Package ‘bhm’

May 11, 2017

Type Package
Title Biomarker Threshold Models
Version 1.11
Date 2017-05-06
Author Bingshu E. Chen
Maintainer Bingshu E. Chen <bingshu.chen@queensu.ca>
Depends R (>= 2.10.0), coda, survival
Imports methods
Description Biomarker threshold models are tools to fit both predictive and prognostic biomarker effects.
License GPL (>= 2)
LazyLoad yes
NeedsCompilation no
Repository CRAN
Date/Publication 2017-05-11 16:28:14 UTC

R topics documented:

bhm-package .................................................. 2
bhm ............................................................ 3
control ....................................................... 5
data ............................................................ 7
pIndex ......................................................... 8
pIndexControl ............................................... 10
print .......................................................... 11

Index 13
**Description**

This package fits biomarker threshold regression models for predictive and prognostic biomarker effects with binary data and survival data with an unknown biomarker cutoff point. Multivariable models can also be fitted for adjusted biomarker effect. Tools such as Probability index are included to measure treatment effect, biomarker effect or treatment-biomarker interaction.

**Details**

"bhm" is a R package for Biomarker Threshold Models. Please use the following steps to install 'bhm' package:

1. First, you need to install the 'devtools' package. You can skip this step if you have 'devtools' installed in your R. Invoke R and then type
   
   ```
   install.packages("devtools")
   ```

2. Load the devtools package.
   
   ```
   library(devtools)
   ```

3. Install "bhm" package with R command
   
   ```
   install_github("statapps/bhm")
   ```

"bhm" uses different statistical methods to identify cut-point (threshold parameter) for the biomarker in either generalized linear models or Cox proportional hazards model.

**Author(s)**

Bingshu E. Chen

Maintainer: Bingshu E. Chen <bingshu.chen@queensu.ca>

**References**


**See Also**

coxph, glm, survival
**bhm**

**Fitting Biomarker Threshold Models**

**Description**

*bhm* is a R package for Biomarker Threshold Models. It uses either Hierarchical Bayes method or profile likelihood method (Chen, et al, 2014 and Tian, et al, 2017) to identify a cut-point (threshold parameter) for the biomarker in either generalized linear models or Cox proportional hazards model. The model is specified by giving a symbolic description of the linear predictor and a description of the distribution family.

**Usage**

```r
bhm(x, ...)  
```

```r
## S3 method for class 'formula'
```

```r
bhm(formula, family, data, control = list(...), ...)
```

```r
# use
#    bhm(y ~ biomarker)
#
# to fit a prog nostic model with biomarker term only
#    bhm(y ~ biomarker+treatment)
#
# to fit a predictive model with interaction between biomarker
# and treatment, use
#    bhmFit(x, y, family, control)
#
# to fit a model without the formula
# Biomarker shall be in the first dependent variable
```

**Arguments**

- `formula`: an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted. The details of model specification are given under 'Details'.
- `family`: a description of the response distribution and link function to be used in the model. The available family function are either "binomial" for fitting a logistic regression model or "surv" for fitting a Cox proportional hazards model.
data

an optional data frame, list or environment (or object coercible by 'as.data.frame' to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which glm is called.

x, y

For 'bhmFit', x is a design matrix of dimension n * p and y is a vector of observations of length n for "glm" models or a "Surv" survival object for "coxph" models.

control

a list of parameters for controlling the fitting process. See 'bhmControl' for details

... additional arguments to be passed to the low level regression fitting functions (see below).

Details

'biomarker' is a Biomarker variable. This variable is required and shall be the first dependent variable in the formula.

'interaction' is an option of fitting model with itneractin term. When interaction = TRUE, a predictive biomarker model will be fitted. When interaction = FALSE, a prognostic biomarker model will be fitted. Both Biomarker and Treatment variables are required if 'interaction' = TRUE and 'treatment' shall be the second variable in the formula.

'bhmFit' and 'bhmGibbs' are the workhorse functions: they are not normally called directly but can be more efficient where the response vector, design matrix and family have already been calculated.

'x.cdf' is a function that maps biomarker values to interval (0, 1) using its empirical cumulative distribution function. After the threshold parameters are identified, the biomarker variable will be transformed back to its original scale.

Value

bhm returns an object of class inheriting from "bhm" which inherits from the class glm or 'coxph'. See later in this section.

The function "summary" (i.e., "summary.bhm") can be used to obtain or print a summary of the results, for example, the 95%

An object of class "bhm" is a list containing at least the following components:

c.max a vector of the mean estimates for the threshold parameter(s)
coefficients a named vector of coefficients from 'bhm'
c.fit fitted conditional regression model given c = c.max
cg Gibbs sample for threshold parmeter c
bg Gibbs sample for the coefficients beta

Note

The logistic regression part are based on codes wrote by Tian Fang.
control  

**Author(s)**

Bingshu E. Chen (bingshu.chen@queensu.ca)

**References**


**See Also**

`glm`, `coxph`, `bhmControl`

**Examples**

```r
## Generate a random data set
n = 300
b = c(0.5, 1, 1.5)
data = surv.gendat(n, c0 = 0.40, beta = b)
age = runif(n, 0, 1)*100
tm = data[, 1]
status = data[, 2]
trt = data[, 3]
ki67 = data[, 4]

## fit a biomarker threshold survival model with one single cut point
fit = bhm(Surv(tm, status)~ki67+trt+age, interaction = TRUE, B=5, R=10)

## here B=5 and R=10 is used for test run. In general, B > 500 and R > 2000 is
## recommend for the analysis of biomarker variable. To fit a model with
## two cut points, use:
##
##    fit = bhm(Surv(tm, status)~bmk+trt+age, B = 500, R = 2000, c.n = 2)
##
## To print the output, use
##
##    print(fit)
```

---

**Description**

Auxiliary function for `bhm` fitting. Typically only used internally by 'bhmFit', but may be used to construct a control argument to either function.
Usage

```r
bhmControl(method = 'Bayes', interaction, biomarker.main, alpha, B, R, thin, epsilon, c.n, beta0, sigma0)
```

Arguments

- `method`: choose either 'Bayes' for Bayes method with MCMC or 'profile' for profile likelihood method with Bootstrap. The default value is 'Bayes'.
- `interaction`: an option of fitting model with interaction term. When `interaction = TRUE`, a predictive biomarker model will be fitted. When `interaction = FALSE`, a prognostic biomarker model will be fitted. The default value is `interaction = TRUE`.
- `biomarker.main`: include biomarker main effect, default is `TRUE`.
- `B`: number of burn in.
- `R`: number of replications for Bayes method or number of Bootstrap for profile likelihood method.
- `thin`: thinning parameter for Gibbs samples, default is 2.
- `epsilon`: step length for profile likelihood method, default is 0.01.
- `alpha`: significance level (e.g. `alpha=0.05`).
- `c.n`: number of threshold (i.e. the cut point), default is 1.
- `beta0`: initial value for mean of the prior distribution of beta, default is 0.
- `sigma0`: initial value for variance of the prior distribution of beta, default is 10000.

Details

Control is used in model fitting of `bhm`.

Value

This function checks the internal consistency and returns a list of value as inputed to control model fit of `bhm`.

Note

Based on code from Tian Fang.

Author(s)

Bingshu E. Chen and Tian Fang

See Also

- `bhm`
Examples

```r
## To fit a prognostic model for biomarker with two cut-points, 
## 500 burn-in samples and 10000 Gibbs samples,

cntl = bhmControl(interaction = FALSE, B = 500, R = 10000, c.n = 2)

## then fit the following model
##
## # fit = bhmFit(x, y, family = 'surv', control = ctl)
##
```

Description

dataset for biomarker threshold model (bhm)

Usage

- To generate survival data, use:
  ```r
  surv.gendat(n, c0, beta)
  ```

- To generate glm data, use:
  ```r
  glm.gendat(n, c0, beta)
  ```

Arguments

- `n` sample size
- `c0` cut off point, for example `c0 = 0.4`
- `beta` regression coefficient, for example, `beta = c(0.3, log(0.5), log(0.25))`

Format

The format of the data set for analysis shall be a data frame with a response variable (either a Surv object for Cox model or a glm response variable object) and at least one dependent variable as the biomarker variable.

Details

data set of prostate cancer in the ’survival’ package is used as an example in paper by Chen, et al. (2014).
Source

prostate dataset can be loaded with 'library(survival)'.

References


Examples

```r
#data(data)
## maybe str(data); plot(data) ...
c0 = 0.4
b = c(-0.5, 1.5, 1.3)
data = surv.gendat(n=300, c0 = c0, beta = b)
```

<table>
<thead>
<tr>
<th>pIndex</th>
<th>Probability Index for Survival Time Difference</th>
</tr>
</thead>
</table>

Description

pIndex is a function to estimate and test difference of survival time among groups. It is defined as $p = \Pr(T_1 < T_2)$, where $T_1$ is survival time for subjects in group 1 and $T_2$ is survival time in group 2.

Usage

```r
pIndex(x, ...)
```

Arguments

- `formula` an object of class "formula"(or one that can be coerced to that class): a symbolic description of the model to be fitted. The details of model specification are given under 'Details'.
data an optional data frame, list or environment (or object coercible by `as.data.frame` to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which pIndex is called.

x Here covariate x is a design matrix of dimension n * 1 (for two sample test) or dimension n * 2 (for treatment * biomarker interaction).

control a list of parameters for controlling the fitting process. See 'pIndexControl' for details

... additional arguments to be passed to the low level regression fitting functions (see below).

Details

pIndex(y~x) will estimate probability index of two groups (e.g. treatment vs control) define by x. pIndex(y~x1 + x2) will estimate the difference of probability index of x1 (e.g. treatment vs control) between biomarker positive and biomarker negative groups (x2). Function print(x) can be used to print a summary of pIndex results.

Value

pIndex returns an object of class inheriting from "pIndex". See later in this section.

Note

This function is part of the bhm package.

Author(s)

Bingshu E. Chen (bingshu.chen@queensu.ca)

References


See Also

bhm, pIndexControl

Examples

```r
## Generate a random data set
n = 300
b = c(0.5, 1, 1.5)
data = surv.gendat(n, c0 = 0.40, beta = b)
age = runif(n, 0, 1)*100
tm = data[, 1]
status = data[, 2]
```
trt = data[, 3]
ki67 = data[, 4]
#
### No run
#
# fit = pIndex(Surv(tm, status) ~ trt + ki67)
#

<table>
<thead>
<tr>
<th>pIndexControl</th>
<th>Auxiliary function for pIndex fitting</th>
</tr>
</thead>
</table>

**Description**

Auxiliary function for `pIndex` fitting. Typically only used internally by `pIndexFit`, but may be used to construct a control argument to either function.

**Usage**

```r
pIndexControl(method = c("Efron", "Elc", "Elw"), ci = c("Bootstrap", "Jackknife"),
               alpha = 0.05, B = 0, pct = 0.5)
```

**Arguments**

- `method` : choose either ‘Efron’ for Efron method, ‘Elc’ for conditional empirical likelihood, or ‘Elw’ for weighted empirical likelihood method. The default value is ‘Efron’
- `ci` : Method to construct confidence interval, ‘Bootstrap’ for Bootstrap method and ‘Jackknife’ for Jackknife method
- `B` : number of Bootstrap sample
- `alpha` : significance level (e.g. alpha=0.05)
- `pct` : Percentile of threshold (i.e. the cut point), default is 0.5

**Details**

Control is used in model fitting of `pIndex`.

**Value**

This function checks the internal consistency and returns a list of value as inputed to control model fit of `pIndex`.

**Note**

Based on code from Bingshu E. Chen.

**Author(s)**

Bingshu E. Chen
See Also

`bhm`, `pIndex`

Examples

```r
## To calculate the probability index for a biomarker with conditional empirical likelihood method,
## and the corresponding 90 percent CI using Bootstrap method with 10000 bootstrap sample

ctl = pIndexControl(method = 'Elc', ci = 'Bootstrap', B = 10000, alpha = .1)

##
## then fit the following model
##
## fit = pIndex(y~x1 + x2, family = 'surv', control = ctl)
##
##
## print
##
## print a bhm object or a summary of bhm object

Description

print and summary are used to provide a short summary of bhm outputs.

Usage

```r
## S3 method for class 'bhm'
print(x, ...)
## S3 method for class 'summary.bhm'
print(x, ...)
## S3 method for class 'bhm'
summary(object, ...)
```

Arguments

- `x` a bhm class returned from bhm fit
- `...` other options used in print()
- `object` object returned from bhm fit

Details

print.bhm is called to print object or summary of object from the biomarker threshold model `bhm`. summary(fit) provides detail summary of ‘bhm’ model fit, including parameter estimates, standard errors, and 95%

The default method, print.default has its own help page. Use methods("print") to get all the methods for the print generic.
Author(s)

Bingshu E. Chen and Fang Tian

See Also

The default method for print `print.default.glm.bhm`

Examples

```r
#
# print(fit)
#
```
# Index

**Biomarker interaction**
- bhm, 3

**Predictive effect**
- bhm, 3

**Prognostic effect**
- bhm, 3

**bhm**
- bhm-package, 2

**control**
- control, 5
- pIndexControl, 10

**datasets**
- data, 7

**print**
- print, 11

**summary**
- print, 11

bhm, 3, 5, 6, 9, 11, 12
bhm-doc (bhm-package), 2
bhm-package, 2
bhmControl, 5
bhmControl (control), 5
bhmFit (bhm), 3
bhmGibbs (bhm), 3
control, 5
coxph, 5
data, 7
glm, 5, 12
glm.gendat (data), 7
pIndex, 8, 10, 11
pIndexControl, 9, 10
print, 11
print.default, 12
print.pIndex (pIndex), 8
prolikControl (control), 5
prolikFit (bhm), 3

summary.bhm (print), 11
surv.gendat (data), 7
thm.fit (bhm), 3
thm.lik (bhm), 3
x.cdf (bhm), 3