

Package ‘PCLassoCox’

March 27, 2021

Title A protein complex-based group lasso-Cox model for accurate prognosis and risk protein complex discovery

Version 0.0.0.9000

Description The PCLasso model is a prognostic model which selects important predictors at the protein complex level to achieve accurate prognosis and identify risk protein complexes.

Depends R (>= 3.5.0)

Imports grpreg, survival

License Artistic-2.0

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

URL <https://github.com/weiliu123/PCLassoCox>

BugReports <https://github.com/weiliu123/PCLassoCox/issues>

Suggests survminer (>= 0.4.9), ggplot2, survcomp

NeedsCompilation no

Author Wei Liu [aut, cre] (<<https://orcid.org/0000-0002-5496-3641>>)

Maintainer Wei Liu <freelw@qq.com>

R topics documented:

cv.PCLasso	2
ext2EntrezID	3
ext2Group	4
GBM	4
LUAD	5
OV	5
PCGroup	6
PCLasso	7
PCLassoCox	8
plot.cv.PCLasso	9
plot.PCLasso	10
plotKMCurve	11
plotRS	12

plotTDAUC	13
predict.cv.PCLasso	13
predict.PCLasso	15

Index	18
--------------	-----------

cv.PCLasso	<i>Cross-validation for PCLasso</i>
------------	-------------------------------------

Description

Perform k-fold cross validations for the PCLasso model with grouped covariates over a grid of values for the regularization parameter lambda.

Usage

```
cv.PCLasso(
  x,
  y,
  group,
  penalty = c("grLasso", "grMCP", "grSCAD", "gel", "cMCP"),
  nfolds = 5,
  standardize = TRUE,
  ...
)
```

Arguments

x	A n x p design matrix of gene expression measurements with n samples and p genes, as in PCLasso.
y	The time-to-event outcome, as a two-column matrix or Surv object, as in PCLasso. The first column should be time on study (follow up time); the second column should be a binary variable with 1 indicating that the event has occurred and 0 indicating (right) censoring.
group	A list of groups as in PCLasso. The feature (gene) names in group should be consistent with the feature (gene) names in x.
penalty	The penalty to be applied to the model. For group selection, one of grLasso, grMCP, or grSCAD. For bi-level selection, one of gel or cMCP. See grpsurv in the R package grpreg for details.
nfolds	The number of cross-validation folds. Default is 5.
standardize	Logical flag for x standardization, prior to fitting the model. Default is TRUE.
...	Arguments to be passed to cv.grpsurv in the R package grpreg.

Details

The function calls PCLasso nfolds times, each time leaving out 1/nfolds of the data. The cross-validation error is based on the deviance. The numbers for each outcome class are balanced across the folds; i.e., the number of outcomes in which y is equal to 1 is the same for each fold, or possibly off by 1 if the numbers do not divide evenly. cv.PCLasso uses the approach of calculating the full Cox partial likelihood using the cross-validated set of linear predictors. See cv.grpsurv in the R package grpreg for details.

Value

An object with S3 class "cv.PCLasso" containing:

cv.fit	An object of class "cv.grpsurv".
group.dt	Groups with features (genes) not included in x being filtered out.

Author(s)

Wei Liu

References

PCLasso: a protein complex-based group lasso-Cox model for accurate prognosis and risk protein complex discovery. To be published.

Park, H., Niida, A., Miyano, S. and Imoto, S. (2015) Sparse overlapping group lasso for integrative multi-omics analysis. Journal of computational biology: a journal of computational molecular cell biology, 22, 73-84.

See Also

[predict.cv.PCLasso](#)

Examples

```
# load data
data(GBM)
data(PCGroup)

x = GBM$GBM.train$Exp
y = GBM$GBM.train$survData

# fit model
cv.fit1 <- cv.PCLasso(x, y, group = PCGroup, penalty = "grLasso",
nolds = 10)
```

ext2EntrezID

Transform selection features into EntrezID

Description

Transform selection features into EntrezID at the gene level.

Usage

```
ext2EntrezID(x)
```

Arguments

x	Names of selected vars by PCLasso or cv.PCLasso.
---	--

Value

Selected genes.

ext2Group	<i>Convert selected features into ComplexID</i>
-----------	---

Description

Convert selected features into ComplexID at protein complex level

Usage

```
ext2Group(x)
```

Arguments

x Names of selected vars by PCLasso or cv.PCLasso.

Value

Selected protein complexes.

GBM	<i>The GBM datasets</i>
-----	-------------------------

Description

A dataset containing the GBM expression data and survival data.

Usage

```
data(GBM)
```

Format

A list with four GBM datasets:

GBM.train A list with two items. Exp: Expression data of GBM.train. survData: Survival data of GBM.train. The first column is the time on study (follow up time); the second column is a binary variable with 1 indicating that the event has occurred and 0 indicating(right) censoring.

GBM.test A list with two items. Exp: Expression data of GBM.test. survData: Survival data of GBM.test. The first column is the time on study (follow up time); the second column is a binary variable with 1 indicating that the event has occurred and 0 indicating(right) censoring.

GBM.valid1 A list with two items. Exp: Expression data of GBM.valid1. survData: Survival data of GBM.valid1. The first column is the time on study (follow up time); the second column is a binary variable with 1 indicating that the event has occurred and 0 indicating(right) censoring.

GBM.valid2 A list with two items. Exp: Expression data of GBM.valid2. survData: Survival data of GBM.valid2. The first column is the time on study (follow up time); the second column is a binary variable with 1 indicating that the event has occurred and 0 indicating(right) censoring.

Examples

```
data(GBM)
```

LUAD*The LUAD datasets*

Description

A dataset containing the LUAD expression data and survival data.

Usage

```
data(LUAD)
```

Format

A list with five LUAD datasets:

LUAD.train A list with two items. Exp: Expression data of LUAD.train. survData: Survival data of LUAD.train. The first column is the time on study (follow up time); the second column is a binary variable with 1 indicating that the event has occurred and 0 indicating(right) censoring.

LUAD.test A list with two items. Exp: Expression data of LUAD.test. survData: Survival data of LUAD.test. The first column is the time on study (follow up time); the second column is a binary variable with 1 indicating that the event has occurred and 0 indicating(right) censoring.

LUAD.valid1 A list with two items. Exp: Expression data of LUAD.valid1. survData: Survival data of LUAD.valid1. The first column is the time on study (follow up time); the second column is a binary variable with 1 indicating that the event has occurred and 0 indicating(right) censoring.

LUAD.valid2 A list with two items. Exp: Expression data of LUAD.valid2. survData: Survival data of LUAD.valid2. The first column is the time on study (follow up time); the second column is a binary variable with 1 indicating that the event has occurred and 0 indicating(right) censoring.

LUAD.valid3 A list with two items. Exp: Expression data of LUAD.valid3. survData: Survival data of LUAD.valid3. The first column is the time on study (follow up time); the second column is a binary variable with 1 indicating that the event has occurred and 0 indicating(right) censoring.

Examples

```
data(LUAD)
```

OV*The OV datasets*

Description

A dataset containing the OV expression data and survival data.

Usage

```
data(OV)
```

Format

A list with four OV datasets:

OV.train A list with two items. Exp: Expression data of OV.train. survData: Survival data of OV.train. The first column is the time on study (follow up time); the second column is a binary variable with 1 indicating that the event has occurred and 0 indicating(right) censoring.

OV.test A list with two items. Exp: Expression data of OV.test. survData: Survival data of OV.test. The first column is the time on study (follow up time); the second column is a binary variable with 1 indicating that the event has occurred and 0 indicating(right) censoring.

OV.valid1 A list with two items. Exp: Expression data of OV.valid1. survData: Survival data of OV.valid1. The first column is the time on study (follow up time); the second column is a binary variable with 1 indicating that the event has occurred and 0 indicating(right) censoring.

OV.valid2 A list with two items. Exp: Expression data of OV.valid2. survData: Survival data of OV.valid2. The first column is the time on study (follow up time); the second column is a binary variable with 1 indicating that the event has occurred and 0 indicating(right) censoring.

Examples

```
data(OV)
```

PCGroup

Protein complexes for "PCLasso"/"cv.PCLasso"

Description

A list of protein complexes. The genes in each protein complex are represented by EntrezID, which are consistent with the gene names in the expression data in GBM, LUAD, and OV.

Usage

```
data(PCGroup)
```

Format

A list of 2417 protein complexes.

Examples

```
data(PCGroup)
```

PCLasso

*Protein complex-based group lasso-Cox model***Description**

Construct a PCLasso model based on a gene expression matrix, survival data, and protein complexes.

Usage

```
PCLasso(
  x,
  y,
  group,
  penalty = c("grLasso", "grMCP", "grSCAD", "gel", "cMCP"),
  standardize = TRUE,
  ...
)
```

Arguments

<code>x</code>	A $n \times p$ matrix of gene expression measurements with n samples and p genes.
<code>y</code>	The time-to-event outcome, as a two-column matrix or <code>Surv</code> object. The first column should be time on study (follow up time); the second column should be a binary variable with 1 indicating that the event has occurred and 0 indicating (right) censoring.
<code>group</code>	A list of groups. The feature (gene) names in group should be consistent with the feature (gene) names in <code>x</code> .
<code>penalty</code>	The penalty to be applied to the model. For group selection, one of <code>grLasso</code> , <code>grMCP</code> , or <code>grSCAD</code> . For bi-level selection, one of <code>gel</code> or <code>cMCP</code> . See <code>grpsurv</code> in the R package <code>grpreg</code> for details.
<code>standardize</code>	Logical flag for <code>x</code> standardization, prior to fitting the model. Default is <code>TRUE</code> .
<code>...</code>	Arguments to be passed to <code>grpsurv</code> in the R package <code>grpreg</code> .

Details

The PCLasso model is a prognostic model which selects important predictors at the protein complex level to achieve accurate prognosis and identify risk protein complexes. The PCLasso model has three inputs: a gene expression matrix, survival data, and protein complexes. It estimates the correlation between gene expression in protein complexes and survival data at the level of protein complexes. Similar to the traditional Lasso-Cox model, PCLasso is based on the Cox PH model and estimates the Cox regression coefficients by maximizing partial likelihood with regularization penalty. The difference is that PCLasso selects features at the level of protein complexes rather than individual genes. Considering that genes usually function by forming protein complexes, PCLasso regards genes belonging to the same protein complex as a group, and constructs a $l_{1/2}$ penalty based on the sum (i.e., l_1 norm) of the l_2 norms of the regression coefficients of the group members to perform the selection of features at the group level. Since a gene may belong to multiple protein complexes, that is, there is overlap between protein complexes, the classical group Lasso-Cox model for non-overlapping groups may lead to false sparse solutions. The PCLasso model deals with the overlapping problem of protein complexes by constructing a latent group Lasso-Cox model. And

by reconstructing the gene expression matrix of the protein complexes, the latent group Lasso-Cox model is transformed into a non-overlapping group Lasso-Cox model in an expanded space, which can be directly solved using the classical group Lasso method. Through the final sparse solution, we can predict the patient's risk score based on a small set of protein complexes and identify risk protein complexes that are frequently selected to construct prognostic models.

Value

An object with S3 class "PCLasso" containing:

<code>fit</code>	An object of class "grpsurv"
<code>group.dt</code>	Groups with features (genes) not included in <code>x</code> being filtered out.

References

PCLasso: a protein complex-based group lasso-Cox model for accurate prognosis and risk protein complex discovery. To be published.

Park, H., Niida, A., Miyano, S. and Imoto, S. (2015) Sparse overlapping group lasso for integrative multi-omics analysis. *Journal of computational biology: a journal of computational molecular cell biology*, 22, 73-84.

See Also

[predict.PCLasso](#), [cv.PCLasso](#)

Examples

```
# load data
data(GBM)
data(PCGroup)

x = GBM$GBM.train$Exp
y = GBM$GBM.train$survData

# fit model
fit1 <- PCLasso(x, y, group = PCGroup, penalty = "grLasso")
```

PCLassoCox

A protein complex-based group lasso-Cox model for accurate prognosis and risk protein complex discovery

Description

The PCLasso model is a prognostic model which selects important predictors at the protein complex level to achieve accurate prognosis and identify risk protein complexes.

Details

Index: This package was not yet installed at build time.

The PCLasso model accepts a gene expression matrix, survival data, and protein complexes for the PCLasso model, and makes predictions for new samples and identifies risk protein complexes.

PCLasso constructs a PCLasso model based on a gene expression matrix, survival data, and protein complexes.

predict.PCLasso makes predictions from a PCLasso model.

cv.PCLasso performs k-fold cross validations for the PCLasso model with grouped covariates over a grid of values for the regularization parameter lambda, and returns an optimal value for lambda.

predict.cv.PCLasso returns predictions from a fitted cv.PCLasso object, using the optimal value chosen for lambda.

plot.PCLasso produces a plot of the coefficient paths for a fitted PCLasso object.

plot.cv.PCLasso plots the cross-validation curve from a cv.PCLasso object, along with standard error bars.

References

PCLasso: a protein complex-based group lasso-Cox model for accurate prognosis and risk protein complex discovery. To be published.

Park, H., Niida, A., Miyano, S. and Imoto, S. (2015) Sparse overlapping group lasso for integrative multi-omics analysis. Journal of computational biology: a journal of computational molecular cell biology, 22, 73-84.

plot.cv.PCLasso	<i>Plot the cross-validation curve from a cv.PCLasso object</i>
-----------------	---

Description

Plot the cross-validation curve from a cv.PCLasso object, along with standard error bars.

Usage

```
## S3 method for class 'cv.PCLasso'
plot(x, type = c("cve", "rsq", "snr", "all"), norm = NULL, ...)
```

Arguments

x	Fitted cv.PCLasso model.
type	What to plot on the vertical axis. "cve" plots the cross-validation error (deviance); "rsq" plots an estimate of the fraction of the deviance explained by the model (R-squared); "snr" plots an estimate of the signal-to-noise ratio; "all" produces all of the above.
norm	If TRUE, plot the norm of each group, rather than the individual coefficients.
...	Other graphical parameters to plot

Details

Error bars representing approximate ± 1 SE (68% intervals) are plotted along with the estimates at value of lambda. See plot.cv.grpreg in the R package grpreg for details.

See Also[cv.PCLasso](#)**Examples**

```
# load data
data(GBM)
data(PCGroup)
x = GBM$GBM.train$Exp
y = GBM$GBM.train$survData

cv.fit1 <- cv.PCLasso(x, y, group = PCGroup, penalty = "grLasso", nfolds = 10)

# plot the norm of each group
plot(cv.fit1, norm = TRUE)

# plot the individual coefficients
plot(cv.fit1, norm = FALSE)

# plot the cross-validation error (deviance)
plot(cv.fit1, type = "cve")
```

plot.PCLasso

*Plot coefficients from a PCLasso object***Description**

Produces a plot of the coefficient paths for a fitted PCLasso object.

Usage

```
## S3 method for class 'PCLasso'
plot(x, norm = TRUE, ...)
```

Arguments

x	Fitted PCLasso model.
norm	If TRUE, plot the norm of each group, rather than the individual coefficients.
...	Other graphical parameters to plot.

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

```
# load data
data(GBM)
data(PCGroup)

x = GBM$GBM.train$Exp
y = GBM$GBM.train$survData
```

```
# fit the PCLasso model
fit1 <- PCLasso(x, y, group = PCGroup, penalty = "grLasso")

# plot the norm of each group
plot(fit1, norm = TRUE)

# plot the individual coefficients
plot(fit1, norm = FALSE)
```

plotKMCurve

*Plot Kaplan-Meier curve***Description**

Plot Kaplan-Meier curve

Usage

```
plotKMCurve(
  rs,
  y,
  cutoff = median(rs),
  col.low = "#1F77B4",
  col.high = "#FF7F0E"
)
```

Arguments

rs	Risk score of patients
y	The time-to-event outcome, as a two-column matrix or Surv object. The first column should be time on study (follow up time); the second column should be a binary variable with 1 indicating that the event has occurred and 0 indicating (right) censoring.
cutoff	The cut-off that is used to divided the patients into two groups according to rs.
col.low	Color used to indicate low-risk patients.
col.high	Color used to indicate high-risk patients.

Examples

```
# load data
data(GBM)
data(PCGroup)

cv.fit1 <- cv.PCLasso(x = GBM$GBM.train$Exp,
                      y = GBM$GBM.train$survData,
                      group = PCGroup,
                      nfolds = 5)

# predict risk scores of samples in GBM.test
s <- predict(object = cv.fit1, x = GBM$GBM.test$Exp, type="link",
             lambda=cv.fit1$cv.fit$lambda.min)

plotKMCurve(rs = s, y = GBM$GBM.test$survData)
```

plotRS	<i>Plot risk scores of patients</i>
--------	-------------------------------------

Description

Plot risk scores of patients

Usage

```
plotRS(
  rs,
  y.status,
  cutoff = median(rs),
  col.dead = "#E64B35",
  col.alive = "#00A087",
  legend.position = c(0.2, 0.9)
)
```

Arguments

rs	Risk score of patients
y.status	A binary variable with 1 indicating that the event has occurred and 0 indicating (right) censoring.
cutoff	The cut-off that is used to divided the patients into two groups according to rs.
col.dead	The color that corresponds to 1 in y.status.
col.alive	The color that corresponds to 0 in y.status.
legend.position	Legend position.

Examples

```
# load data
data(GBM)
data(PCGroup)

cv.fit1 <- cv.PCLasso(x = GBM$GBM.train$Exp,
                     y = GBM$GBM.train$survData,
                     group = PCGroup,
                     nfolds = 5)

# predict risk scores of samples in GBM.test
s <- predict(object = cv.fit1, x = GBM$GBM.test$Exp, type="link",
             lambda=cv.fit1$cv.fit$lambda.min)

plotRS(rs = s, y.status = GBM$GBM.test$survData$status)
```

plotTDAUC	<i>Title</i>
-----------	--------------

Description

Title

Usage

```
plotTDAUC(rs, y, npoint = 50, col = "red")
```

Arguments

rs	Risk score of patients
y	The time-to-event outcome, as a two-column matrix or Surv object. The first column should be time on study (follow up time); the second column should be a binary variable with 1 indicating that the event has occurred and 0 indicating (right) censoring.
npoint	Integer: number of time points for calculating time-dependent AUC.
col	Color of the time-dependent AUC curve.

Examples

```
# load data
data(GBM)
data(PCGroup)

cv.fit1 <- cv.PCLasso(x = GBM$GBM.train$Exp,
                     y = GBM$GBM.train$survData,
                     group = PCGroup,
                     nfolds = 5)

# predict risk scores of samples in GBM.test
s <- predict(object = cv.fit1, x = GBM$GBM.test$Exp, type="link",
             lambda=cv.fit1$cv.fit$lambda.min)

plotTDAUC(rs = s, y = GBM$GBM.test$survData)
```

predict.cv.PCLasso	<i>Make predictions from a cross-validated PCLasso model</i>
--------------------	--

Description

Similar to other predict methods, this function returns predictions from a fitted "cv.PCLasso" object, using the optimal value chosen for lambda.

Usage

```
## S3 method for class 'cv.PCLasso'
predict(
  object,
  x = NULL,
  type = c("link", "response", "survival", "median", "norm", "coefficients", "vars",
    "nvars", "vars.unique", "nvars.unique", "groups", "ngroups"),
  lambda,
  ...
)
```

Arguments

object	Fitted cv.PCLasso model object.
x	Matrix of values at which predictions are to be made. The features (genes) contained in x should be consistent with those contained in x in the PCLasso function. Not used for type="coefficients" or for some of the type settings in predict.
type	Type of prediction: "link" returns the linear predictors; "response" gives the risk (i.e., $\exp(\text{link})$); "vars" returns the indices for the nonzero coefficients; "vars.unique" returns unique features (genes) with nonzero coefficients (If a feature belongs to multiple groups and multiple groups are selected, the feature will be repeatedly selected. Compared with "var", "var.unique" will filter out repeated features.); "groups" returns the groups with at least one nonzero coefficient; "nvars" returns the number of nonzero coefficients; "nvars.unique" returns the number of unique features (genes) with nonzero coefficients; "ngroups" returns the number of groups with at least one nonzero coefficient; "norm" returns the L2 norm of the coefficients in each group. "survival" returns the estimated survival function; "median" estimates median survival times.
lambda	Values of the regularization parameter lambda at which predictions are requested. For values of lambda not in the sequence of fitted models, linear interpolation is used.
...	Arguments to be passed to predict.cv.grpsurv in the R package grpreg.

Value

The object returned depends on type.

See Also

[cv.PCLasso](#)

Examples

```
# load data
data(GBM)
data(PCGroup)

cv.fit1 <- cv.PCLasso(x = GBM$GBM.train$Exp,
  y = GBM$GBM.train$survData,
  group = PCGroup,
  nfolds = 5)
```

```
# predict risk scores of samples in x.test
s <- predict(object = cv.fit1, x = GBM$GBM.test$Exp, type="link",
             lambda=cv.fit1$cv.fit$lambda.min)

# Nonzero coefficients
sel.groups <- predict(object = cv.fit1, type="groups",
                     lambda = cv.fit1$cv.fit$lambda.min)
sel.ngroups <- predict(object = cv.fit1, type="ngroups",
                     lambda = cv.fit1$cv.fit$lambda.min)
sel.vars.unique <- predict(object = cv.fit1, type="vars.unique",
                        lambda = cv.fit1$cv.fit$lambda.min)
sel.nvars.unique <- predict(object = cv.fit1, type="nvars.unique",
                        lambda = cv.fit1$cv.fit$lambda.min)
sel.vars <- predict(object = cv.fit1, type="vars",
                  lambda=cv.fit1$cv.fit$lambda.min)
sel.nvars <- predict(object = cv.fit1, type="nvars",
                  lambda=cv.fit1$cv.fit$lambda.min)
```

predict.PCLasso

Make predictions from a PCLasso model

Description

Similar to other predict methods, this function returns predictions from a fitted PCLasso object.

Usage

```
## S3 method for class 'PCLasso'
predict(
  object,
  x = NULL,
  type = c("link", "response", "survival", "median", "norm", "coefficients", "vars",
           "nvars", "vars.unique", "nvars.unique", "groups", "ngroups"),
  lambda,
  ...
)
```

Arguments

object	Fitted PCLasso model object.
x	Matrix of values at which predictions are to be made. The features (genes) contained in x should be consistent with those contained in x in the PCLasso function. Not used for type="coefficients" or for some of the type settings in predict.
type	Type of prediction: "link" returns the linear predictors; "response" gives the risk (i.e., exp(link)); "vars" returns the indices for the nonzero coefficients; "vars.unique" returns unique features (genes) with nonzero coefficients (If a feature belongs to multiple groups and multiple groups are selected, the feature will be repeatedly selected. Compared with "var", "var.unique" will filter out repeated features.); "groups" returns the groups with at least one nonzero coefficient; "nvars" returns the number of nonzero coefficients; "nvars.unique" returns the number of unique features (genes) with nonzero coefficients; "ngroups"

	returns the number of groups with at least one nonzero coefficient; "norm" returns the L2 norm of the coefficients in each group. "survival" returns the estimated survival function; "median" estimates median survival times.
lambda	Values of the regularization parameter lambda at which predictions are requested. For values of lambda not in the sequence of fitted models, linear interpolation is used.
...	Arguments to be passed to <code>predict.grpsurv</code> in the R package <code>grpreg</code> .

Details

See `predict.grpsurv` in the R package `grpreg` for details.

Value

The object returned depends on type.

See Also

[PCLasso](#)

Examples

```
# load data
data(GBM)
data(PCGroup)

fit1 <- PCLasso(x = GBM$GBM.train$Exp, y = GBM$GBM.train$survData, group =
PCGroup)

# predict risk scores of samples in x.test
s <- predict(object = fit1, x = GBM$GBM.test$Exp, type="link",
lambda=fit1$fit$lambda)

s <- predict(object = fit1, x = GBM$GBM.test$Exp, type="link",
lambda=fit1$fit$lambda[10])

s <- predict(object = fit1, x = GBM$GBM.test$Exp, type="link", lambda=c(0.1,
0.01))

# Nonzero coefficients
sel.groups <- predict(object = fit1, type="groups",
                      lambda = fit1$fit$lambda)
sel.ngroups <- predict(object = fit1, type="ngroups",
                      lambda = fit1$fit$lambda)
sel.vars.unique <- predict(object = fit1, type="vars.unique",
                          lambda = fit1$fit$lambda)
sel.nvars.unique <- predict(object = fit1, type="nvars.unique",
                           lambda = fit1$fit$lambda)
sel.vars <- predict(object = fit1, type="vars",
                   lambda=fit1$fit$lambda)
sel.nvars <- predict(object = fit1, type="nvars",
                    lambda=fit1$fit$lambda)

# For values of lambda not in the sequence of fitted models,
# linear interpolation is used.
sel.groups <- predict(object = fit1, type="groups",
```



```
        lambda = c(0.1, 0.01))
sel.ngroups <- predict(object = fit1, type="ngroups",
        lambda = c(0.1, 0.01))
sel.vars.unique <- predict(object = fit1, type="vars.unique",
        lambda = c(0.1, 0.01))
sel.nvars.unique <- predict(object = fit1, type="nvars.unique",
        lambda = c(0.1, 0.01))
sel.vars <- predict(object = fit1, type="vars",
        lambda=c(0.1, 0.01))
sel.nvars <- predict(object = fit1, type="nvars",
        lambda=c(0.1, 0.01))
```

Index

* datasets

GBM, [4](#)

LUAD, [5](#)

OV, [5](#)

PCGroup, [6](#)

cv.PCLasso, [2](#), [8](#), [10](#), [14](#)

ext2EntrezID, [3](#)

ext2Group, [4](#)

GBM, [4](#)

LUAD, [5](#)

OV, [5](#)

PCGroup, [6](#)

PCLasso, [7](#), [10](#), [16](#)

PCLassoCox, [8](#)

plot.cv.PCLasso, [9](#)

plot.PCLasso, [10](#)

plotKMCurve, [11](#)

plotRS, [12](#)

plotTDAUC, [13](#)

predict.cv.PCLasso, [3](#), [13](#)

predict.PCLasso, [8](#), [15](#)