Package 'plrs'

January 28, 2020

Version 1.27.0

Type Package

Title Piecewise Linear Regression Splines (PLRS) for the association between DNA copy number and gene expression

Author Gwenael G.R. Leday

Maintainer Gwenael G.R. Leday to <gleday@few.vu.nl>

Depends R (>= 2.10), Biobase

Imports BiocGenerics, CGHbase, graphics, grDevices, ic.infer, marray, methods, quadprog, Rcsdp, stats, stats4, utils

Suggests mvtnorm, methods

Description The present package implements a flexible framework for modeling the relationship between DNA copy number and gene expression data using Piecewise Linear Regression Splines (PLRS).

License GPL (>=2.0)

biocViews Regression

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

git_branch master

git_last_commit b1a6cb4

git_last_commit_date 2019-10-29

Date/Publication 2020-01-27

R topics documented:

`plrs-package` *Piecewise Linear Regression Splines (PLRS) for the association between DNA copy number and mRNA expression*

Description

The present package implements a framework for modeling the relationship between DNA copy number and gene expression data using Piecewise Linear Regression Splines (PLRS). It includes (point and interval) estimation, model selection and testing procedures for such models (possibly under biologically motivated constraints).

Details

The use of the present package can be divided into two approaches:

1. Analysis of a single DNA-mRNA relationship

Main functions are:
`plrs`: Fit a single plrs model.
`plrs.select`: Model selection based on AIC, AICC, OSAIC or BIC.
`plrs.test`: Likelihood ratio test for a given plrs model.
`plrs.cb`: Confidence bands for a plrs model.

2. Analysis of multiple DNA-mRNA relationships sequentially

Main function is:
`plrs.series`: point and interval estimation, model selection and testing of DNA-mRNA association for a series of arrays.

Note: This function extend the aforementioned univariate analysis genomewise in the same spirit as some functions of the **limma** package do.

Author(s)

Gwenael G.R. Leday

Maintainer: Gwenael G.R. Leday <g.g.r.leday@vu.nl>

References

Leday GGR, Van der Vaart AW, Van Wieringen WN, Van de Wiel MA. Modeling association between DNA copy number and gene expression with constrained piecewise linear regression splines. Accepted for publication. *Ann Appl Stat*. (2012).

`criteria` *Compute AIC, AICC, BIC and OSAIC for a given* `plrs` *model.*

Description

Extract AIC, AICC, BIC and OSAIC from an object of class `plrs-class`.

Usage

```
criteria(obj, crit = "all")
```

Arguments

`obj`	object of class `plrs-class`
`crit`	A `character` (vector) among `"aic"`, `"aicc"`, `"bic"`, `"osaic"` or `"all"`.

Value

A `list` with the following components (if specified):

`aic`	Akaike's information criterion
`aicc`	Small sample correction of AIC
`bic`	Bayesian Information Criterion
`osaic`	One-Sided AIC. See Hughes and King (2003) for more details.

Author(s)

Gwenael G.R. Leday <g.g.r.leday@vu.nl>

References

Hughes, A. W. and King, M. L. (2003). Model selection using AIC in the presence of one-sided information. *J Stat Plan Infer*, 115(2): 397 411.

Examples

```
# Simulate data
sim <- plrs.sim(n=80, states=4, sigma=0.5)

# Fit
model <- plrs(expr=sim$expr, cghseg=sim$seg, cghcall=sim$cal)

criteria(model)
```

`modify.conf` *Modify the configuration (of calls) of the plrs model*

Description

This function changes the discrete copy number values for a given gene in order to force a minimum number of observations per state.

Usage

```
modify.conf(cghcall, min.obs = 3, discard = TRUE)
```

Arguments

`cghcall`	Vector of called values
`min.obs`	Minimum number of observations per state
`discard`	Logical. Whether discrete states with few observations should be discarded from analysis.

Details

Consider that the number of observations of a given state is lower than `min.obs`, then:

- if `discard = FALSE`, observations are not discarded and a rearrangement of called values is carried out as follows. The "normal" copy number state is taken as a reference. If the minimum number of observations is not obtained, "losses" will be merged to "normals", "gains" to "normals" and "amplifications" to "gains". Note that this modifies the configuration of the model. Thus, after fitting a model using `plrs`, original and modified data are stored in the resulting `plrs-class` object, respectively under slots `data` and `mdata`.

- if `discard = TRUE`, states for which the number of observations is lower than `min.obs` are discarded (replaced by NAs).

Value

`val`	Vector of new called values

Note

This function is implemented within function `plrs` and `plrs.series`.

Author(s)

Gwenael G.R. Leday <g.g.r.leday@vu.nl>

Examples

```
called <- sample(c(rep(-1,5),rep(0,15),rep(1,2),rep(2,1)))
table(called)
table(modify.conf(called, min.obs=3))
```

neveCN17 *Copy number for chromosome 17.*

Description

Preprocessed copy number data of Neve et al. (2006) for chromosome 17.

Usage

```
neveCN17
```

Format

An object of class cghCall

Source

M. Neve et al. in Gray Lab at LBL. Neve2006: expression and CGH data on breast cancer cell lines. R package version 0.1.10.

References

Neve, R.M. et al. (2006). A collection of breast cancer cell lines for the study of functionally distinct cancer subtypes. *Cancer cell*, 10, 515-527.

Examples

```
data(neveCN17)
dim(neveCN17)
head(fData(neveCN17))
```

neveGE17 *mRNA expression for chromosome 17.*

Description

Normalized gene expression data of Neve et al. (2006) for chromosome 17.

Usage

```
neveGE17
```

Format

An object of class ExpressionSet

Source

M. Neve et al. in Gray Lab at LBL. Neve2006: expression and CGH data on breast cancer cell lines. R package version 0.1.10.

References

Neve, R.M. et al. (2006). A collection of breast cancer cell lines for the study of functionally distinct cancer subtypes. *Cancer cell*, 10, 515-527.

Examples

```
data(neveGE17)
dim(neveGE17)
head(fData(neveGE17))
```

`plot-methods` *Plot functions in package 'plrs'*

Description

Methods `plot` in package 'plrs'

Usage

```
## S3 method for class 'plrs'
plot(x, col.line = "black", col.pts = c("red", "blue","green2", "green4"),
col.cb = "yellow", xlim = c(floor(min(x@data$cghseg)),ceiling(max(x@data$cghseg))),
ylim = c(floor(min(x@data$expr)),ceiling(max(x@data$expr))),
pch = 16, lwd=4, cex = 1.2, xlab="", ylab="", main = "",
add = FALSE, lty = 1, lin = FALSE, ...)
```

Arguments

`x`	An object of class `plrs-class` or `plrs.select-class`
`col.line`	Color of the fitted line
`col.pts`	Vector of length 4, for colors associated with each state
`col.cb`	Color for the confidence band
`xlim`	The x limits of the plot
`ylim`	The y limits of the plot
`pch`	See `par`
`lwd`	See `par`
`cex`	See `par`
`xlab`	Title of the x-axis
`ylab`	Title of the y-axis
`main`	Main title for the plot
`add`	If the plot should be added to the current device. Default is `FALSE`
`lty`	See `par`
`lin`	Logical. Whether the simple linear model should also be plotted
`...`	Other arguments, see `par`

Details

`plot.plrs` plots the observed points, the fitted line and potentially the confidence band.

Methods

`signature(x = "plrs")` Plot observed points and the fitted line

`signature(x = "plrs.select")` Plot observed points and the fitted line of the selected model.

Author(s)

Gwenael G.R. Leday <g.g.r.leday@vu.nl>

---

plrs *Fit a (constrained) piecewise linear regression spline*

Description

The function fits a piecewise linear regression spline to explain gene expression by the segmented DNA copy number. The called copy number values are used as a template for model building.

Usage

```
plrs(expr, cghseg, cghcall=NULL, probloss = NULL, probnorm = NULL,
probgain = NULL, probamp = NULL, knots = NULL, continuous = FALSE,
constr = TRUE, constr.slopes = 2, constr.intercepts = TRUE,
min.obs = 3, discard.obs = TRUE)
```

Arguments

`expr`	Vector of gene expression values
`cghseg`	Vector of segmented copy number values
`cghcall`	Vector of called copy number values. If not provided, we are reduced to a simple linear model.
`probloss`	Vector of call probabilities associated with state "loss". Default is `NULL`.
`probnorm`	Vector of call probabilities associated with state "normal". Default is `NULL`.
`probgain`	Vector of call probabilities associated with state "gain". Default is `NULL`.
`probamp`	Vector of call probabilities associated with state "amplification". Default is `NULL`.
`knots`	knots or change points. If `NULL` (default), there are estimated. See details.
`continuous`	Logical, whether the model is continuous (no jump) or not.
`constr`	Logical, whether the model is constrained or not. (this has been implemented to turn on and off easily the constraints)
`constr.slopes`	Type of non-negativity constraints applied on slopes. Either 1 or 2 (default). See details.
`constr.intercepts`	If `TRUE` (default) jumps from state to state are also constrained to be non-negative
`min.obs`	See `modify.conf`
`discard.obs`	See `modify.conf`

Details

If `cghcall=NULL`, discrete copy number values are omitted, which results in fitting a simple linear model.

If `constr.slopes=1`, all slopes are constrained to be non-negative. If `constr.slopes=2`, the slope associated with state "normal" is constrained to be non-negative and all others are forced to be at least equal to the latter.

Two methods are implemented for the estimation of knots. If call probabilities are provided, a knot is determined so that the sum of (the two adjacent) states membership probabilities is maximized. Otherwise, this is defined as the midpoint of the interval between the two consecutive states.

The constrained least squares problem is solved using function `solve.QP` of package **quadprog**.

Value

An object of class `plrs-class`

Author(s)

Gwenael G.R. Leday <g.g.r.leday@vu.nl>

Examples

```
# Simulate data
sim <- plrs.sim(n=80, states=4, sigma=0.5)

# Fit a model
model <- plrs(expr=sim$expr, cghseg=sim$seg, cghcall=sim$cal)
model

# Methods
coef(model)
effects(model)
fitted(model)
knots(model)
model.matrix(model)
plot(model)
predict(model, newcghseg=seq(0,5, length.out=100))
residuals(model)
summary(model)
```

---

plrs-class *Class* plrs

Description

An S4 class representing the output of the `plrs` function.

Slots

`coefficients`: Object of class `numeric` containing spline coefficients

`fitted.values`: Object of class `numeric` containing the fitted values

`residuals`: Object of class `numeric` containing the residuals

`X`: Object of class `matrix` containing the design matrix

`data`: Object of class `list` containing input data

`mdata`: Object of class `list` containing (possibly modified) data used to fit the model (See `modify.conf`).

`QP`: Object of class `list` containing input elements used for quadratic programming. If the model is unconstrained this contains a light version of an `lm` object.

`test`: Object of class `list` containing results from testing.

`cb`: Object of class `list` containing lower and upper bounds for predicted values.

`selected`: Object of class `logical` indicating whether the model results from a selection procedure.

`type`: Object of class `character` giving the type of model

`call.arg`: Object of class `list` containing the input arguments (for reproducibility)

Methods

coef Returns the coefficients

criteria See `criteria`

effects Returns matrix of effects

fitted Returns the fitted values

knots Returns the knots

model.matrix Returns the design matrix

plot See `plot.plrs`

predict See `predict.plrs`

print Print the object information

residuals Returns the residuals

show Print the object information

summary Print a summary of the object information

Author(s)

Gwenael G.R. Leday <g.g.r.leday@vu.nl>

plrs.cb *Uniform confidence bands (CB) for plrs models*

Description

Determine uniform confidence intervals for predicted values of a 'plrs' model.

Usage

```
plrs.cb(object, alpha=0.05, newcgh=NULL)
```

Arguments

object	An object of class plrs-class.
alpha	Significance level
newcgh	Vector of segmented values. Support for building CB.

Details

The input object of class plrs-class has to result from function plrs.test.

The problem of finding (at a given x) a confidence interval for the mean response is expressed as a semi-definite optimization problem and solved using function csdp of package **Rcsdp**.

Value

An object of class plrs-class that contains CB information.

Author(s)

Gwenael G.R. Leday <g.g.r.leday@vu.nl>

References

Leday GGR, Van der Vaart AW, Van Wieringen WN, Van de Wiel MA. Modeling association between DNA copy number and gene expression with constrained piecewise linear regression splines. Accepted for publication. *Ann Appl Stat*. (2012).

See Also

plrs.test

Examples

```
# Simulate data
sim <- plrs.sim(n=80, states=4, sigma=0.5)

# Fit a model
model <- plrs(expr=sim$expr, cghseg=sim$seg, cghcall=sim$cal)

# Confidence bands
```

```
model <- plrs.test(model)
model <- plrs.cb(model, alpha=0.05)
plot(model)
```

`plrs.select` *Model selection*

Description

Selection of a model based on an information criterion (AIC, AICC, BIC or OSAIC).

Usage

```
plrs.select(object, crit = ifelse(object@call.arg$constr,"osaic","aic"))
```

Arguments

`object`	An object of class `plrs-class`
`crit`	Character corresponding to the criterion to use. See `criteria`.

Value

An object of class `plrs.select-class`

Author(s)

Gwenael G.R. Leday <g.g.r.leday@vu.nl>

`plrs.select-class` *Class* `plrs.select`

Description

An S4 class representing the output of the `plrs.select` function.

Slots

`table`: Object of class `matrix` containing the criterion value for all models

`model`: Object of class `plrs` containing the selected model

`crit`: Object of class `character` containing the criterion used for model selection

Methods

plot See `plot.plrs`

print Print the object information

show Print the object information

summary Print a summary of the object information

Author(s)

Gwenael G.R. Leday <g.g.r.leday@vu.nl>

plrs.series *Fit plrs models for a series of arrays.*

Description

The function fits plrs models for a series of arrays. Model selection and testing procedures may be applied.

Usage

```
plrs.series(expr, cghseg, cghcall=NULL,
probloss = NULL, probnorm = NULL, probgain = NULL, probamp = NULL,
control.model  = list(continuous = FALSE,
                      constr = TRUE,
                      constr.slopes = 2,
                      constr.intercepts = TRUE,
                      min.obs = 3,
                      discard.obs = TRUE),
control.select = list(crit = ifelse(control.model$constr, "osaic","aic")),
control.test   = list(testing = TRUE,
                      cb = FALSE,
                      alpha = 0.05),
control.output = list(save.models = FALSE,
                      save.plots = FALSE,
                      plot.lin = FALSE,
                      type = "jpeg"))
```

Arguments

expr	Either a matrix of expression profiles or an ExpressionSet object.
cghseg	Either a matrix of segmented copy number values or objects of class cghSeg or cghCall
cghcall	Matrix of called copy number
probloss	Matrix of call probabilities associated with state "loss". Default is NULL.
probnorm	Matrix of call probabilities associated with state "normal". Default is NULL.
probgain	Matrix of call probabilities associated with state "gain". Default is NULL.
probamp	Matrix of call probabilities associated with state "amplification". Default is NULL.
control.model	See details
control.select	See details
control.test	See details
control.output	See details

Details

If DNA and mRNA input data are matrices, rows should correspond to genes and columns to arrays. Alternatively, expression data may be provided as an ExpressionSet object and aCGH data as cghSeg or cghCall objects. A cghCall object contain all data from the calling step, thus arguments probloss, probnorm, probnorm and probamp can be omitted. An object of class cghSeg does not contain such data so only simple linear models will be fitted.

control.model allows the user to specify the type of model that has to be fitted. This must be a list with one or more of the following components: constr, constr.slopes, constr.intercepts, min.obs and discard.obs. See functions plrs and modify.conf for more details.

control.select allows the user to specify whether model selection should be done and how. This must be a list with a component named crit. See function plrs.select for more details. If control.select = NULL then no model selection is done.

control.output allows the user to plot and save each plrs model. This must be a list with components:
save.models, a logical. This will create within the work directory a new directory named "plrsSeriesObjects" that will contain all objects.
save.plots, a logical. This will create within the work directory a new directory named "plrsSeriesPlots" that will contains all saved plots.
plot.lin, a logical. Whether the simple linear model should aslo be plotted.
type, a character. Format of file. To pass through function savePlot.

Value

An object of class plrs.series-class

Author(s)

Gwenael G.R. Leday <g.g.r.leday@vu.nl>

Examples

```
# Simulate data
ngenes <- 10
narray <- 48
rna <- dnaseg <- dnacal <- matrix(NA, ngenes, narray)
idx <- sample(1:4, ngenes, replace=TRUE, prob=rep(1/4,4))
for(i in 1:ngenes){
Sim <- plrs.sim(n=narray, states=idx[i], sigma=0.5)
rna[i,] <- Sim$expr
dnaseg[i,] <- Sim$seg
dnacal[i,] <- Sim$cal
}


# Screening procedure with linear model
series <- plrs.series(expr = rna, cghseg = dnaseg, cghcall = NULL, control.select = NULL)

# Screening procedure with full plrs model
series <- plrs.series(expr = rna, cghseg = dnaseg, cghcall = dnacal, control.select = NULL)
```

```
# Model selection
series <- plrs.series(expr = rna, cghseg = dnaseg, cghcall = dnacal)
```

`plrs.series-class` *Class* `plrs.series`

Description

An S4 class representing the output of the `plrs.series` function.

Slots

`coefficients`**:** Matrix containing coefficients of models

`effects`**:** List containing effects

`test`**:** Matrix containing results from testing.

`general`**:** Matrix providing the distribution of the number genes and arrays regarding the copy number states

`modelsType`**:** List providing models' type

`call.arg`**:** List providing details on the type of models that have been fitted.

Methods

print Print the object information

show Print the object information

summary Print a summary of the object information

Author(s)

Gwenael G.R. Leday <g.g.r.leday@vu.nl>

`plrs.sim` *Simulation of a plrs model*

Description

Simulation of a piecewise relationship.

The function has been only implemented for convenience of simulations and R examples.

Usage

```
plrs.sim(n = 80, states = 4, sigma = 01, x = NULL)
```

Arguments

n	Number of simulated data points
states	Number of states for the model
sigma	Noise
x	Segmented values.

Details

To be written...

Author(s)

Gwenael G.R. Leday <g.g.r.leday@vu.nl>

Examples

```
# Simulate 1-state model
sim <- plrs.sim(n=80, states=1, sigma=0.5)
model <- plrs(expr=sim$expr, cghseg=sim$seg, cghcall=sim$cal)
plot(model)

# Simulate 2-state model
sim <- plrs.sim(n=80, states=2, sigma=0.5)
model <- plrs(expr=sim$expr, cghseg=sim$seg, cghcall=sim$cal)
plot(model)

# Simulate 3-state model
sim <- plrs.sim(n=90, states=3, sigma=0.5)
model <- plrs(expr=sim$expr, cghseg=sim$seg, cghcall=sim$cal)
plot(model)

# Simulate 4-state model
sim <- plrs.sim(n=80, states=4, sigma=0.5)
model <- plrs(expr=sim$expr, cghseg=sim$seg, cghcall=sim$cal)
plot(model)
```

---

plrs.test *Likelihood ratio test for a* plrs *model*

Description

Test whether copy number has an effect on mRNA expression.

Usage

```
plrs.test(object, alpha=0.05)
```

Arguments

object	An object of class plrs-class
alpha	Significance level

Details

Two cases present themselves:

1. The model is unconstrained. Thus, the model under the null hypothesis is the intercept and an F-test is performed.

2. The model is constrained and the following hypothesis are tested:
H0: All constraints are actives (=)
H1: At least one constraint is strict (>)
Under H0, we always have the intercept model. Indeed, if `constr.slopes` = 1 (or 2) and `constr.intercepts` = `T`, then the only parameter free of inequality constraint is the overall intercept. If `constr.intercepts` = `F`, the local intercepts are additionnaly constrained to be 0 in order to obtain the intercept model under the null. The likelihood ratio statistic (unknown variance) is asymptotically distributed as a weighted mixture of Beta distribution (cf Gromping (2010)). Calculation of p-values is based on functions `ic.weights` and `pbetabar` of package **ic.infer**. The package **mvtnorm** is also involved.

In both cases the input model is taken as the model under the alternative.

Value

A `list` object with the following components:

`stat`	Test statistic
`pvalue`	Calculated pvalue
`wt.bar`	Weights (if the model is constrained)
`df.bar`	Degrees of freedom.
`unconstr`	Unconstrained model of class `plrs-class`
`qbetabar`	(1-`alpha`) quantile of the beta mixture distribution
`alpha`	Significance level

Author(s)

Gwenael G.R. Leday <g.g.r.leday@vu.nl>

References

Gromping, U. (2010). Inference with linear equality and inequality constraints using R: The package ic.infer. *J Stat Softw*, 33(i10).

Examples

```
# Simulate data
sim <- plrs.sim(n=80, states=2, sigma=0.5)

# Fit a model
model <- plrs(expr=sim$expr, cghseg=sim$seg, cghcall=sim$cal)

# Testing
model <- plrs.test(model)
model
```

predict.plrs *Predict method for* plrs *models*

Description

Determine predicted values based on a given plrs model

Usage

```
## S3 method for class 'plrs'
predict(object, newcghseg, ...)
```

Arguments

object	An object of class plrs-class
newcghseg	A vector of new segmented CGH values
...	further arguments

Value

A vector containing the fitted values.

Author(s)

Gwenael G.R. Leday <g.g.r.leday@vu.nl>

Index