(Week3_otu)
Week3_otu = data.frame(Week3_otu)
otu_mat_w3 = t(data.matrix(Week3_otu[ , 1:2720]))
colnames(otu_mat_w3) = Week3_otu$SampleID
# Filtering first step
otu_w3 = FirstFilter(Micro.mat = otu_mat_w3)
```

GetRA

This function convert OTU matrix to RA matrix.

Description

This function convert OTU matrix to RA matrix.

Usage

```
GetRA(Micro.mat)
```

Arguments

Micro.mat an OTU matrix with OTUs in rows and subjects in columns.

Value

A relative abundance matrix of OTUs

ra Relative abundance matrixs

Author(s)

Thi Huyen Nguyen, <thihuyen.nguyen@uhasselt.be>

Olajumoke Evangelina Owokotomo, <olajumoke.x.owokotomo@gsk.com>

Ziv Shkedy

See Also

[GetRA](#)

Examples

```
# Read dataset
data(Week3_otu)
Week3_otu = data.frame(Week3_otu)
otu_mat_w3 = t(data.matrix(Week3_otu[ , 1:2720]))

# Convert absolute abundance to relative abundance
ra_otu_trim_w3 = GetRA(Micro.mat = otu_mat_w3)
```

hello	<i>Hello, World!</i>
-------	----------------------

Description

Prints 'Hello, world!'.

Usage

```
hello()
```

Examples

```
hello()
```

Lasoelascox	<i>Wrapper function for glmnet</i>
-------------	------------------------------------

Description

The function uses the glmnet function to firstly do the variable selection either with Lasso, Elastic net or ridge regressions before the survival analysis. The survival analysis is based on the selected taxa in the presence or absence of prognostic factors.

Usage

```
Lasoelascox(  
  Survival,  
  Censor,  
  Micro.mat,  
  Prognostic,  
  Plots = FALSE,  
  Standardize = TRUE,  
  Alpha = 1,  
  Fold = 4,  
  nlambda = 100,  
  Mean = TRUE,  
  Quantile = 0.5  
)
```

Arguments

Survival	A vector of survival time with length equals to number of subjects
Censor	A vector of censoring indicator
Micro.mat	A large or small microbiome matrix. A matrix with microbiome profiles where the number of rows is equal to the number of taxa and number of columns is equal to number of patients.
Prognostic	A dataframe containing possible prognostic(s) factor and/or treatment effect to be used in the model.
Plots	A boolean parameter indicating if plots should be shown. Default is FALSE. If TRUE, the first plot is the partial likelihood deviance against the logarithm of each lambda while the second is the coefficients versus the lambdas
Standardize	A Logical flag for the standardization of the microbiome matrix, prior to fitting the model sequence. The coefficients are always returned on the original scale. Default is standardize=TRUE.
Alpha	The mixing parameter for glmnet (see glmnet). The range is $0 \leq \text{Alpha} \leq 1$. The Default is 1
Fold	number of folds to be used for the cross validation. Its value ranges between 3 and the number of subjects in the dataset
nlambda	The number of lambda values - default is 100 as in glmnet.
Mean	The cut off value for the classifier, default is the mean cutoff
Quantile	If user want to use quantile as cutoff point. They need to specify Mean = FALSE and a quantile that they want to use. The default is the median cutoff

Details

This is a wrapper function for glmnet and it fits models using either Lasso, Elastic net and Ridge regressions. This is done in the presence or absence of prognostic factors. The prognostic factor when available will always be forced to be in the model so no penalty for it. Optimum lambda will be used to select the non-zero shrinkage coefficients, the nonzero selected taxa will thus be used in the survival analysis and in calculation of the risk scores.

Value

A object is returned with the following values

Coefficients.NonZero	The coefficients of the selected taxa
Selected.Mi	The selected taxa
n	The number of selected taxa
Risk.scores	The risk scores of the subjects
Risk.group	The risk classification of the subjects based on the specified cutoff point
SurvFit	The cox analysis of the riskgroup based on the selected taxa and the prognostic factors

Author(s)

Thi Huyen Nguyen, <thihuyen.nguyen@uhasselt.be>
 Olajumoke Evangelina Owokotomo, <olajumoke.x.owokotomo@gsk.com>
 Ziv Shkedy

See Also

[coxph](#)
[coxph](#), [EstimateHR](#), [glmnet](#),

Examples

```
# Prepare data
data(Week3_response)
Week3_response = data.frame(Week3_response)
surv_fam_shan_w3 = data.frame(cbind(as.numeric(Week3_response$T1Dweek),
as.numeric(Week3_response$T1D)))
colnames(surv_fam_shan_w3) = c("Survival", "Censor")
prog_fam_shan_w3 = data.frame(factor(Week3_response$Treatment_new))
colnames(prog_fam_shan_w3) = c("Treatment")
data(fam_shan_trim_w3)
names_fam_shan_trim_w3 =
c("Unknown", "Lachnospiraceae", "S24.7", "Lactobacillaceae", "Enterobacteriaceae", "Rikenellaceae")
fam_shan_trim_w3 = data.matrix(fam_shan_trim_w3[,2:82])
rownames(fam_shan_trim_w3) = names_fam_shan_trim_w3
# Using the function
lasso_fam_shan_w3 = Lasoelascox(Survival = surv_fam_shan_w3$Survival,
                               Censor = surv_fam_shan_w3$Censor,
                               Micro.mat = fam_shan_trim_w3,
                               Prognostic = prog_fam_shan_w3,
                               Plots = TRUE,
                               Standardize = TRUE,
                               Alpha = 1,
                               Fold = 4,
                               nlambda = 100,
                               Mean = TRUE)

# View the selected taxa
lasso_fam_shan_w3$Selected.mi

# Number of selected taxa
lasso_fam_shan_w3$n

# View the classification group of each subject
lasso_fam_shan_w3$Risk.Group

# View the survival analysis result
lasso_fam_shan_w3$SurvFit
```

Majorityvotes	<i>Classification for Majority Votes</i>
---------------	--

Description

The Function fits cox proportional hazard model and does classification based on the majority votes.

Usage

```
Majorityvotes(Result, Prognostic, Survival, Censor, J = 1)
```

Arguments

Result	An object obtained from the taxon specific analysis (MSpecificCoxPh) which is of class "ms"
Prognostic	A dataframe containing possible prognostic(s) factor and/or treatment effect to be used in the model.
Survival	A vector of survival time with length equals to number of subjects
Censor	A vector of censoring indicator
J	The jth set of subjects required for the visualization. The default is J=1 which is the first set of subjects. For visualization, J should be less than the number of subjects divided by 25

Details

The Function fits cox proportional hazard model and does classification based on the majority votes while estimating the Hazard ratio of the low risk group. The function firstly count the number of low risk classification for each subject based on the taxon specific analysis which determines the majority votes. In addition, function visualizes the taxon specific classification for the subjects. 25 subjects is taken for visualization purpose.

Value

A list is returned with the following values

Model.result	The cox proportional regression result based on the majority vote classification
N	The majority vote for each subject
Classif	The majority vote classification for each subjects
Group	The classification of the subjects based on each taxon analysis

Author(s)

Thi Huyen Nguyen, <thihuyen.nguyen@uhasselt.be>

Olajumoke Evangelina Owokotomo, <olajumoke.x.owokotomo@gsk.com>

Ziv Shkedy

See Also

[MSpecificCoxPh](#), [coxph](#), [EstimateHR](#)

Examples

```
# Prepare data
data(Week3_response)
Week3_response = data.frame(Week3_response)
surv_fam_shan_w3 = data.frame(cbind(as.numeric(Week3_response$T1Dweek),
as.numeric(Week3_response$T1D)))
colnames(surv_fam_shan_w3) = c("Survival", "Censor")
prog_fam_shan_w3 = data.frame(factor(Week3_response$Treatment_new))
colnames(prog_fam_shan_w3) = c("Treatment")
data(fam_shan_trim_w3)
names_fam_shan_trim_w3 =
c("Unknown", "Lachnospiraceae", "S24.7", "Lactobacillaceae", "Enterobacteriaceae", "Rikenellaceae")
fam_shan_trim_w3 = data.matrix(fam_shan_trim_w3[,2:82])
rownames(fam_shan_trim_w3) = names_fam_shan_trim_w3
# Running the taxon specific function
Cox_taxon_fam_shan_w3 = MSpecificCoxPh(Survival = surv_fam_shan_w3$Survival,
Micro.mat = fam_shan_trim_w3,
Censor = surv_fam_shan_w3$Censor,
Reduce=FALSE,
Select=5,
Prognostic = prog_fam_shan_w3,
Mean = TRUE,
Method = "BH")

# Using the function
Majority_fam_shan_w3 = Majorityvotes(Result = Cox_taxon_fam_shan_w3,
Prognostic = prog_fam_shan_w3,
Survival = surv_fam_shan_w3$Survival,
Censor = surv_fam_shan_w3$Censor,
J=1)

# The survival analysis for majority vote result
Majority_fam_shan_w3$Model.result

# The majority vote for each subject
Majority_fam_shan_w3$N

# The majority vote classification for each subject
Majority_fam_shan_w3$Classif

# The group for each subject based on the taxon specific analysis
Majority_fam_shan_w3$Group
```

Description

A dataset containing the information of all levels in the ecosystem: OTU, order, family, kingdom, ...

Usage

```
data(metadata_taxonomy)
```

Format

A data frame with 2720 rows and 3 variables:

OTUID,Taxon,Confidence OTU ID and information at higher levels ...

Source

<https://elifesciences.org/articles/37816>

MiFreq	<i>Frequency of Selected Taxa from the LASSO, Elastic-net Cross-Validation</i>
--------	--

Description

The function selects the frequency of selection from the shrinkage method (LASSO, Elastic-net) based on cross validation, that is the number of times each taxon occur during the cross-validation process. This function outputs the mostly selected taxa during the LASSO and Elastic-net cross validation. Selected top taxa are ranked based on frequency of selection and also a particular frequency can be selected. In addition, it visualizes the selected top taxa based on the minimum frequency specified.

Usage

```
MiFreq(Object, TopK = 20, N = 3)
```

Arguments

Object	An object of class <code>cvle</code> returned from the function <code>CVLasoeLascox</code> .
TopK	The number of Top K taxa (5 by default) to be displayed in the frequency of selection graph.
N	The taxa with the specified frequency should be displayed in the frequency of selection graph.

Value

A vector of taxa and their frequency of selection. Also, a graphical representation is displayed.

Author(s)

Thi Huyen Nguyen, <thihuyen.nguyen@uhasselt.be>

Olajumoke Evangelina Owokotomo, <olajumoke.x.owokotomo@gsk.com>

Ziv Shkedy

See Also

[cvmm](#), [coxph](#), [EstimateHR](#), [CVLasoelascox](#)

Examples

```
# Prepare data
data(Week3_response)
Week3_response = data.frame(Week3_response)
surv_fam_shan_w3 = data.frame(cbind(as.numeric(Week3_response$T1Dweek),
as.numeric(Week3_response$T1D)))
colnames(surv_fam_shan_w3) = c("Survival", "Censor")
prog_fam_shan_w3 = data.frame(factor(Week3_response$Treatment_new))
colnames(prog_fam_shan_w3) = c("Treatment")
data(fam_shan_trim_w3)
names_fam_shan_trim_w3 =
c("Unknown", "Lachnospiraceae", "S24.7", "Lactobacillaceae", "Enterobacteriaceae", "Rikenellaceae")
fam_shan_trim_w3 = data.matrix(fam_shan_trim_w3[,2:82])
rownames(fam_shan_trim_w3) = names_fam_shan_trim_w3
# Cross-Validation for LASSO and ELASTIC-NET
CV_lasso_fam_shan_w3 = CVLasoelascox(Survival = surv_fam_shan_w3$Survival,
                                     Censor = surv_fam_shan_w3$Censor,
                                     Micro.mat = fam_shan_trim_w3,
                                     Prognostic = prog_fam_shan_w3,
                                     Standardize = TRUE,
                                     Alpha = 1,
                                     Fold = 4,
                                     Ncv = 10,
                                     nlambda = 100)

# Using the function
MiFreq_fam_shan_w3 = MiFreq(Object = CV_lasso_fam_shan_w3, TopK=5, N=3)
```

Description

Class of object returned by function [MSpecificCoxPh](#).

Usage

```
## S4 method for signature 'ms'
show(object)

## S4 method for signature 'ms'
summary(object)

## S4 method for signature 'ms,ANY'
plot(x, y, ...)
```

Arguments

object	A ms class object
x	A ms class object
y	missing
...	The usual extra arguments to generic functions — see plot , plot.default

Details

plot signature(x = "ms"): Plots for ms class analysis results signature(x = "ms"): Plots for ms class analysis results.

Any parameters of [plot.default](#) may be passed on to this particular plot method.

show(ms-object)

Slots

Result A list of dataframes of each output object of coxph for the taxa.

HRRG A dataframe with estimated taxon-specific HR for low risk group and 95 percent CI.

Group A matrix of the classification group a subject belongs to for each of the taxon analysis. The taxa are on the rows and the subjects are the columns

Mi.names The names of the taxon for the analysis

Author(s)

Thi Huyen Nguyen, <thihuyen.nguyen@uhasselt.be>

Olajumoke Evangelina Owokotomo, <olajumoke.x.owokotomo@gsk.com>

Ziv Shkedy

See Also

[MSpecificCoxPh](#)

MSpecificCoxPh

*Taxon by taxon Cox proportional analysis***Description**

The Function fits cox proportional hazard model and does classification for each taxon separately

Usage

```
MSpecificCoxPh(
  Survival,
  Micro.mat,
  Censor,
  Reduce = FALSE,
  Select = 5,
  Prognostic = NULL,
  Mean = TRUE,
  Quantile = 0.5,
  Method = "BH"
)
```

Arguments

Survival	A vector of survival time with length equals to number of subjects
Micro.mat	A large or small microbiome profile matrix. A matrix with microbiome profiles where the number of rows should be equal to the number of taxa and number of columns should be equal to number of subjects.
Censor	A vector of censoring indicator.
Reduce	A boolean parameter indicating if the microbiome profile matrix should be reduced, default is TRUE and larger microbiome profile matrix is reduced by supervised pca approach.
Select	Number of taxa (default is 5) to be selected from supervised PCA. This is valid only if the argument Reduce=TRUE.
Prognostic	A dataframe containing possible prognostic(s) factor and/or treatment effect to be used in the model.
Mean	The cut off value for the classifier, default is the mean cutoff.
Quantile	If users want to use quantile as cutoff point. They need to specify Mean = FALSE and a quantile that they wish to use. The default is the median cutoff.
Method	Multiplicity adjustment methods.

Details

This function fits taxon by taxon Cox proportional hazard model and perform the classification based on a microbiome risk score which has been estimated using a single taxon. Function is useful for majority vote classification method and taxon by taxon analysis and also for top K taxa.

Value

A object of class `ms` is returned with the following values

Result	The cox proportional regression result for each taxon
HRRG	The hazard ratio statistics (Hazard-ratio, Lower confidence interval and upper confidence interval) of the riskgroup based on the riskscore and the cut off value for each taxon
Group	The classification of the subjects based on each taxon analysis
Mi.names	The names of the taxa for the analysis

Author(s)

Thi Huyen Nguyen, <thihuyen.nguyen@uhasselt.be>

Olajumoke Evangelina Owokotomo, <olajumoke.x.owokotomo@gsk.com>

Ziv Shkedy

See Also

[coxph](#), [EstimateHR](#)

Examples

```
# Prepare data
data(Week3_response)
Week3_response = data.frame(Week3_response)
surv_fam_shan_w3 =
data.frame(cbind(as.numeric(Week3_response$T1Dweek), as.numeric(Week3_response$T1D)))
colnames(surv_fam_shan_w3) = c("Survival", "Censor")
prog_fam_shan_w3 = data.frame(factor(Week3_response$Treatment_new))
colnames(prog_fam_shan_w3) = c("Treatment")
data(fam_shan_trim_w3)
names_fam_shan_trim_w3 =
c("Unknown", "Lachnospiraceae", "S24.7", "Lactobacillaceae", "Enterobacteriaceae", "Rikenellaceae")
fam_shan_trim_w3 = data.matrix(fam_shan_trim_w3[,2:82])
rownames(fam_shan_trim_w3) = names_fam_shan_trim_w3
# Using the function
Cox_taxon_fam_shan_w3 = MSpecificCoxPh(Survival = surv_fam_shan_w3$Survival,
                                       Micro.mat = fam_shan_trim_w3,
                                       Censor = surv_fam_shan_w3$Censor,
                                       Reduce=FALSE,
                                       Select=5,
                                       Prognostic = prog_fam_shan_w3,
                                       Mean = TRUE,
                                       Method = "BH")

# Results
show(Cox_taxon_fam_shan_w3)
summary(Cox_taxon_fam_shan_w3)
```

perm-class

The perm Class.

Description

Class of object returned by function [DistHR](#).

Usage

```
## S4 method for signature 'perm'  
show(object)  
  
## S4 method for signature 'perm'  
summary(object)  
  
## S4 method for signature 'perm,ANY'  
plot(x, y, ...)
```

Arguments

object	A perm class object
x	A perm class object
y	missing
...	The usual extra arguments to generic functions — see plot , plot.default

Slots

HRobs Estimated HR for low risk group on the original data.
HRperm Estimated HR for low risk group on the permuted data.
nperm Number of permutations carried out.
Validation The validation scheme that was used.

Note

The first, third and last vertical line on the plot are the lower, median and upper CI of the permuted data estimated HR while the red line is the estimated HR of the original data

Author(s)

Thi Huyen Nguyen, <thihuyen.nguyen@uhasselt.be>
Olajumoke Evangelina Owokotomo, <olajumoke.x.owokotomo@gsk.com>
Ziv Shkedy

See Also

[DistHR](#), [EstimateHR](#), [SurvPcaClass](#), [SurvPlsClass](#), [Majorityvotes](#), [Lasoelascox](#)

QuantileAnalysis *Quantile sensitivity analysis*

Description

The function performs sensitivity of the cut off quantile for obtaining the risk group obtained under [SurvPlsClass](#), [SurvPcaClass](#) or [Lasoelascox](#) requires for the survival analysis and classification.

Usage

```
QuantileAnalysis(
  Survival,
  Micro.mat,
  Censor,
  Reduce = TRUE,
  Select = 5,
  Prognostic = NULL,
  Plots = FALSE,
  DM = c("PLS", "PCA", "SM"),
  Alpha = 1
)
```

Arguments

Survival	A vector of survival time with length equals to number of subjects.
Micro.mat	A large or small microbiome profile matrix. A matrix with microbiome profiles where the number of rows should be equal to the number of taxa and number of columns should be equal to number of patients.
Censor	A vector of censoring indicator.
Reduce	A boolean parameter indicating if the microbiome profile matrix should be reduced, default is TRUE and larger microbiome profile matrix is reduced by supervised pca approach and first pca is extracted from the reduced matrix to be used in the classifier.
Select	Number of taxa (default is 5) to be selected from supervised PCA. This is valid only if the argument Reduce=TRUE.
Prognostic	A dataframe containing possible prognostic(s) factor and/or treatment effect to be used in the model.
Plots	A boolean parameter indicating if the graphical representation of the analysis should be shown. Default is FALSE and it is only valid for the PCA or PLS dimension method.
DM	The dimension method to be used. PCA implies using the SurvPcaClass , PLS uses SurvPcaClass while SM uses the Lasoelascox which uses the shrinkage method techniques such as lasso and elastic net.
Alpha	The mixing parameter for glmnet (see glmnet). The range is $0 \leq \text{Alpha} \leq 1$. The Default is 1.

SecondFilter	<i>This function is used for the second step of filtering which removes OTUs based on a threshold.</i>
--------------	--

Description

This function is used for the second step of filtering which removes OTUs based on a threshold.

Usage

```
SecondFilter(zero.per.group, Micro.mat, threshold = 0.7, week = 0)
```

Arguments

zero.per.group	a n x 9 matrix. Columns are number of zero in control groups, proportion of zeros in control group, number of subject in control group, number of zero in treated groups, proportion of zeros in treated group, number of subject in treated group, total number of zeros, proportion of zeros in total, number of subject
Micro.mat	OTU matrix (rows are otus, columns are subjects)
threshold	user can choose. For instance, if threshold is 0.7, the function will remove OTUs having at least 70% of zeros in one of two groups
week	A specific time point. To use when having different time points in the dataset.

Value

A smaller microbiome matrix.

Micro.mat.new an smaller OTU matrix with less OTUs

Author(s)

Thi Huyen Nguyen, <thihuyen.nguyen@uhasselt.be>

Olajumoke Evangelina Owokotomo, <olajumoke.x.owokotomo@gsk.com>

Ziv Shkedy

See Also

[SecondFilter](#)

[SecondFilter](#)

Examples

```
# Read dataset
data(Week3_otu)
Week3_otu = data.frame(Week3_otu)
otu_mat_w3 = t(data.matrix(Week3_otu[ , 1:2720]))

# Import dataset from the result of zero_per_group
data(data_zero_per_group_otu_w3)

# Using the function
otu_trim_w3 = SecondFilter(zero.per.group = data_zero_per_group_otu_w3,
                          Micro.mat = otu_mat_w3, threshold = 0.7, week = 3)
```

SITaxa

*Sequential Increase in Taxa for the PCA or PLS classifier***Description**

The Function fits cox proportional hazard model and does classification by sequentially increasing the taxa using either PCA or PLS based on the topK taxa specified.

Usage

```
SITaxa(
  TopK = 15,
  Survival,
  Micro.mat,
  Censor,
  Reduce = TRUE,
  Select = 5,
  Prognostic = NULL,
  Plot = FALSE,
  DM = c("PLS", "PCA"),
  ...
)
```

Arguments

TopK	Top K taxa (5 by default) to be used in the sequential analysis.
Survival	A vector of survival time with length equals to number of subjects.
Micro.mat	A large or small microbiome profile matrix. A matrix with microbiome profiles where the number of rows should be equal to the number of taxa and number of columns should be equal to number of patients.
Censor	A vector of censoring indicator.

Reduce	A boolean parameter indicating if the microbiome profile matrix should be reduced, default is TRUE and larger microbiome profile matrix is reduced by supervised pca approach and first pca is extracted from the reduced matrix to be used in the classifier.
Select	Number of taxa to be selected from supervised PCA. This is valid only if the argument Reduce=TRUE.
Prognostic	A dataframe containing possible prognostic(s) factor and/or treatment effect to be used in the model.
Plot	A boolean parameter indicating if Plot should be shown. Default is FALSE.
DM	Dimension reduction method which can either be PLS or PCA.
...	Additional arguments for plotting and only valid if Plot=TRUE

Details

This function sequentially increase the number of top K taxa to be used in the PCA or PLS methods in order to obtain the risk score. This function internally calls [MSpecificCoxPh](#) to rank the taxa based on HR for each taxon. Therefore taxa can be ordered based on increasing order of the HR for low risk group. Thereafter, the function takes few top K (5 is the default) to be used in the sequential analysis.

Value

A list containing a data frame with estimated HR along with 95% CI at each TopK value for the sequential analysis.

Result	The hazard ratio statistics (HR, Lower confidence interval and upper confidence interval) of the lower riskgroup based for each sequential metabolite analysis
TopKplot	A graphical representation of the Result containing the hazard ratio statistics

Author(s)

Thi Huyen Nguyen, <thihuyen.nguyen@uhasselt.be>

Olajumoke Evangelina Owokotomo, <olajumoke.x.owokotomo@gsk.com>

Ziv Shkedy

See Also

[coxph](#), [EstimateHR](#), [MSpecificCoxPh](#), [SurvPcaClass](#), [SurvPlsClass](#)

Examples

```
# Prepare data
data(Week3_response)
Week3_response = data.frame(Week3_response)
surv_fam_shan_w3 = data.frame(cbind(as.numeric(Week3_response$T1Dweek),
as.numeric(Week3_response$T1D)))
colnames(surv_fam_shan_w3) = c("Survival", "Censor")
prog_fam_shan_w3 = data.frame(factor(Week3_response$Treatment_new))
```

```

colnames(prog_fam_shan_w3) = c("Treatment")
data(fam_shan_trim_w3)
names_fam_shan_trim_w3 =
c("Unknown", "Lachnospiraceae", "S24.7", "Lactobacillaceae", "Enterobacteriaceae", "Rikenellaceae")
fam_shan_trim_w3 = data.matrix(fam_shan_trim_w3[,2:82])
rownames(fam_shan_trim_w3) = names_fam_shan_trim_w3
# Using the function
SITaxa_fam_shan_w3 = SITaxa(TopK=5,
                           Survival = surv_fam_shan_w3$Survival,
                           Micro.mat = fam_shan_trim_w3,
                           Censor = surv_fam_shan_w3$Censor,
                           Reduce=TRUE,
                           Select=5,
                           Prognostic=prog_fam_shan_w3,
                           Plot = TRUE,
                           DM="PLS")

# For the HR statistics
SITaxa_fam_shan_w3$Result

# For the graphical output
SITaxa_fam_shan_w3$TopKplot

```

SummaryData

This function gives indices such as Observed richness, Shannon index, Inverse Simpson, ... of higher level such as levelily, order, phylum, ...

Description

This function gives indices such as Observed richness, Shannon index, Inverse Simpson, ... of higher level such as levelily, order, phylum, ...

Usage

```
SummaryData(Micro.mat, info, measure = "observed")
```

Arguments

Micro.mat	an OTU matrix with OTUs in rows and subjects in columns.
info	A n x 2 matrix containing a column of OTU's names and a column of the corresponding information of the chosen level.
measure	The indices at chosen level that user wishes to use. It can be observed richness, Shannon index, inverse Simpson, ...

Value

A matrix of the selected measurement of the chosen level.

level.measure A matrix of measurements at levelily level of patients

Author(s)

Thi Huyen Nguyen, <thihuyen.nguyen@uhasselt.be>

Olajumoke Evangelina Owokotomo, <olajumoke.x.owokotomo@gsk.com>

Ziv Shkedy

See Also

[SummaryData](#)

Examples

```
# Read dataset
data(Week3_otu)
Week3_otu = data.frame(Week3_otu)
otu_mat_w3 = t(data.matrix(Week3_otu[ , 1:2720]))
data(fam_info_w3)

# USING the function
fam_shan_w3 = SummaryData(Micro.mat = otu_mat_w3, info = fam_info_w3, measure = "shannon")
```

SurvPcaClass

Survival PCA and Classification for microbiome data

Description

The function performs principal component analysis (PCA) on microbiome matrix and fit Cox proportional hazard model with covariates using also the first PCA as covariates.

Usage

```
SurvPcaClass(
  Survival,
  Micro.mat,
  Censor,
  Reduce = TRUE,
  Select = 5,
  Prognostic = NULL,
  Plots = FALSE,
  Mean = TRUE,
  Quantile = 0.5
)
```

Arguments

Survival	A vector of survival time with length equals to number of subjects
Micro.mat	A large or small microbiome profile matrix. A matrix with microbiome profiles where the number of rows should be equal to the number of microbiome and number of columns should be equal to number of patients.
Censor	A vector of censoring indicator
Reduce	A boolean paramier indicating if the microbiome profile matrix should be reduced, default is TRUE and larger microbiome profile matrix is reduced by supervised pca approach and first pca is extracted from the reduced matrix to be used in the classifier.
Select	Number of microbiome (default is 15) to be selected from supervised PCA. This is valid only if the argument Reduce=TRUE
Prognostic	A dataframe containing possible prognostic(s) factor and/or treatment effect to be used in the model.
Plots	A boolean paramier indicating if the plots should be shown. Default is FALSE
Mean	The cut off value for the classifier, default is the mean cutoff
Quantile	If user want to use quantile as cutoff point. They need to specify Mean = FALSE and a quantile that they want to use. The default is the median cutoff

Details

This function can handle single and multiple microbiome. For larger microbiome matrix, this function will reduce largermicrobiome matrix to smaller version using supervised pca approach and this is by default done and can be control by using the argument Reduce. Other prognostic factors can be included to the model.

Value

A object of class SurvPca is returned with the following values

Survfit	The cox proportional regression result using the first PCA
Riskscores	A vector of risk scores which is equal to the number of patents.
Riskgroup	The classification of the subjects based on the PCA into low or high risk group
pc1	The First PCA scores based on either the reduced microbiome matrix or the full matrix
KMplot	The Kaplan-Meier survival plot of the riskgroup
SurvBPlot	The distribution of the survival in the riskgroup
Riskpca	The plot of Risk scores vs first PCA

Author(s)

Thi Huyen Nguyen, <thihuyen.nguyen@uhasselt.be>

Olajumoke Evangelina Owokotomo, <olajumoke.x.owokotomo@gsk.com>

Ziv Shkedy

See Also

[coxph](#), [EstimateHR](#), [princomp](#), [SurvPlsClass](#)

Examples

```
# Prepare data
data(Week3_response)
Week3_response = data.frame(Week3_response)
surv_fam_shan_w3 = data.frame(cbind(as.numeric(Week3_response$T1Dweek),
as.numeric(Week3_response$T1D)))
colnames(surv_fam_shan_w3) = c("Survival", "Censor")
prog_fam_shan_w3 = data.frame(factor(Week3_response$Treatment_new))
colnames(prog_fam_shan_w3) = c("Treatment")
data(fam_shan_trim_w3)
names_fam_shan_trim_w3 =
c("Unknown", "Lachnospiraceae", "S24.7", "Lactobacillaceae", "Enterobacteriaceae", "Rikenellaceae")
fam_shan_trim_w3 = data.matrix(fam_shan_trim_w3[,2:82])
rownames(fam_shan_trim_w3) = names_fam_shan_trim_w3
# Using the function
SPCA_fam_shan_w3 = SurvPcaClass(Survival = surv_fam_shan_w3$Survival,
                               Micro.mat = fam_shan_trim_w3,
                               Censor = surv_fam_shan_w3$Censor,
                               Reduce=TRUE,
                               Select=5,
                               Prognostic = prog_fam_shan_w3,
                               Plots = TRUE,
                               Mean = TRUE)

# Getting the survival regression output
SPCA_fam_shan_w3$SurvFit

# Getting the riskscores
SPCA_fam_shan_w3$RiskScores

# Getting the riskgroup
SPCA_fam_shan_w3$RiskGroup

# Obtaining the first principal component scores
SPCA_fam_shan_w3$pc1
```

SurvPlsClass

Survival PLS and Classification for microbiome data

Description

The function performs partial least squares (PLS) and principal component regression on microbiome matrix and fit Cox proportional hazard model with covariates using the first PLS scores as covariates.

Usage

```
SurvPlsClass(
  Survival,
  Micro.mat,
  Censor,
  Reduce = TRUE,
  Select = 150,
  Prognostic = NULL,
  Plots = FALSE,
  Mean = TRUE,
  Quantile = 0.5
)
```

Arguments

Survival	A vector of survival time with length equals to number of subjects
Micro.mat	A large or small microbiome profile matrix. A matrix with microbiome profiles where the number of rows should be equal to the number of taxa and number of columns should be equal to number of patients.
Censor	A vector of censoring indicator
Reduce	A boolean parameter indicating if the microbiome profile matrix should be reduced, default is TRUE and larger microbiome profile matrix is reduced by supervised pca approach and first pca is extracted from the reduced matrix to be used in the classifier.
Select	Number of taxa (default is 5) to be selected from supervised PCA. This is valid only if the argument Reduce=TRUE
Prognostic	A dataframe containing possible prognostic(s) factor and/or treatment effect to be used in the model.
Plots	A boolean parameter indicating if the plots should be shown. Default is FALSE
Mean	The cut off value for the classifier, default is the mean cutoff
Quantile	If user want to use quantile as cutoff point. They need to specify Mean = FALSE and a quantile that they want to use. The default is the median cutoff

Details

This function reduces larger microbiome matrix to smaller version using supervised pca approach. The function performs the PLS on the reduced microbiome matrix and fit Cox proportional hazard model with first PLS scores as a covariate afterwards. And classifier is then built based on the first PLS scores multiplied by its estimated regression coefficient. Patients are classified using mean of the risk scores as default. However, user can choose any quantile. This function can handle single and multiple taxa. Prognostic factors can also be included to enhance classification.

Value

A object is returned with the following values

Survfit	The cox proportional regression result using the first PCA
---------	--

Riskscores	A vector of risk scores which is equal to the number of patents.
Riskgroup	The classification of the subjects based on the PCA into low or high risk group
pc1	The First PCA scores based on either the reduced Metabolite matrix or the full matrix
KMplot	The Kaplan-Meier survival plot of the riskgroup
SurvBPlot	The distribution of the survival in the riskgroup
Riskpls	The plot of Risk scores vs first PLS

Author(s)

Thi Huyen Nguyen, <thihuyen.nguyen@uhasselt.be>

Olajumoke Evangelina Owokotomo, <olajumoke.x.owokotomo@gsk.com>

Ziv Shkedy

See Also

[coxph](#), [EstimateHR](#), [plsr](#), [SurvPcaClass](#)

Examples

```
# Prepare data
data(Week3_response)
Week3_response = data.frame(Week3_response)
surv_fam_shan_w3 = data.frame(cbind(as.numeric(Week3_response$T1Dweek),
as.numeric(Week3_response$T1D)))
colnames(surv_fam_shan_w3) = c("Survival", "Censor")
prog_fam_shan_w3 = data.frame(factor(Week3_response$Treatment_new))
colnames(prog_fam_shan_w3) = c("Treatment")
data(fam_shan_trim_w3)
names_fam_shan_trim_w3 =
c("Unknown", "Lachnospiraceae", "S24.7", "Lactobacillaceae", "Enterobacteriaceae", "Rikenellaceae")
fam_shan_trim_w3 = data.matrix(fam_shan_trim_w3[,2:82])
rownames(fam_shan_trim_w3) = names_fam_shan_trim_w3
# Using the function
SPLS_fam_shan_w3 = SurvPlsClass(Survival = surv_fam_shan_w3$Survival,
                               Micro.mat = fam_shan_trim_w3,
                               Censor = surv_fam_shan_w3$Censor,
                               Reduce=TRUE,
                               Select=5,
                               Prognostic = prog_fam_shan_w3,
                               Plots = TRUE,
                               Mean = TRUE)

# Getting the survival regression output
SPLS_fam_shan_w3$SurvFit

# Getting the riskscores
SPLS_fam_shan_w3$Riskscores
```

```
# Getting the riskgroup
SPLS_fam_shan_w3$Riskgroup

# Obtaining the first principal component scores
SPLS_fam_shan_w3$pc1
```

Top1Uni	<i>This function finds out the taxon has the smallest p-value, then calculate risk score of patients based on that taxon. Categorized subjects into high or low risk groups based on the mean of the risk score as a cutoff point Checking whether the two groups are significant difference in the probability to be survival.</i>
---------	---

Description

This function finds out the taxon has the smallest p-value, then calculate risk score of patients based on that taxon. Categorized subjects into high or low risk groups based on the mean of the risk score as a cutoff point Checking whether the two groups are significant difference in the probability to be survival.

Usage

```
Top1Uni(Result, Micro.mat, Survival, Censor, Plots = FALSE)
```

Arguments

Result	A Result statistic of all taxon.
Micro.mat	A large or small microbiome matrix. A matrix with microbiome profiles where the number of rows should be equal to the number of taxa and number of columns should be equal to number of patients.
Survival	Survival A vector of survival time with length equals to number of subjects
Censor	A vector of censoring indicator
Plots	A boolean parameter indicating if plots should be shown. Default is FALSE. If TRUE, the first plot is plot of the observed Kaplan-Meier curves per group while the second is boxplot of the two groups.

Value

A list is returned with the following values

name.top1	Taxon having the smallest p-value in the univariate coxPH model
sum.top1	Result statistic of the taxon containing coefficient, exponential of coefficient, raw p.value using LRT, and p.value after using BH adjustment
KMplot.top1	Kaplan-Meier plot
log.rank.top1	Log-rank test

Author(s)

Thi Huyen Nguyen, <thihuyen.nguyen@uhasselt.be>

Olajumoke Evangelina Owokotomo, <olajumoke.x.owokotomo@gsk.com>

Ziv Shkedy [Top1Uni](#)

Examples

```
# Prepare data
data(Week3_response)
Week3_response = data.frame(Week3_response)
surv_fam_shan_w3 = data.frame(cbind(as.numeric(Week3_response$T1Dweek),
as.numeric(Week3_response$T1D)))
colnames(surv_fam_shan_w3) = c("Survival", "Censor")
prog_fam_shan_w3 = data.frame(factor(Week3_response$Treatment_new))
colnames(prog_fam_shan_w3) = c("Treatment")
data(fam_shan_trim_w3)
names_fam_shan_trim_w3 =
c("Unknown", "Lachnospiraceae", "S24.7", "Lactobacillaceae", "Enterobacteriaceae", "Rikenellaceae")
fam_shan_trim_w3 = data.matrix(fam_shan_trim_w3[,2:82])
rownames(fam_shan_trim_w3) = names_fam_shan_trim_w3
# Obtain summary statistics for families
summary_fam_shan_w3 = CoxPHUni(Survival = surv_fam_shan_w3$Survival,
                               Censor = surv_fam_shan_w3$Censor,
                               Prognostic = prog_fam_shan_w3,
                               Micro.mat = fam_shan_trim_w3,
                               Method = "BH")

# Analysis of the taxon having smallest p-value (in the result of using CoxPHUni function)
top1_fam_shan_w3 = Top1Uni(Result = summary_fam_shan_w3,
                           Micro.mat = fam_shan_trim_w3,
                           Survival = surv_fam_shan_w3$Survival,
                           Censor = surv_fam_shan_w3$Censor,
                           Plots = TRUE)
```

Week3_otu

OTU table at week 3.

Description

A dataset containing the count of OTUs.

Usage

```
data(Week3_otu)
```

Format

A data frame with 81 rows and 2724 variables, we only use 2720 first variables:

X226097bd7a1661a286a3b62d1c1f0e3a-X005d3193f381b0793f0c928bde66dd21 Names of the OTUs

SampleID ID of the subject

Treatment Treatment variable

T1DWeek Time to develop T1D in week

T1D Censored indicator

Source

<https://elifesciences.org/articles/37816>

Week3_response	<i>Response dataset.</i>
----------------	--------------------------

Description

A dataset containing the information of subjects.

Usage

```
data(Week3_response)
```

Format

A data frame with 81 rows and 30 variables:

SampleID ID of the subject

Treatment Treatment variable

T1Dweek Time to develop T1D in week

T1D Censored indicator

Treatment_new Treatment indicator obtained from treatment variable

Source

<https://elifesciences.org/articles/37816>

ZerosPerGroup	<i>This function returns a matrix with rows are Micros and 9 columns containing number and the proportion of zeros per groups of treatments and in total.</i>
---------------	---

Description

This function returns a matrix with rows are Micros and 9 columns containing number and the proportion of zeros per groups of treatments and in total.

Usage

```
ZerosPerGroup(
  Micro.mat,
  groups,
  week = 0,
  n.obs = n.obs,
  n.control = n.control,
  n.treated = n.treated,
  n.mi = n.mi,
  plot = FALSE
)
```

Arguments

Micro.mat	Micro matrix (rows are Micros, columns are subjects)
groups	Treatment groups or groups of any binary variables
week	A specific time point. To use when having different time points in the dataset.
n.obs	Number of patients.
n.control	Number of patients in control group or in the first group.
n.treated	Number of patients in treated group or in the second group.
n.mi	Number of taxa.
plot	A boolean parameter indicating if the plot should be shown. Default is FALSE.

Value

	A matrix with information of number and the proportion of zeros per groups.
zero.per.group	A matrix with rows are Micros and 9 columns containing number and the proportion of zeros per groups of treatments and in total.
plot	Plot percentage of zeros per group

Author(s)

Thi Huyen Nguyen, <thihuyen.nguyen@uhasselt.be>
 Olajumoke Evangelina Owokotomo, <olajumoke.x.owokotomo@gsk.com>
 Ziv Shkedy

See Also[ZerosPerGroup](#)**Examples**

```
# Preparing data for analysis at OTU level
data(Week3_otu)
data(Week3_response)
Week3_otu = data.frame(Week3_otu)
otu_mat_w3 = t(data.matrix(Week3_otu[ , 1:2720]))
n_obs = dim(otu_mat_w3)[2]
n_control = table(Week3_response$Treatment_new)[1]
n_treated = table(Week3_response$Treatment_new)[2]
n_otu = dim(otu_mat_w3)[1]
# Calculate zeros per groups
zero_per_group_otu_w3 = ZerosPerGroup(Micro.mat = otu_mat_w3,
                                       groups = Week3_response$Treatment_new,
                                       week = 3,
                                       n.obs = n_obs,
                                       n.control = n_control,
                                       n.treated = n_treated,
                                       n.mi = n_otu,
                                       plot = TRUE)
```

Index

* datasets

- data_zero_per_group_otu_w3, 21
 - fam_info_w3, 26
 - fam_shan_trim_w3, 27
 - metadata_taxonomy, 34
 - Week3_otu, 53
 - Week3_response, 54
- coxph, 6, 13, 24, 26, 32, 34, 36, 39, 42, 45, 49, 51
- CoxPHUni, 3, 3
- CVLasoelascox, 4, 7, 35, 36
- cvle, 5, 35
- cvle (cvle-class), 7
- cvle-class, 7
- cvle-method (cvle-class), 7
- CVMajorityvotes, 8, 14
- cvmm, 13, 20, 36
- cvmm (cvmm-class), 10
- cvmm-class, 10
- cvmm-method (cvmm-class), 10
- CVMSpecificCoxPh, 10, 11, 12
- cvmv, 9
- cvmv (cvmv-class), 14
- cvmv-class, 14
- cvmv-method (cvmv-class), 14
- CVPcaPls, 15, 15, 17–19
- cvpp, 16
- cvpp (cvpp-class), 17
- cvpp-class, 17
- cvpp-method (cvpp-class), 17
- cvsit, 18, 20
- cvsit (cvsit-class), 18
- cvsit-class, 18
- cvsit-method (cvsit-class), 18
- CVSITaxa, 19
- data_zero_per_group_otu_w3, 21
- DistHR, 22, 40
- EstimateHR, 6, 8, 13, 24, 24, 32, 34, 36, 39, 40, 42, 45, 49, 51
- fam_info_w3, 26
- fam_shan_trim_w3, 27
- FirstFilter, 28, 28
- GetRA, 29, 29
- glmnet, 5, 6, 8, 31, 32, 41
- hello, 30
- Lasoelascox, 6, 8, 22, 24–26, 30, 40–42
- Majorityvotes, 8, 9, 15, 22, 24, 33, 40
- metadata_taxonomy, 34
- MiFreq, 35
- ms, 39
- ms (ms-class), 36
- ms, ANY (ms-class), 36
- ms-class, 36
- ms-method (ms-class), 36
- MSSpecificCoxPh, 13, 21, 33, 34, 36, 37, 38, 45
- perm, 23
- perm (perm-class), 40
- perm-class, 40
- perm-method (perm-class), 40
- plot, 7, 11, 14, 18, 19, 37, 40
- plot, cvle, missing-method (cvle-class), 7
- plot, cvmm, ANY-method (cvmm-class), 10
- plot, cvmv, ANY-method (cvmv-class), 14
- plot, cvpp, missing-method (cvpp-class), 17
- plot, cvsit, missing-method (cvsit-class), 18
- plot, ms, ANY-method (ms-class), 36
- plot, perm, ANY-method (perm-class), 40
- plot.default, 7, 11, 14, 18, 19, 37, 40
- plsr, 51
- princomp, 49

QuantileAnalysis, [41](#)

SecondFilter, [43](#), [43](#)

show, cvle-method (cvle-class), [7](#)

show, cvmm-method (cvmm-class), [10](#)

show, cvmv-method (cvmv-class), [14](#)

show, cvpp-method (cvpp-class), [17](#)

show, cvsit-method (cvsit-class), [18](#)

show, ms-method (ms-class), [36](#)

show, perm-method (perm-class), [40](#)

SITaxa, [20](#), [21](#), [44](#)

summary, cvle-method (cvle-class), [7](#)

summary, cvmm-method (cvmm-class), [10](#)

summary, cvmv-method (cvmv-class), [14](#)

summary, cvpp-method (cvpp-class), [17](#)

summary, cvsit-method (cvsit-class), [18](#)

summary, ms-method (ms-class), [36](#)

summary, perm-method (perm-class), [40](#)

SummaryData, [46](#), [47](#)

SurvPcaClass, [15](#), [16](#), [18](#), [19](#), [22](#), [24](#), [40–42](#),
[45](#), [47](#), [51](#)

SurvPlsClass, [15](#), [16](#), [18](#), [19](#), [22](#), [24](#), [40–42](#),
[45](#), [49](#), [49](#)

Top1Uni, [52](#), [53](#)

Week3_otu, [53](#)

Week3_response, [54](#)

ZerosPerGroup, [55](#), [56](#)