Package 'PCLassoReg'

Title Group Regression Models for Risk Protein Complex Identification

Version 1.0.0

Description Two protein complex-based group regression models (PCLasso and PCLasso2) for risk protein complex identification. PCLasso is a prognostic model that identifies risk protein complexes associated with survival. PCLasso2 is a classification model that identifies risk protein complexes associated with classes. For more information, see Wang and Liu (2021) <doi:10.1093/bib/bbab212>.

Depends R (>= 3.5.0)

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Imports survival, grpreg

URL https://github.com/weiliu123/PCLassoReg

BugReports https://github.com/weiliu123/PCLassoReg/issues

Suggests rmarkdown, knitr

VignetteBuilder knitr

NeedsCompilation no

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

Maintainer Wei Liu <freelw@qq.com>

Repository CRAN

Contents

---

classData   *A dataset for classification*

Description

A dataset for classification

Usage

```
classData
```

Format

A list containing a protein expression matrix and a response vector

Exp a protein expression matrix

Label a response vector

---

cv.PCLasso   *Cross-validation for* PCLasso

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"),
  nfolds = 5,
  standardize = TRUE,
  ...
)
```

Arguments

x	A n x p design matrix of gene/protein expression measurements with n samples and p genes/proteins, 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/protein) names in `group` should be consistent with the feature (gene/protein) 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 censored samples are balanced across the folds. `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".
complexes.dt	Complexes with features (genes/proteins) 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. Brief Bioinform, 2021.

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(survivalData)
data(PCGroups)

x = survivalData$Exp
y = survivalData$survData

PC.Human <- getPCGroups(Groups = PCGroups, Organism = "Human",
Type = "EntrezID")

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

---

cv.PCLasso2   *Cross-validation for* PCLasso2

Description

Perform k-fold cross validations for the PCLasso2 model with grouped covariates over a grid of values for the regularization parameter `lambda`.

Usage

```
cv.PCLasso2(
  x,
  y,
  group,
  penalty = c("grLasso", "grMCP", "grSCAD"),
  family = c("binomial", "gaussian", "poisson"),
  nfolds = 5,
  gamma = 8,
  standardize = TRUE,
  ...
)
```

Arguments

x	A n x p design matrix of gene/protein expression measurements with n samples and p genes/proteins, as in `PCLasso2`.
y	The response vector.
group	A list of groups as in `PCLasso`. The feature (gene/protein) names in `group` should be consistent with the feature (gene/protein) names in `x`.
penalty	The penalty to be applied to the model. For group selection, one of grLasso, grMCP, or grSCAD. See `grpreg` in the R package `grpreg` for details.
family	Either "binomial" or "gaussian", depending on the response.
nfolds	The number of cross-validation folds. Default is 5.
gamma	Tuning parameter of the `grSCAD`/`grMCP` penalty. Default is 8.
standardize	Logical flag for x standardization, prior to fitting the model. Default is `TRUE`.
...	Arguments to be passed to `cv.grpreg` in the R package `grpreg`.

Details

The function calls `PCLasso2` nfolds times, each time leaving out 1/nfolds of the data. The cross-validation error is based on the deviance. The numbers for each 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. See `cv.grpreg` in the R package `grpreg` for details.

Value

An object with S3 class "cv.PCLasso2" containing:

cv.fit	An object of class "cv.grpreg".
complexes.dt	Complexes with features (genes/proteins) not included in x being filtered out.

Author(s)

Wei Liu

References

PCLasso2: a protein complex-based, group Lasso-logistic model for 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.PCLasso2

Examples

```
# load data
data(classData)
data(PCGroups)

x = classData$Exp
y = classData$Label

PC.Human <- getPCGroups(Groups = PCGroups, Organism = "Human",
Type = "GeneSymbol")

# fit model
cv.fit1 <- cv.PCLasso2(x, y, group = PC.Human, penalty = "grLasso",
family = "binomial", nfolds = 5)


cv.fit1 <- cv.PCLasso2(x, y, group = PC.Human, penalty = "grSCAD",
family = "binomial", nfolds = 5, gamma = 10)
cv.fit1 <- cv.PCLasso2(x, y, group = PC.Human, penalty = "grMCP",
family = "binomial", nfolds = 5, gamma = 15)
```

---

getPCGroups   *get protein complexes*

Description

get protein complexes

Usage

```
getPCGroups(
  Groups,
  Organism = c("Human", "Mouse", "Rat", "Mammalia", "Bovine", "Dog", "Rabbit"),
  Type = c("GeneSymbol", "EntrezID", "UniprotID")
)
```

Arguments

Groups	A data frame containing the protein complexes
Organism	Organism. one of Human, Mouse, Rat, Mammalia, Bovine, Dog, or Rabbit.
Type	The name type of the proteins in the protein complexes. One of GeneSymbol, EntrezID, UniprotID

Value

A list of protein complexes

Examples

```
data(PCGroups)
PC.Human <- getPCGroups(Groups = PCGroups, Organism = "Human",
Type = "GeneSymbol")
```

PCGroups   *Protein complexes.*

Description

A dataset containing the protein complexes

Usage

```
PCGroups
```

Format

A data frame with 3512 rows and 6 variables:

ComplexID ID of the protein complex

ComplexName name of the protein complex

Organism organism

UniprotID Uniprot IDs of the proteins in the protein complex

EntrezID Entrez IDs of the proteins in the protein complex

GeneSymbol gene symbols of the proteins in the protein complex

Source

<https://mips.helmholtz-muenchen.de/corum/>

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

Description

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

Usage

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

Arguments

x	A n x p matrix of gene/protein expression measurements with n samples and p genes/proteins.
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.
group	A list of groups. The feature (gene/protein) names in `group` should be consistent with the feature (gene/protein) names in `x`.
penalty	The penalty to be applied to the model. For group selection, one of grLasso, grMCP, or grSCAD. See `grpsurv` in the R package `grpreg` for details.
standardize	Logical flag for x standardization, prior to fitting the model. Default is `TRUE`.
...	Arguments to be passed to `grpsurv` in the R package `grpreg`.

Details

The function `PCLasso` implements the PCLasso model when the parameter `penalty` is set to "gr-Lasso". 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 l1/l2 penalty based on the sum (i.e., l1 norm) of the l2 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. The

penalty parameters `grSCAD` and `grMCP` can also be used to identify survival-related risk protein complexes. Their penalty for large coefficients is smaller than `grLasso`, so they tend to choose less risk protein complexes.

Value

An object with S3 class \code{PCLasso} containing:

fit   An object of class grpsurv

complexes.dt   Complexes with features (genes/proteins) not included in x being filtered out.

References

PCLasso: a protein complex-based, group lasso-Cox model for accurate prognosis and risk protein complex discovery. Brief Bioinform, 2021.

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(survivalData)
data(PCGroups)

x = survivalData$Exp
y = survivalData$survData

PC.Human <- getPCGroups(Groups = PCGroups, Organism = "Human",
Type = "EntrezID")

# fit PCLasso model
fit.PCLasso <- PCLasso(x, y, group = PC.Human, penalty = "grLasso")

# fit PCSCAD model
fit.PCSCAD <- PCLasso(x, y, group = PC.Human, penalty = "grSCAD")

# fit PCMCP model
fit.PCMCP <- PCLasso(x, y, group = PC.Human, penalty = "grMCP")
```

PCLasso2   *Protein complex-based group Lasso-logistic model*

Description

Protein complex-based group Lasso-logistic model

Usage

```
PCLasso2(
  x,
  y,
  group,
  penalty = c("grLasso", "grMCP", "grSCAD"),
  family = c("binomial", "gaussian", "poisson"),
  gamma = 8,
  standardize = TRUE,
  ...
)
```

Arguments

x	A n x p matrix of gene/protein expression measurements with n samples and p genes/proteins.
y	The response vector.
group	A list of groups. The feature (gene/protein) names in group should be consistent with the feature (gene/protein) names in x.
penalty	The penalty to be applied to the model. For group selection, one of grLasso, grMCP, or grSCAD. See grpreg in the R package grpreg for details.
family	Either "binomial" or "gaussian", depending on the response.
gamma	Tuning parameter of the grSCAD/grMCP penalty. Default is 8.
standardize	Logical flag for x standardization, prior to fitting the model. Default is TRUE.
...	Arguments to be passed to grpreg in the R package grpreg.

Details

The PCLasso2 model is a classification model that selects important predictors at the protein complex level to achieve accurate classification and identify risk protein complexes. The PCLasso2 model has three inputs: a protein expression matrix, a vector of binary response variables, and a number of known protein complexes. It estimates the correlation between protein expression and response variable at the level of protein complexes. Similar to traditional Lasso-logistic model, PCLasso2 is based on the logistic regression model and estimates the logistic regression coefficients by maximizing likelihood function with regularization penalty. The difference is that PCLasso2 selects features at the level of protein complexes rather than individual proteins. Considering that proteins usually function by forming protein complexes, PCLasso2 regards proteins belonging to

the same protein complex as a group and constructs a group Lasso penalty (l1/l2 penalty) based on the sum (i.e. l1 norm) of the l2 norms of the regression coefficients of the group members to perform the selection of features at the group level. With the group Lasso penalty, PCLasso2 trains the logistic regression model and obtains a sparse solution at the protein complex level, that is, the proteins belonging to a protein complex are either wholly included or wholly excluded from the model. PCLasso2 outputs a prediction model and a small set of protein complexes included in the model, which are referred to as risk protein complexes. The PCSCAD and PCMCP are performed by setting the penalty parameter `penalty` as `grSCAD` and `grMCP`, respectively.

Value

An object with S3 class `PCLasso2` containing:

`fit`	An object of class `grpreg`
`Complexes.dt`	Complexes with features (genes/proteins) not included in x being filtered out.

References

PCLasso2: a protein complex-based, group Lasso-logistic model for risk protein complex discovery. To be published.

PCLasso: a protein complex-based, group lasso-Cox model for accurate prognosis and risk protein complex discovery. Brief Bioinform, 2021.

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.PCLasso2, cv.PCLasso2

Examples

```
# load data
data(classData)
data(PCGroups)

x = classData$Exp
y = classData$Label

PC.Human <- getPCGroups(Groups = PCGroups, Organism = "Human",
Type = "GeneSymbol")

# fit PCLasso2 model
fit.PCLasso2 <- PCLasso2(x, y, group = PC.Human, penalty = "grLasso",
family = "binomial")

# fit PCSCAD model
fit.PCSCAD <- PCLasso2(x, y, group = PC.Human, penalty = "grSCAD",
family = "binomial", gamma = 10)
```

```
# fit PCMCP model
fit.PCMCP <- PCLasso2(x, y, group = PC.Human, penalty = "grMCP",
family = "binomial", gamma = 9)
```

PCLassoReg   *Group Regression Models for Risk Protein Complex Identification*

Description

Two protein complex-based group regression models (PCLasso and PCLasso2) for risk protein complex identification. PCLasso is a prognostic model that identifies risk protein complexes associated with survival. PCLasso2 is a classification model that identifies risk protein complexes associated with classes. For more information, see Wang and Liu (2021) <doi:10.1093/bib/bbab212>.

Details

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

The PCLasso2 model accepts a protein expression matrix, a response vector, and protein complexes for training the classification model, and makes predictions for new samples and identifies risk protein complexes associated with classes.

Both PCLasso and PCLasso2 use grLasso as the penalty function. The other two penalties grSCAD and grMCP can also be used for model construction and risk protein complex identification. The package also provides methods for plotting coefficient paths and cross-validation curves.

References

PCLasso2: a protein complex-based, group Lasso-logistic model for risk protein complex discovery. To be published.

PCLasso: a protein complex-based group lasso-Cox model for accurate prognosis and risk protein complex discovery. Brief Bioinform, 2021.

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   *Plot the cross-validation curve from a* `cv.PCLasso` *object*

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% confidence intervals) are plotted along with the estimates at value of lambda. See `plot.cv.grpreg` in the R package `grpreg` for details.

Value

No return value, called for plotting of `cv.PCLasso` objects.

See Also

[cv.PCLasso](#)

Examples

```
# load data
data(survivalData)
data(PCGroups)

x = survivalData$Exp
y = survivalData$survData

PC.Human <- getPCGroups(Groups = PCGroups, Organism = "Human",
Type = "EntrezID")

# fit model
```

```
cv.fit1 <- cv.PCLasso(x, y, group = PC.Human, 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.cv.PCLasso2   *Plot the cross-validation curve from a* cv.PCLasso2 *object*

Description

Plot the cross-validation curve from a cv.PCLasso2 object, along with standard error bars.

Usage

```
## S3 method for class 'cv.PCLasso2'
plot(x, type = c("cve", "rsq", "snr", "all"), norm = NULL, ...)
```

Arguments

x	Fitted cv.PCLasso2 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% confidence intervals) are plotted along with the estimates at value of lambda. See plot.cv.grpreg in the R package grpreg for details.

Value

No return value, called for plotting of cv.PCLasso2 objects.

See Also

cv.PCLasso2

Examples

```
# load data
data(classData)
data(PCGroups)

x = classData$Exp
y = classData$Label

PC.Human <- getPCGroups(Groups = PCGroups, Organism = "Human",
Type = "GeneSymbol")

# fit model
cv.fit1 <- cv.PCLasso2(x, y, group = PC.Human, penalty = "grLasso",
family = "binomial", 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.

Value

No return value, called for plotting of PCLasso objects.

See Also

PCLasso

Examples

```
# load data
data(survivalData)
data(PCGroups)

x = survivalData$Exp
y = survivalData$survData

PC.Human <- getPCGroups(Groups = PCGroups, Organism = "Human",
Type = "EntrezID")

# fit PCLasso model
fit.PCLasso <- PCLasso(x, y, group = PC.Human, penalty = "grLasso")

# plot the norm of each group
plot(fit.PCLasso, norm = TRUE)

# plot the individual coefficients
plot(fit.PCLasso, norm = FALSE)
```

---

plot.PCLasso2 *Plot coefficients from a PCLasso2 object*

Description

Produces a plot of the coefficient paths for a fitted PCLasso2 object.

Usage

```
## S3 method for class 'PCLasso2'
plot(x, norm = TRUE, ...)
```

Arguments

x	Fitted PCLasso2 model.
norm	If TRUE, plot the norm of each group, rather than the individual coefficients.
...	Other graphical parameters to plot.

Value

No return value, called for plotting of PCLasso2 objects.

See Also

[PCLasso2](#)

Examples

```
# load data
data(classData)
data(PCGroups)

x = classData$Exp
y = classData$Label

PC.Human <- getPCGroups(Groups = PCGroups, Organism = "Human",
Type = "GeneSymbol")

# fit PCLasso2 model
fit.PCLasso2 <- PCLasso2(x, y, group = PC.Human, penalty = "grLasso")

# plot the norm of each group
plot(fit.PCLasso2, norm = TRUE)

# plot the individual coefficients
plot(fit.PCLasso2, norm = FALSE)
```

---

predict.cv.PCLasso   *Make predictions from a cross-validated PCLasso model*

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/proteins)
                contained in x should be consistent with those contained in x in the cv.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/proteins) 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" returens the number of unique features (genes/proteins) 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(survivalData)
data(PCGroups)

x <- survivalData$Exp
y <- survivalData$survData
PC.Human <- getPCGroups(Groups = PCGroups, Organism = "Human",
Type = "EntrezID")

set.seed(20150122)
idx.train <- sample(nrow(x), round(nrow(x)*2/3))
x.train <- x[idx.train,]
y.train <- y[idx.train,]
x.test <- x[-idx.train,]
y.test <- y[-idx.train,]

# fit cv.PCLasso model
cv.fit1 <- cv.PCLasso(x = x.train,
                      y = y.train,
                      group = PC.Human,
                      nfolds = 5)

# predict risk scores of samples in x.test
s <- predict(object = cv.fit1, x = x.test, 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.cv.PCLasso2   *Make predictions from a cross-validated PCLasso2 model*

Description

Similar to other predict methods, this function returns predictions from a fitted `cv.PCLasso2` object, using the optimal value chosen for `lambda`.

Usage

```
## S3 method for class 'cv.PCLasso2'
predict(
  object,
  x = NULL,
  type = c("link", "response", "class", "norm", "coefficients", "vars", "nvars",
    "vars.unique", "nvars.unique", "groups", "ngroups"),
  lambda,
  ...
)
```

Arguments

object	Fitted `cv.PCLasso2` model object.
x	Matrix of values at which predictions are to be made. The features (genes/proteins) contained in x should be consistent with those contained in x in the `cv.PCLasso2` 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)); "class" returns the binomial outcome with the highest probability; "vars" returns the indices for the nonzero coefficients; "vars.unique" returns unique features (genes/proteins) 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/proteins) 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.

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.grpreg` in the R package `grpreg`.

Value

The object returned depends on `type`.

See Also

[`cv.PCLasso2`](#)

Examples

```
# load data
data(classData)
data(PCGroups)

x = classData$Exp
y = classData$Label

PC.Human <- getPCGroups(Groups = PCGroups, Organism = "Human",
Type = "GeneSymbol")

#' set.seed(20150122)
idx.train <- sample(nrow(x), round(nrow(x)*2/3))
x.train <- x[idx.train,]
y.train <- y[idx.train]
x.test <- x[-idx.train,]
y.test <- y[-idx.train]

# fit model
cv.fit1 <- cv.PCLasso2(x = x.train, y = y.train, group = PC.Human,
penalty = "grLasso", family = "binomial", nfolds = 10)

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

# predict classes of samples in x.test
s <- predict(object = cv.fit1, x = x.test, type="class",
             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/proteins) 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/proteins) 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/proteins) 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.grpsurv` in the R package `grpreg`.

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(survivalData)
data(PCGroups)

x <- survivalData$Exp
y <- survivalData$survData
PC.Human <- getPCGroups(Groups = PCGroups, Organism = "Human",
Type = "EntrezID")

set.seed(20150122)
idx.train <- sample(nrow(x), round(nrow(x)*2/3))
x.train <- x[idx.train,]
y.train <- y[idx.train,]
x.test <- x[-idx.train,]
y.test <- y[-idx.train,]

# fit PCLasso model
fit.PCLasso <- PCLasso(x = x.train, y = y.train, group = PC.Human,
                   penalty = "grLasso")

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

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

# Nonzero coefficients
sel.groups <- predict(object = fit.PCLasso, type="groups",
```

```
                              lambda = fit.PCLasso$fit$lambda)
sel.ngroups <- predict(object = fit.PCLasso, type="ngroups",
                          lambda = fit.PCLasso$fit$lambda)
sel.vars.unique <- predict(object = fit.PCLasso, type="vars.unique",
                             lambda = fit.PCLasso$fit$lambda)
sel.nvars.unique <- predict(object = fit.PCLasso, type="nvars.unique",
                              lambda = fit.PCLasso$fit$lambda)
sel.vars <- predict(object = fit.PCLasso, type="vars",
                      lambda=fit.PCLasso$fit$lambda)
sel.nvars <- predict(object = fit.PCLasso, type="nvars",
                       lambda=fit.PCLasso$fit$lambda)

# For values of lambda not in the sequence of fitted models,
# linear interpolation is used.
sel.groups <- predict(object = fit.PCLasso, type="groups",
                         lambda = c(0.1, 0.05))
sel.ngroups <- predict(object = fit.PCLasso, type="ngroups",
                          lambda = c(0.1, 0.05))
sel.vars.unique <- predict(object = fit.PCLasso, type="vars.unique",
                             lambda = c(0.1, 0.05))
sel.nvars.unique <- predict(object = fit.PCLasso, type="nvars.unique",
                              lambda = c(0.1, 0.05))
sel.vars <- predict(object = fit.PCLasso, type="vars",
                      lambda=c(0.1, 0.05))
sel.nvars <- predict(object = fit.PCLasso, type="nvars",
                       lambda=c(0.1, 0.05))
```

---

predict.PCLasso2   *Make predictions from a PCLasso2 model*

Description

Similar to other predict methods, this function returns predictions from a fitted `PCLasso2` object.

Usage

```
## S3 method for class 'PCLasso2'
predict(
  object,
  x = NULL,
  type = c("link", "response", "class", "norm", "coefficients", "vars", "nvars",
    "vars.unique", "nvars.unique", "groups", "ngroups"),
  lambda,
  ...
)
```

Arguments

`object`	Fitted PCLasso2 model object.
`x`	Matrix of values at which predictions are to be made. The features (genes/proteins) contained in x should be consistent with those contained in x in the PCLasso2 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)); "class" returns the binomial outcome with the highest probability; "vars" returns the indices for the nonzero coefficients; "vars.unique" returns unique features (genes/proteins) 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/proteins) 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.
`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.grpreg` in the R package `grpreg`.

Details

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

Value

The object returned depends on `type`.

See Also

[PCLasso2](#)

Examples

```
# load data
data(classData)
data(PCGroups)

x <- classData$Exp
y <- classData$Label
PC.Human <- getPCGroups(Groups = PCGroups, Organism = "Human",
Type = "GeneSymbol")

set.seed(20150122)
idx.train <- sample(nrow(x), round(nrow(x)*2/3))
x.train <- x[idx.train,]
y.train <- y[idx.train]
```

```
x.test <- x[-idx.train,]
y.test <- y[-idx.train]

# fit PCLasso2 model
fit.PCLasso2 <- PCLasso2(x = x.train, y = y.train, group = PC.Human,
                  penalty = "grLasso", family = "binomial")

# predict risk scores of samples in x.test
s <- predict(object = fit.PCLasso2, x = x.test, type="link",
lambda=fit.PCLasso2$fit$lambda)

# predict classes of samples in x.test
s <- predict(object = fit.PCLasso2, x = x.test, type="class",
lambda=fit.PCLasso2$fit$lambda[10])

# Nonzero coefficients
sel.groups <- predict(object = fit.PCLasso2, type="groups",
                      lambda = fit.PCLasso2$fit$lambda)
sel.ngroups <- predict(object = fit.PCLasso2, type="ngroups",
                       lambda = fit.PCLasso2$fit$lambda)
sel.vars.unique <- predict(object = fit.PCLasso2, type="vars.unique",
                           lambda = fit.PCLasso2$fit$lambda)
sel.nvars.unique <- predict(object = fit.PCLasso2, type="nvars.unique",
                            lambda = fit.PCLasso2$fit$lambda)
sel.vars <- predict(object = fit.PCLasso2, type="vars",
                    lambda=fit.PCLasso2$fit$lambda)
sel.nvars <- predict(object = fit.PCLasso2, type="nvars",
                     lambda=fit.PCLasso2$fit$lambda)

# For values of lambda not in the sequence of fitted models,
# linear interpolation is used.
sel.groups <- predict(object = fit.PCLasso2, type="groups",
                      lambda = c(0.1, 0.05))
sel.ngroups <- predict(object = fit.PCLasso2, type="ngroups",
                       lambda = c(0.1, 0.05))
sel.vars.unique <- predict(object = fit.PCLasso2, type="vars.unique",
                           lambda = c(0.1, 0.05))
sel.nvars.unique <- predict(object = fit.PCLasso2, type="nvars.unique",
                            lambda = c(0.1, 0.05))
sel.vars <- predict(object = fit.PCLasso2, type="vars",
                    lambda=c(0.1, 0.05))
sel.nvars <- predict(object = fit.PCLasso2, type="nvars",
                     lambda=c(0.1, 0.05))
```

---

survivalData   *A dataset for prognostic model*

Description

A dataset for prognostic model

Usage

```
survivalData
```

Format

A list containing a protein expression matrix and survival data

Exp a protein expression matrix

survData Survival data. 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.

Index