Package 'VSOLassoBag'

November 25, 2022

Type Package

Title Variable Selection Oriented LASSO Bagging Algorithm

Version 0.99.0

Date 2022-11-10

biocViews Software, StatisticalMethod, FeatureExtraction

Description A wrapped LASSO approach by integrating an ensemble learning strategy to help select efficient, stable, and high confidential variables from omics-based data. Using a bagging strategy in combination of a parametric method or inflection point search method for cut-off threshold determination. This package can integrate and vote variables generated from multiple LASSO models to determine the optimal candidates. Luo H, Zhao Q, et al (2020) <doi:10.1126/scitranslmed.aax7533> for more details.

License GPL-3

Encoding UTF-8

Depends R (>= 3.6.0)

Imports glmnet, survival, ggplot2, POT, parallel, utils, pbapply, methods, SummarizedExperiment

RoxygenNote 7.2.0

Suggests rmarkdown, knitr, rmdformats, qpdf

VignetteBuilder knitr

LazyData true

NeedsCompilation no

Author Jiaqi Liang [aut], Chaoye Wang [aut, cre]

Maintainer Chaoye Wang <wangcy1@sysucc.org.cn>

Repository CRAN

Date/Publication 2022-11-25 12:50:02 UTC

R topics documented:

`ExpressionData` *Simulated Example Data for VSOLassoBag Application*

Description

Simulated Example Data for VSOLassoBag Application

Usage

```
ExpressionData
```

Format

An object constructed by SummarizedExperiment. it contains the Simulated Example Data for VSOLassoBag with two parts.

assay(ExpressionData) The independent variables matrix (X) contains 500 variates (rows) x 200 samples (columns).

colData(ExpressionData) The dependent variable(s) matrix (Y) contains same rows as samples and 1 variate (column) for gaussian, binomial, possion model application, or 2 variates (columns) for mgaussian, multinomial and cox model application. The first 1~10 independent variables (X_1~X_10) are simulated to be related to the dependent variable (D_1), and the first 6~15 independent variables (X_6~X_15) are simulated to be related to the second dependent variable (D_2) for mgaussian and multinomial model application. Survival data for cox model application were simulated with right-censored rate = 0.5 using `sim.survdata` function derived from the coxed R package.

Examples

```
data("ExpressionData")
```

kneedle *Kneedle Algorithm: to detect elbow point(s) on the curve*

Description

An internal function utilized by VSOLassoBag.

Usage

```
kneedle(res, S = 10, auto.loose = TRUE, min.S = 0.1, loosing.factor = 0.5)
```

Arguments

res	a dataframe with variables and observed frequency
S	numeric, determines how aggressive the elbow points on the curve to be called, smaller means more aggressive and larger means more conservative
auto.loose	if TRUE, will reduce ‘kneedle.S‘ in case no elbow point is found with the set ‘kneedle.S‘
min.S	a numeric value determines the minimal value that ‘kneedle.S‘ will be loosed to.
loosing.factor	a numeric value range in (0,1), which ‘kneedle.S‘ is multiplied by to reduce itself.

Value

the original input dataframe along with the elbow point indicator "elbow.point" with elbow point(s) marked with "*", "Diff" the difference curve, "Thres" the threshold.

References

Original Kneedle Algorithm, the algorithm utilized in LassoBag has been modified.

Examples

```
load(system.file("extdata/Results.RData", package="VSOLassoBag"))
kneedle(Results)
```

`LessPermutation` *Reduce permutation times*

Description

Reduce permutation times by fitting generalized pareto distribution of the right tail data.

Usage

```
LessPermutation(
  X,
  x0,
  fitting.method = "mle",
  search.step = 0.01,
  fit.cutoff = 0.05,
  when.to.fit = 0.05
)
```

Arguments

`X`	a vector of data recording the permutation values
`x0`	observed value
`fitting.method`	method to fit GPD, default is "mle", alternative "gd"(gradient descend)
`search.step`	the length of step (this param * length(X)) to find threshold. Default 0.01
`fit.cutoff`	the cutoff of p value to judge whether it fits well to GPD, default is 0.05
`when.to.fit`	a cutoff to tell how many sample values are bigger than the target value then we don't need to fit GPD. it is a portion.Default 0.05

Value

p value of the observed value in the permutation test

Examples

```
x = POT::rgpd(200, 1, 2, 0.25)
LessPermutation(x,1,fitting.method='gd')
```

simpleEstimation *Parametric Statistical Test*

Description

An internal function utilized by VSOLassoBag.

Usage

```
simpleEstimation(res.df, bootN)
```

Arguments

res.df	a dataframe with variables and observed frequency
bootN	an integer, bagging times

Value

a list of p-value of each variable and the average selection ratio

References

RRLASSO, Park H., et al, 2015, the algorithm utilized in LassoBag has been modified.

Examples

```
load(system.file("extdata/Results.RData", package="VSOLassoBag"))
simpleEstimation(Results, 10)
```

VSOLassoBag *One-step main function of VSOLassoBag framework*

Description

An one-step function that can be easily utilized for selecting important variables from multiple models inherited from R package *glmnet*. Several methods (Parametric Statistical Test, Curve Elbow Point Detection and Permutation Test) are provided for the cut-off point decision of the importance measure (i.e. observed selection frequency) of variables.

Usage

```
VSOLassoBag(
  ExpressionData,
  outcomevariable,
  observed.fre = NULL,
  bootN = 1000,
  boot.rep = TRUE,
 a.family = c("gaussian", "binomial", "poisson", "multinomial", "cox", "mgaussian"),
  additional.covariable = NULL,
  bagFreq.sigMethod = "CEP",
  kneedle.S = 10,
  auto.loose = TRUE,
  loosing.factor = 0.5,
  min.S = 0.1,
  use.gpd = FALSE,
  fit.pareto = "gd",
  imputeN = 1000,
  imputeN.max = 2000,
  permut.increase = 100,
  parallel = FALSE,
  n.cores = 1,
  nfolds = 4,
  lambda.type = "lambda.1se",
  plot.freq = "part",
  plot.out = FALSE,
  do.plot = TRUE,
  output.dir = NA,
  filter.method = "auto",
  inbag.filter = TRUE,
  filter.thres.method = "fdr",
  filter.thres.P = 0.05,
  filter.rank.cutoff = 0.05,
  filter.min.variables = -Inf,
  filter.max.variables = Inf,
  filter.result.report = TRUE,
  filter.report.all.variables = TRUE,
  post.regression = FALSE,
  post.LASSO = FALSE,
  pvalue.cutoff = 0.05,
  used.elbow.point = "middle"
)
```

Arguments

`ExpressionData`	ExpressionData is an object constructed by SummarizedExperiment. It contains the independent variables matrix and outcome variables matrix.
`outcomevariable`	Variables which must be the column name of the outcome variables matrix were

used to construct models.

`observed.fre` dataframe with columns 'variable' and 'Frequency', which can be obtained from existed VSOLassoBag results for re-analysis. A warning will be issued if the variables in ‘observed.fre‘ not found in ‘mat‘, and these variables will be excluded.

`bootN` the size of re-sampled samples for bagging, default 1000; smaller consumes less processing time but may not get robust results.

`boot.rep` whether sampling with return or not in the bagging procedure

`a.family` a character determine the data type of out.mat, the same used in `glmnet`.

`additional.covariable`
provide additional covariable(s) to build the cox model, only valid in Cox method (‘a.family‘ == "cox"); a data.frame with same rows as ‘mat‘

`bagFreq.sigMethod`
a character to determine the cut-off point decision method for the importance measure (i.e. the observed selection frequency). Supported methods are "Parametric Statistical Test" (abbr. "PST"), "Curve Elbow Point Detection" ("CEP") and "Permutation Test" ("PERT"). The default and preferable method is "CEP". The method "PERT" is not recommended due to consuming time and memmory requirement.

`kneedle.S` numeric, an important parameter that determines how aggressive the elbow points on the curve to be called, smaller means more aggressive and may find more elbow points. Default ‘kneedle.S‘=10 seems fine, but feel free to try other values. The selection of ‘kneedle.S‘ should be based on the shape of observed frequency curve. It is suggested to use larger S first.

`auto.loose` if TRUE, will reduce ‘kneedle.S‘ in case no elbow point is found with the set ‘kneedle.S‘; only valid when ‘bagFreq.sigMethod‘ is "Curve Elbow Point Detection" ("CEP").

`loosing.factor` a numeric value range in (0,1), which ‘kneedle.S‘ is multiplied by to reduce itself; only valid when ‘auto.loose‘ set to TRUE.

`min.S` a numeric value determines the minimal value that ‘kneedle.S‘ will be loosed to; only valid when ‘auto.loose‘ set to TRUE.

`use.gpd` whether to fit Generalized Pareto Distribution to the permutation results to accelerate the process. Only valid when ‘bagFreq.sigMethod‘ is "Permutation Test" ("PERT").

`fit.pareto` the method of fitting Generalized Pareto Distribution, default choice is "gd", for gradient descend, and alternative as "mle", for Maximum Likelihood Estimation (only valid in "PERT" mode).

`imputeN` the initial permutation times (only valid in "PERT" mode).

`imputeN.max` the max permutation times. Regardless of whether p-value has meet the requirement (only valid in "PERT" mode).

`permut.increase`
if the initial imputeN times of permutation doesn't meet the requirement, then we add ‘permut.increase times of permutation to get more random/permutation values (only valid in "PERT" mode).

`parallel` whether the script run in parallel mode; you also need to set n.cores to determine how much CPU resource to use.

`n.cores` how many threads/process to be assigned for this function; more threads used results in more resource of CPU and memory used.

`nfolds` integer > 2, how many folds to be created for n-folds cross-validation LASSO in `cv.glmnet`.

`lambda.type` character, which model should be used to obtain the variables selected in one bagging. Default is "lambda.1se" for less variables selected and lower probability being over-fitting. See the help of 'cv.glmnet' for more details.

`plot.freq` whether to show all the non-zero frequency in the final barplot or not. If "full", all the variables(including zero frequency) will be plotted. If "part", all the non-zero variables will be plotted. If "not", will not print the plot.

`plot.out` the file's name of the frequency plot. If set to FALSE, no plot will be output. If you run this function in Linux command line, you don't have to set this param for the plot.freq will output your plot to your current working directory with name "Rplot.pdf".Default to FALSE.

`do.plot` if TRUE generate result plots.

`output.dir` the path to save result files generated by `VSOLassoBag` (if not existed, will be created). Default is NA, will save in the same space as the current working dir.

`filter.method` the filter method applied to input matrix; default is 'auto', automatically select the filter method according to the data type of 'out.mat'. Specific supported methods are "pearson", "spearman", "kendall" from `cor.test` method, and "cox" from `coxph` method, and "none" (no filter applied).

`inbag.filter` if TRUE, apply filters to the re-sampled bagging samples rather than the original samples; default is TRUE.

`filter.thres.method`
the method determines the threshold of importance in filters. Supported methods are "fdr" and "rank".

`filter.thres.P` if 'filter.thres.method' is "fdr", use 'filter.thres.P' as the (adjusted) cut-off p-value. Default is 0.05.

`filter.rank.cutoff`
if 'filter.thres.method' is "rank", use 'filter.rank.cutoff' as the cut-off rank. Default is 0.05.

`filter.min.variables`
minimum important variables selected by filters. Useful when building a multivariable cox model since cox model can only be built on limited variables. Default is -Inf (not applied).

`filter.max.variables`
maximum important variables selected by filters. Useful when building a multivariable cox model since cox model can only be built on limited variables. Default is Inf (not applied).

`filter.result.report`
if TRUE generate filter reports for filter results, only vaild when 'inbag.filter' set to FALSE (i.e. only valid in out-bag filters mode).

`filter.report.all.variables`
: if TRUE report all variables in the filter report, only valid when 'filter.result.report' set to TRUE.

`post.regression`
: build a regression model based on the variables selected by VSOLassoBag process. Default is FALSE.

`post.LASSO`
: build a LASSO regression model based on the variables selected by VSOLassoBag process, only vaild when 'post.regression' set to TRUE.

`pvalue.cutoff`
: determine the cut-off p-value for what variables were selected by VSOLassoBag, only vaild when 'post.regression' is TRUE and 'bagFreq.sigMethod' set to "Parametric Statistical Test" or "Permutation Test".

`used.elbow.point`
: determine which elbow point to use if multiple elbow points were detected for what variables were selected by VSOLassoBag. Supported methods are "first", "middle" and "last". Default is "middle", use the middle one among all detected elbow points. Only vaild when 'post.regression' is TRUE and 'bagFreq.sigMethod' set to "Curve Elbow Point Detection".

Value

A list with (1) the result dataframe, "results", contains "variable" with selection frequency >=1 and their "Frequency", the "P.value" and the adjusted p value "P.adjust" of each variable (if set 'bagFreq.sigMethod' = "PST" or "PERT"), or the elbow point indicators "elbow.point", while elbow point(s) will be marked with "*" (if set 'bagFreq.sigMethod' = "CEP"). This is the main result VSOLassoBag obtained. (2) other utility results, including permutation results, "permutations", the regression model built on VSOLassoBag results, "model".

See Also

`glmnet` and `cv.glmnet` in R package *glmnet*.

Examples

```
data("ExpressionData")
set.seed(19084)

# binomial
VSOLassoBag(ExpressionData, "label", bootN=2, a.family="binomial",
bagFreq.sigMethod="PST", do.plot = FALSE, plot.freq = "not")


# gaussian
VSOLassoBag(ExpressionData, "y", bootN=2, a.family="gaussian",
bagFreq.sigMethod="PST", do.plot = FALSE, plot.freq = "not")
VSOLassoBag(ExpressionData, "y", bootN=2, a.family="gaussian",
bagFreq.sigMethod="CEP", do.plot = FALSE, plot.freq = "not")


# cox
VSOLassoBag(ExpressionData, c("time","status"), bootN=2,
```

```
a.family="cox", bagFreq.sigMethod="PST", do.plot = FALSE,
plot.freq = "not")
VSOLassoBag(ExpressionData, c("time","status"), bootN=2, a.family="cox",
bagFreq.sigMethod="CEP", do.plot = FALSE, plot.freq = "not")



# mgaussian
VSOLassoBag(ExpressionData, c("multi.label.D_1","multi.label.D_2"), bootN=2,
a.family="mgaussian", bagFreq.sigMethod="PST", do.plot = FALSE,
plot.freq = "not")
VSOLassoBag(ExpressionData, c("multi.label.D_1","multi.label.D_2"), bootN=2,
a.family="mgaussian", bagFreq.sigMethod="CEP", do.plot = FALSE,
plot.freq = "not")

# poisson
VSOLassoBag(ExpressionData, "pois", bootN=10, a.family="poisson",
bagFreq.sigMethod="PST", do.plot = FALSE, plot.freq = "not")
VSOLassoBag(ExpressionData, "pois", bootN=2, a.family="poisson",
bagFreq.sigMethod="CEP", do.plot = FALSE, plot.freq = "not")

# multi-thread processing is supported if run on a multi-thread,
# forking-supported platform (detailed see R package 'parallel'),
# which can significantly accelerate the process
# you can achieve this by flag 'parallel' to TRUE and set 'n.cores' to an
# integer larger than 1, depending on the available threads
# multi-thread processing using 2 threads
VSOLassoBag(ExpressionData, "y", bootN=1000, a.family="binomial",
bagFreq.sigMethod="PST", do.plot = FALSE, plot.freq = "not",
parallel=TRUE,n.cores=2)
```

Index