Package 'EnMCB'

January 22, 2021

Type Package

Title Predicting Disease Progression Based on Methylation Correlated Blocks using Ensemble Models

Version 1.2.2

Date 2019-10-04

Author Xin Yu

Maintainer Xin Yu <whirlsyu@gmail.com>

Depends R (>= 4.0)

Encoding UTF-8

Imports foreach, doParallel, parallel, stats, survivalROC, glmnet, rms, survivalsvm, ggplot2, minfi, IlluminaHumanMethylation450kanno.ilmn12.hg19, survival, utils

VignetteBuilder knitr

Suggests SummarizedExperiment, testthat, Biobase, survminer, affycoretools, knitr, plotROC, prognosticROC

Description Creation of the correlated blocks using DNA methylation profiles. A stacked ensemble of machine learning models, which combined the support vector machine and elastic-net regression model, can be constructed to predict disease progression.

License GPL-2

BugReports https://github.com/whirlsyu/EnMCB/issues

biocViews Normalization, DNAMethylation, MethylationArray, SupportVectorMachine

LazyData FALSE

RoxygenNote 7.1.1

git_url https://git.bioconductor.org/packages/EnMCB

git_branch RELEASE_3_12

git_last_commit 944981f

git_last_commit_date 2021-01-07

Date/Publication 2021-01-21

R topics documented:

`create_demo` *create demo matrix*

Description

Demo matrix for methylation matrix.

Usage

```
create_demo(model = c("all", "short")[1])
```

Arguments

`model`	Two options, 'all' or 'short' for creating full dataset or very brief demo.

Value

This function will generate a demo data.

Author(s)

Xin Yu

Examples

```
demo_set<-create_demo()
```

demo_data *Expression matrix of demo dataset.*

Description

A Expression matrix containing the 10020 CpGs beta value of 455 samples in TCGA lung Adenocarcinoma dataset. This will call from create_demo() function.

Usage

```
data(demo_data)
```

Format

ExpressionSet:

rownames rownames of 10020 CpG features

colnames colnames of 455 samples

realdata Real data matrix for demo.

demo_MCBinformation *MCB information.*

Description

A dataset containing the number and other attributes of 94 MCBs; This results was created by the identification function IdentifyMCB. This data used for metricMCB function.

Usage

```
data(demo_MCBinformation)
```

Format

A data frame with 94 rows and 8 variables:

MCB_no MCB code

start Start point of this MCB in the chromosome.

end End point of this MCB in the chromosome.

CpGs All the CpGs probe names in the MCB.

location Start, end point and the chromosome number of this MCB.

chromosomes the chromosome number of this MCB.

length the length of bps of this MCB in the chromosome.

CpGs_num number of CpG probes of this MCB.

demo_survival_data *Survival data of demo dataset.*

Description

A Surv containing survival value of 455 samples in TCGA lung Adenocarcinoma dataset.

Usage

```
data(demo_survival_data)
```

Format

Surv data created by Surv() function in survival package. This data have two unnamed arguments, they will match time and event.

draw_survival_curve *draw survival curve*

Description

Draw a survival curve based on survminer package. This is a wrapper function of ggsurvplot.

Usage

```
draw_survival_curve(
  exp,
  living_days,
  living_events,
  write_name,
  title_name = "",
  threshold = NA
)
```

Arguments

exp	expression level for gene.
living_days	The survival time (days) for each individual.
living_events	The survival event for each individual, 0 indicates alive and 1 indicates death. Other choices are TRUE/FALSE (TRUE = death) or 1/2 (2=death). For interval censored data, the status indicator is 0=right censored, 1=event at time, 2=left censored, 3=interval censored.
write_name	The name for pdf file which contains the result figure.
title_name	The title for the result figure.
threshold	Threshold used to indicate the high risk or low risk.

Value

This function will generate a pdf file with 300dpi which compare survival curves using the Kaplan-Meier (KM) test.

Author(s)

Xin Yu

Examples

```
data(demo_survival_data)
library(survival)
demo_set<-create_demo()
draw_survival_curve(demo_set[1,],
    living_days = demo_survival_data[,1],
    living_events =demo_survival_data[,2],
    write_name = "demo_data" )
```

---

ensemble_model *Trainging stacking ensemble model for Methylation Correlation Block*

Description

Method for training a stacking ensemble model for Methylation Correlation Block.

Usage

```
ensemble_model(single_res,training_set,Surv_training,testing_set,Surv_testing)
```

Arguments

single_res	Methylation Correlation Block information returned by the IndentifyMCB function.
training_set	methylation matrix used for training the model in the analysis.
Surv_training	Survival function contain the survival information for training.
testing_set	methylation matrix used for testing the model in the analysis.
Surv_testing	Survival function contain the survival information for testing.

Value

Object of class `list` with elements (XXX repesents the model you choose):

cox	Model object for the cox model at first level.
svm	Model object for the svm model at first level.
enet	Model object for the enet model at first level.
stacking	Model object for the stacking model.

Author(s)

Xin Yu

References

Xin Yu et al. 2019 Predicting disease progression in lung adenocarcinoma patients based on methylation correlated blocks using ensemble machine learning classifiers (under review)

Examples

```
#import datasets
library(survival)
data(demo_survival_data)
datamatrix<-create_demo()
data(demo_MCBinformation)
#select MCB with at least 3 CpGs.
demo_MCBinformation<-demo_MCBinformation[demo_MCBinformation[,"CpGs_num"]>2,]
trainingset<-colnames(datamatrix) %in% sample(colnames(datamatrix),0.6*length(colnames(datamatrix)))
select_single_one=1
em<-ensemble_model(t(demo_MCBinformation[select_single_one,]),
    training_set=datamatrix[,trainingset],
    Surv_training=demo_survival_data[trainingset])
```

---

ensemble_prediction *fitting function using stacking ensemble model for Methylation Correlation Block*

Description

predict is a generic function for predictions from the results of stacking ensemble model fitting functions. The function invokes particular methods which is the ensemble model described in the reference.

Usage

```
ensemble_prediction(ensemble_model, predition_data, multiple_results = FALSE)
```

Arguments

`ensemble_model`	ensemble model which built by ensemble_model() function
`predition_data`	A vector, matrix, list, or data frame containing the predictions (input).
`multiple_results`	Boolean vector, True for including the single model results.

Value

Object of numeric class `double`

References

Xin Yu et al. 2019 Predicting disease progression in lung adenocarcinoma patients based on methylation correlated blocks using ensemble machine learning classifiers (under review)

Examples

```
library(survival)
#import datasets
data(demo_survival_data)
datamatrix<-create_demo()
data(demo_MCBinformation)
#select MCB with at least 3 CpGs.
demo_MCBinformation<-demo_MCBinformation[demo_MCBinformation[,"CpGs_num"]>2,]
trainingset<-colnames(datamatrix) %in% sample(colnames(datamatrix),0.6*length(colnames(datamatrix)))
testingset<-!trainingset
#select one MCB
select_single_one=1
em<-ensemble_model(t(demo_MCBinformation[select_single_one,]),
    training_set=datamatrix[,trainingset],
    Surv_training=demo_survival_data[trainingset])

em_prediction_results<-ensemble_prediction(ensemble_model = em,
predition_data = datamatrix[,testingset])
```

---

fast_roc_calculation *Fast calculation of AUC for ROC using parallel strategy*

Description

This function is used to create time-dependent ROC curve from censored survival data using the Kaplan-Meier (KM) or Nearest Neighbor Estimation (NNE) method of Heagerty, Lumley and Pepe, 2000

Usage

```
fast_roc_calculation(test_matrix, y_surv, predict_time = 5, roc_method = "NNE")
```

Arguments

test_matrix	Test matrix used in the analysis. Colmuns are samples, rows are markers.
y_surv	Survival information created by Surv function in survival package.
predict_time	Time point of the ROC curve, default is 5 year.
roc_method	Method for fitting joint distribution of (marker,t), either of KM or NNE, the default method is NNE.

Value

This will retrun a numeric vector contains AUC results for each row in test_matrix.

Author(s)

Xin Yu

Examples

```
data(demo_survival_data)
data('demo_data',package = "EnMCB")
demo_set<-demo_data$realdata
res<-fast_roc_calculation(demo_set[1:2,],demo_survival_data)
```

`IdentifyMCB` *Identification of methylation correlated blocks*

Description

This function is used to partition the genome into blocks of tightly co-methylated CpG sites, Methylation correlated blocks. This function calculates Pearson correlation coefficients r^2 between the beta values of any two CpGs r^2 < CorrelationThreshold was used to identify boundaries between any two adjacent markers indicating uncorrelated methylation. Markers not separated by a boundary were combined into MCB. Pearson correlation coefficients between two adjacent CpGs were calculated.

Usage

```
IdentifyMCB(
  MethylationProfile,
  method = c("pearson", "spearman", "kendall")[1],
  CorrelationThreshold = 0.8,
  PositionGap = 1000,
  platform = "Illumina Methylation 450K"
)
```

Arguments

`MethylationProfile`	Methylation matrix is used in the analysis.
`method`	method used for calculation of correlation, should be one of "pearson","spearman","kendall". Defualt is "pearson".
`CorrelationThreshold`	coef correlation threshold is used for define boundaries.
`PositionGap`	CpG Gap between any two CpGs positioned CpG sites less than 1000 bp (default) will be calculated.
`platform`	This parameter indicates the platform used to produce the methlyation profile.

Details

Currently, only illumina 450k platform is supported, the methylation profile need to convert into matrix format.

Value

Object of class `list` with elements:

`MCBsites`	Character set contains all CpG sites in MCBs.
`MCBinformation`	Matrix contains the information of results.

Author(s)

Xin Yu

References

Xin Yu et al. 2019 Predicting disease progression in lung adenocarcinoma patients based on methylation correlated blocks using ensemble machine learning classifiers (under review)

Examples

```
data('demo_data',package = "EnMCB")

#import the demo TCGA data with 10000+ CpGs site and 455 samples
#remove # to run
res<-IdentifyMCB(demo_data$realdata)
demo_MCBinformation<-res$MCBinformation
```

`metricMCB` *Calculation of the metric matrix for Methylation Correlation Block*

Description

To enable quantitative analysis of the methylation patterns within individual Methylation Correlation Blocks across many samples, a single metric to define the methylated pattern of multiple CpG sites within each block. Compound scores which calculated all CpGs within individual Methylation Correlation Blocks by linear, SVM or elastic-net model Predict values were used as the compound methylation values of Methylation Correlation Blocks.

Usage

```
metricMCB(MCBset,training_set,Surv,testing_set,Surv.new,Method,silent)
```

Arguments

`MCBset`	Methylation Correlation Block information returned by the IndentifyMCB function.
`training_set`	methylation matrix used for training the model in the analysis.
`Surv`	Survival function contain the survival information for training.
`testing_set`	methylation matrix used in the analysis. This can be missing then training set itself will be used as testing set.
`Surv.new`	Survival function contain the survival information for testing.
`Method`	model used to calculate the compound values for multiple Methylation correlation blocks. Options include "svm" "cox" and "enet". The default option is SVM method.
`silent`	Ture indicates that processing information and progress bar will be shown.

Value

Object of class `list` with elements (XXX will be replaced with the model name you choose):

MCB_XXX_matrix_training	Prediction results of model for training set.
MCB_XXX_matrix_test_set	Prediction results of model for test set.
XXX_auc_results	AUC results for each model.
best_XXX_model	Model object for the model with best AUC.
maximum_auc	Maximum AUC for the whole generated models.

Author(s)

Xin Yu

References

Xin Yu et al. 2019 Predicting disease progression in lung adenocarcinoma patients based on methylation correlated blocks using ensemble machine learning classifiers (under review)

Examples

```
#import datasets
data(demo_survival_data)
datamatrix<-create_demo()
data(demo_MCBinformation)
#select MCB with at least 3 CpGs.
demo_MCBinformation<-demo_MCBinformation[demo_MCBinformation[,"CpGs_num"]>2,]

trainingset<-colnames(datamatrix) %in% sample(colnames(datamatrix),0.6*length(colnames(datamatrix)))
testingset<-!trainingset
#create the results using Cox regression.
mcb_cox_res<-metricMCB(MCBset = demo_MCBinformation,
               training_set = datamatrix[,trainingset],
               Surv = demo_survival_data[trainingset],
               testing_set = datamatrix[,testingset],
               Surv.new = demo_survival_data[testingset],
               Method = "cox"
               )
```

`metricMCB.cv` *Calculation of model AUC for Methylation Correlation Blocks using cross validation*

Description

To enable quantitative analysis of the methylation patterns within individual Methylation Correlation Blocks across many samples, a single metric to define the methylated pattern of multiple CpG sites within each block. Compound scores which calculated all CpGs within individual Methylation Correlation Blocks by SVM model were used as the compound methylation values of Methylation Correlation Blocks.

Usage

```
metricMCB.cv(MCBset,data_set,Surv,nfold,Method,seed,silent)
```

Arguments

MCBset	Methylation Correlation Block information returned by the IndentifyMCB function.
data_set	methylation matrix used for training the model in the analysis.
Surv	Survival function contain the survival information for training.
nfold	fold used in the cross validation precedure.
Method	model used to calculate the compound values for multiple Methylation correlation blocks. Options include "svm" "cox" and "lasso". The default option is SVM method.
seed	seed int for cross validation sampling.
silent	Ture indicates that processing information and progress bar will be shown.

Value

Object of class `list` with elements (XXX will be replaced with the model name you choose):

MCB_matrix	Prediction results of model.
auc_results	AUC results for each model.

Author(s)

Xin Yu

References

Xin Yu et al. 2019 Predicting disease progression in lung adenocarcinoma patients based on methylation correlated blocks using ensemble machine learning classifiers (under review)

Examples

```
#import datasets
data(demo_survival_data)
datamatrix<-create_demo()
data(demo_MCBinformation)
#select MCB with at least 3 CpGs.
demo_MCBinformation<-demo_MCBinformation[demo_MCBinformation[,"CpGs_num"]>2,]

trainingset<-colnames(datamatrix) %in% sample(colnames(datamatrix),0.6*length(colnames(datamatrix)))
testingset<-!trainingset
#create the results using Cox regression.
mcb_cox_res<-metricMCB.cv(MCBset = demo_MCBinformation,
               data_set = datamatrix,
               Surv = demo_survival_data,
               Method = "cox")
```

pre_process_methylation *Preprocess the Beta value matrix*

Description

This process is optional for the pipeline. This function pre-process the Beta matrix and transform the Beta value into M value.

Usage

```
pre_process_methylation(met,Mvalue,constant_offset,remove_na,remove_percentage)
```

Arguments

met	methylation matrix for CpGs. Rows are the CpG names, columns are samples.
Mvalue	Boolean value, TRUE for the M transformation.
constant_offset	the constant offset used in the M transformation formula.
remove_na	Boolean value, if TRUE ,CpGs with NA values will be removed.
remove_percentage	If precentage of NA value exceed the threshold(percentage), the whole CpG probe will be removed. Otherwise, the NA values are replaced with rowmeans.

Value

Object of class `matrix`.

Examples

```
demo_set<-create_demo()
pre_process_methylation(demo_set,Mvalue=FALSE)
```

univ_coxph *Batch test for variables using coxph*

Description

Batch test for variables using coxph

Usage

```
univ_coxph(dataframe, y_surv, digits = 4, asnumeric = TRUE)
```

Arguments

dataframe	Clinic data and covariates ready to be tested. Rows are variables and columns are samples.
y_surv	Survival function contain survival data, usually are obtained form Surv() function in survival package.
digits	Integer indicating the number of decimal places.
asnumeric	indicator that the data will be (True) / not (False) transformed into numeric. Default is true.

Value

Object of class matrix with results.

Author(s)

Xin Yu

Examples

```
data(demo_survival_data)
data('demo_data',package = "EnMCB")
demo_set<-demo_data$realdata
res<-univ_coxph(demo_set,demo_survival_data)
```

Index