Package ‘biospear’

December 4, 2018

Type Package

Title Biomarker Selection in Penalized Regression Models

Version 1.0.2

Author Nils Ternes [aut],
Federico Rotolo [aut], Stefan Michiels [aut, cre]

Maintainer Stefan Michiels <stefan.michiels@gustaveroussy.fr>

Depends R (>= 2.10), pkgconfig


License GPL-2

Encoding UTF-8

LazyData true

Imports cobs, corpcor, devtools, glmnet, graphics, grplasso, MASS, Matrix, mboost, parallel, plsRcox, pROC, PRROC, RCurl, stats, survAUC, survival

NeedsCompilation no

Repository CRAN

Date/Publication 2018-12-04 10:20:19 UTC

R topics documented:

BMsel ................................................................. 2
Breast ................................................................. 6
expSurv .............................................................. 7
BMsel

Biomarker selection in a Cox regression model

Description

This function enables to fit a Cox regression model for a prognostic or a biomarker-by-treatment interaction setting subject to a selection procedure to perform variable selection.

Usage

BMsel(data, x, y, z, tt, inter, std.x = TRUE, std.i = FALSE, std.tt = TRUE,
method = c('lasso', 'lassoR', 'lassoU', 'enet', 'gboost',
'glasso', 'lasso', 'lasso-1se', 'lasso-AIC', 'lasso-BIC',
'lasso-HQIC', 'lasso-pct', 'lasso-pcvl', 'lasso-RIC', 'modCov',
'PCAlasso', 'PLSlasso', 'ridge', 'ridgelasso', 'stabSel', 'unifDOR'),
folds = 5, uni.fdr = 0.05, uni.test = 1, ss.rando = F, ss.nsub = 100,
ss.fsub = 0.5, ss.fwer = 1, ss.thr = 0.6, dfmax = ncol(data) + 1,
pct.rep = 1, pct.qtl = 0.95, showWarn = TRUE, trace = TRUE)

## S3 method for class 'resBMsel'
summary(object, show = TRUE, keep = c('tt', 'z', 'x', 'xt'),
add.ridge = FALSE, ...)

Arguments

data input data.frame. Each row is an observation.
x colnames or position of the biomarkers in data.
y colnames or position of the survival outcome in data. The first column must be the time and the second must be the indicator (0/1).
z colnames or position of the clinical covariates in data, if any.
nt colname or position of the treatment in data, if any.
inter logical parameter indicating if biomarker-by-treatment interactions should be computed.
std.x logical parameter indicating if the biomarkers should be standardized (i.e. subtracting by the mean and dividing by the standard deviation of each biomarker).
std.i logical parameter indicating if the biomarker-by-treatment interactions should be standardized (i.e. subtracting by the mean and dividing by the standard deviation of each interaction).
std.tt logical parameter indicating if the treatment should be recoded as +/-0.5.
The objects x, y, z (if any) and tt (if any) are mandatory for non-simulated data sets. The method parameter specifies the approaches for model selection. Most of these selection methods are based on the lasso penalty (Tibshirani, 1996). The tuning parameter is usually chosen though the cross-validated log-likelihood criterion (cvl), except for the empirical extensions of the lasso in which additional penalties to the cvl (given with a suffix, e.g. lasso-pcvl) are used to estimate the tuning parameter. Other methods based on the lasso are also implemented such as the adaptive lasso (alassol, alassor and alassou) for which the last letter indicates the procedure...
used to estimate the preliminary weights: "L" for lasso, "R" for ridge and "U" for univariate), the elastic-net (enet) or the stability selection (stabSel). For the interaction setting, specific methods were implemented: to reduce/control the main effects matrix (i.e. ridge (ridgelasso) or dimension reduction (PCAlasso or PLSlasso)), to select or discard main effects and interactions simultaneously (i.e. group-lasso (glasso)), or to include only the interaction part in the model (i.e. modCov). Some selection methods not based on penalized regression are also proposed: univariate selection (uniFDR), gradient boosting (gboost). The ridge penalty without selection can also be applied.

For all methods but the uniFDR, tuning parameters are chosen by maximizing the cross-validated log-likelihood (max-cvl). For the elastic-net, the "alpha" parameter (trade-off between ridge and lasso) is investigated among a predefined grid of values (as suggested by the authors, Zou et al. 2005) and the "lambda" is estimated by maximizing the above-mentioned cvl criterion for each of the "alpha" parameter. The combination (alpha; lambda) that maximizes the cvl is finally retained. For the gradient boosting, the number of steps is also estimated by the max-cvl. For the univariate selection, the tuning parameter is the FDR threshold defined by the user to control for multiple testing (using the parameter uni.fdr).

We have included the possibility to adjust for clinical covariates (z) for all methods. For penalized regressions, these covariates are considered as unpenalized. For the gradient boosting, a model with clinical covariates is preliminary implemented and regression coefficients are fixed as offset in the boosting approach. For the univariate selection, clinical covariates are forced as adjustment variables in the model and the FDR is calculated on the Wald p-values of the coefficient associated with the biomarker in such models.

Value

An object of class ‘resBMsel’ containing the list of the selected biomarkers and their estimated regression coefficients for the chosen methods.

Author(s)

Nils Ternes, Federico Rotolo, and Stefan Michiels
Maintainer: Nils Ternes <nils.ternes@yahoo.com>

References


Examples

# Simulated data set

```r
### # Simulated data set
```
## Low calculation time

```r
set.seed(654321)
sdata <- simdata(
    n = 500, p = 20, q.main = 3, q.inter = 0,
    prob.tt = 0.5, alpha.tt = 0,
    beta.main = -0.8,
    b.corr = 0.6, b.corr.by = 4,
    m0 = 5, wei.shape = 1, recr = 4, fu = 2,
    timefactor = 1)
resBM <- BMsel(
    data = sdata,
    method = c("lasso", "lasso-pcv1"),
    inter = FALSE,
    folds = 5)
summary(resBM)
```

## Not run:

## Moderate calculation time

```r
set.seed(123456)
sdata <- simdata(
    n = 500, p = 100, q.main = 5, q.inter = 5,
    prob.tt = 0.5, alpha.tt = -0.5,
    beta.main = c(-0.5, -0.2), beta.inter = c(-0.7, -0.4),
    b.corr = 0.6, b.corr.by = 10,
    m0 = 5, wei.shape = 1, recr = 4, fu = 2,
    timefactor = 1,
    active.inter = c("bm003", "bm021", "bm044", "bm049", "bm097"))
resBM <- BMsel(
    data = sdata,
    method = c("lasso", "lasso-pcv1"),
    inter = TRUE,
    folds = 5)
summary(resBM)
summary(resBM, keep = "xt")
```

## Not run:

## Breast cancer data set

```r
set.seed(123456)
resBM <- BMsel(
    data = Breast,
    x = 4:ncol(Breast),
```
Breast contains clinical and genomic data of 614 early breast cancer patients.

Usage

data(Breast)

Format

A dataframe with variables:

- **time** Distant-relapse free survival time (in years).
- **status** Distant-relapse free survival indicator (0 = censored, 1 = event).
- **treat** Treatment arm (Anthracycline-based adjuvant chemotherapy with (treat = +0.5) or without (treat = -0.5) taxane).

... All other covariates (p=1689) are gene expression values.

References


Examples

```r
library(survival)
data(Breast)
dim(Breast)

km <- survfit(Surv(time, status) ~ treat, data = Breast)
km
plot(km, col = c("black", "red"), lwd = 2, xaxt = "n", yaxt = "n")
legend("bottomleft", legend = c("Control", "Experimental"), col = 1:2,
      lty = 1, lwd = 2, cex = 1.5)
axis(1, cex.axis = 1.3)
axis(2, las = 2, cex.axis = 1.3)
mtext("Distant-recurrence free survival", side = 2, line = 3.2, cex = 1.5)
mtext("Time (in years)", side = 1, line = 2.5, cex = 1.5)
```

expSurv

**Computation of expected survival based on a prediction model**

Description

Based on a prediction model, this function computes expected survival for patients with associated confidence intervals. The returned object can be plotted to obtain a meaningful graphical visualization.

Usage

```r
expSurv(res, traindata, method, ci.level = .95, boot = FALSE, nboot, smooth = TRUE,
        pct.group = 4, time, trace = TRUE, ncores = 1)
```

## S3 method for class 'resexpsurv'
predict(object, newdata, ...)

## S3 method for class 'resexpsurv'
plot(x, method, pr.group, print.ci = TRUE,
      xlim, ylim, xlab, ylab, ...)

Arguments

- `res` an object of class `resBMsel` generated by `BMsel`.
- `traindata` the data.frame used to compute the `res` object (training set).
- `method` selection method to compute. If missing, all methods contained in `res` are computed.
- `ci.level` the nominal level for the two-sided confidence interval (CI) of the survival probability.
- `boot` logical value: TRUE = boostraped CI, FALSE = analytical CI.
- `nboot` number of bootstrap replicates (only used when boot=TRUE).
smooth logical value indicating if smoothed B-splines should be computed.
pct.group number or percentile of the prognostic-risk groups. If a single number is provided, all the groups must be defined according to Cox (1957). If percentiles are provided, the sum must be 1 (e.g. 0.164, 0.336, 0.336, 0.164).
time single time point to estimate the expected survival probabilities.
trace logical parameter indicating if messages should be printed.
ncores number of CPUs used (for the bootstrap CI).
object, x an object of class 'resexpsurv' generated by expSurv.
newdata data.frame containing new patients data. newdata must have the same variables as traindata.
pr.group parameter for the plot() indicating the number of the prognostic-risk group for which the plot will be printed.
print.ci logical parameter for the plot() indicating if CI will be printed.
xlim, ylim, xlab, ylab usual parameters for plot.
... other parameters for predict or plot.

Details

Using an object of class 'resBMsel' generated by BMsel, expSurv computes expected survival at a given time and constructs confidence intervals thereof either with an analytical (boot = FALSE) or non-parametric bootstrap approach (boot = TRUE). Smoothed B-splines (logical option smooth) and categorization of the prognostic score into risk groups (using the option pct.group) may be used to obtain a meaningful graphical visualization. Predictions for new patients (newdata data frame) can be computed using predict(). Graphical visualization can be obtained using plot().

Value

A list of length three containing the expected survival (surv) and their corresponding confidence intervals (lower and upper). Each element of the list contains a matrix of dimension number of patients x number of implemented methods.

Author(s)

Nils Ternes, Federico Rotolo, and Stefan Michiels
Maintainer: Nils Ternes <nils.ternes@yahoo.com>

Examples

```
# Simulated data set

# Low calculation time
set.seed(654321)
sdata <- simdata(
  n = 500, p = 20, q.main = 3, q.inter = 0,
```
prob.tt = 0.5, alpha.tt = 0,
beta.main = -0.8,
b.corr = 0.6, b.corr.by = 4,
m0 = 5, wei.shape = 1, recr = 4, fu = 2,
timefactor = 1)

resBM <- BMsel(
data = sdata,
method = c("lasso", "lasso-pcvl"),
inter = FALSE,
folds = 5)
esurv <- expSurv(
res = resBM,
traindata = sdata,
boot = FALSE,
time = 5,
trace = TRUE)
plot(esurv, method = "lasso-pcvl")

## Not run:
## Moderate calculation time
set.seed(123456)
sdata <- simdata(
n = 500, p = 100, q.main = 5, q.inter = 5,
prob.tt = 0.5, alpha.tt = -0.5,
beta.main = c(-0.5, -0.2), beta.inter = c(-0.7, -0.4),
b.corr = 0.6, b.corr.by = 10,
m0 = 5, wei.shape = 1, recr = 4, fu = 2,
timefactor = 1,
active.inter = c("bm003", "bm021", "bm044", "bm049", "bm097")
)
resBM <- BMsel(
data = sdata,
method = c("lasso", "lasso-pcvl"),
inter = TRUE,
folds = 5)
esurv <- expSurv(
res = resBM,
traindata = sdata,
boot = TRUE,
nboot = 100,
smooth = TRUE,
pct.group = 4,
time = 5,
ncores = 5)
plot(esurv, method = "lasso", pr.group = 3)

## End(Not run)

# Breast cancer data set
 predRes

Evaluation of the prediction accuracy of a prediction model

Description

This function computes several criteria to assess the prediction accuracy of a prediction model.

Usage

predRes(res, method, traindata, newdata, int.cv, int.cv.nfold = 5, time, trace = TRUE, ncores = 1)

## S3 method for class 'predRes'
plot(x, method, crit = c("C", "PE", "dC"), xlim, ylim, xlab, ylab, col,...)
Arguments

res
an object of class 'resBMsel' generated by \texttt{BMsel}.

method
methods for which prediction criteria are computed. If missing, all methods contained in \texttt{res} are computed.

traindata
input \texttt{data.frame} used to compute the \texttt{res} object. This object is mandatory.

newdata
input \texttt{data.frame} not used to compute the \texttt{res} object. This object is not mandatory (see Details section).

int.cv
logical parameter indicating if a double cross-validation process (2CV) should be performed to mimic an external validation set.

int.cv.nfold
number of folds for the double cross-validation. Considering a large value for \texttt{int.cv.nfold} should provide extremely large computation time. \texttt{int.cv.nfold} must not be considered when \texttt{int.cv = FALSE}.

time
time points to compute the prediction criteria.

trace
logical parameter indicating if messages should be printed.

ncores
number of CPUs used (for the double cross-validation).

x
an object of class 'predRes' generated from \texttt{predRes}.

crit
parameter indicating the criterion for which the results will be printed (C: concordance via Uno’s C-statistic, PE: prediction error via integrated Brier score and dC: delta Uno’s C-statistic (for the interaction setting only)).

\texttt{xlim, ylim, xlab, ylab, col}
usual parameters for \texttt{plot}.

\texttt{...}
other parameters for \texttt{plot}.

Details

To evaluate the accuracy of the selected models, three predictive accuracy measures are implemented:
- the integrated Brier score (PE) to measure the overall prediction error of the prediction model. The time-dependent Brier score is a quadratic score based on the predicted time-dependent survival probability.
- the Uno’s C-statistic (C) to evaluate the discrimination of the prediction model. It’s one of the least biased concordance statistic estimator in the presence of censoring (Uno et al., 2011).
- the absolute difference of the treatment-specific Uno’s C-statistics (dC) to evaluate the interaction strength of the prediction model (Ternes et al., 2016).

For simulated datasets, the predictive accuracy metrics are also computed for the “oracle model” that is the unpenalized Cox proportional hazards model fitted to the active biomarkers only.

Value

A list of the same length of the \texttt{time} considered. Each element of the list contains between 1 and 3 sublists depending on the chosen validation (i.e. training set [always computed], internal validation through double cross-validation (2CV) [if \texttt{int.cv = TRUE}] and/or external validation [if \texttt{newdata} is provided]). Each sublist is a \texttt{matrix} containing the predictive accuracy metrics of the implemented methods.
Author(s)

Nils Ternes, Federico Rotolo, and Stefan Michiels
Maintainer: Nils Ternes <nils.ternes@yahoo.com>

References


Examples

```r
# Simulated data set
## Simulated data set
## Low calculation time
set.seed(654321)
sdata <- simdata(
  n = 500, p = 20, q.main = 3, q.inter = 0,
  prob.tt = 0.5, alpha.tt = 0,
  beta.main = -0.8,
  b.corr = 0.6, b.corr.by = 4,
  m0 = 5, wei.shape = 1, recr = 4, fu = 2,
  timefactor = 1)

newdata <- simdataV(
  traindata = sdata,
  Nvalid = 500
)

resBM <- BMsel(
  data = sdata,
  method = c("lasso", "lasso-pcvl"),
  inter = FALSE,
  folds = 5)

predAcc <- predRes(
  res = resBM,
  traindata = sdata,
  newdata = newdata,
  time = 1:5)

plot(predAcc, crit = "C")
```
predRes

```r
## Not run:
## Moderate calculation time
set.seed(123456)
sdata <- simdata(
  n = 500, p = 100, q.main = 5, q.inter = 5,
  prob.tt = 0.5, alpha.tt = -0.5,
  beta.main = c(-0.5, -0.2), beta.inter = c(-0.7, -0.4),
  b.corr = c(0.5, 0.01), b.corr.by = 0,
  m0 = 5, wei.shape = 1, recr = 4, fu = 2,
  timefactor = 1,
  active.inter = c("bm003", "bm021", "bm044", "bm049", "bm097"))

resBM <- BMsel(
  data = sdata,
  method = c("lasso", "lasso-pcvl"),
  inter = TRUE,
  folds = 5)

predAcc <- predRes(
  res = resBM,
  traindata = sdata,
  int.cv = TRUE,
  time = 1:5,
  ncores = 5)
plot(predAcc, crit = "dC")

## End(Not run)
```

```
# Breast cancer data set

```
Description

This function computes several criteria to assess the selection accuracy of a prediction model. Of note, this function is only available for simulated data sets for which true biomarkers are known.

Usage

`selRes(res)`

Arguments

`res` an object of class `resBMSel` generated by `BMSel` with data simulated using `simdata`.

Details

Based on the 2x2 contingency table (active vs. inactive / selected vs. unselected), four selection criteria are provided:
- the false discovery rate (FDR) that is the proportion of inactive biomarkers among the selected ones,
- the false non-discovery rate (FNDR) that is the proportion of active biomarkers among the unselected ones,
- the false negative rate (FNR) that is the proportion of unselected biomarkers among the active ones,
- the false positive rate (FPR) that is the proportion of selected biomarkers among the inactive ones.

These four criteria are between 0 and 1, and must be minimized.

We also provided two discrimination criteria, translating the ability to discard inactive biomarkers more likely than active ones independently of the tuning parameters:
- the area under the ROC curve (AUC) depending on the sensitivity [1 - FNR] and specificity [1 - FPR],
- the area under the precision-recall curve (AUPRC) depending on the FNR and FDR (Davis and Goadrich, 2006).

Of note, the AUPRC is more meaningful than the AUC when there are many more inactive than active biomarkers. These two criteria are between 0 and 1, and must be maximized.

Value

A matrix of dimension 6 x the number of methods used to fit `res`. 
Author(s)
Nils Ternes, Federico Rotolo, and Stefan Michiels
Maintainer: Nils Ternes <nils.ternes@yahoo.com>

References

Examples

```
# Simulated data set
n = 500, p = 20, q.main = 3, q.inter = 0,
prob.tt = 0.5, alpha.tt = 0,
beta.main = -0.8,
b.corr = 0.6, b.corr.by = 4,
m0 = 5, wei.shape = 1, recr = 4, fu = 2,
timefactor = 1)
```

```
# Low calculation time
set.seed(654321)
sdata <- simdata(
  n = 500, p = 20, q.main = 3, q.inter = 0,
  prob.tt = 0.5, alpha.tt = 0,
  beta.main = -0.8,
b.corr = 0.6, b.corr.by = 4,
m0 = 5, wei.shape = 1, recr = 4, fu = 2,
timefactor = 1)
```

```
resBM <- BMsel(
data = sdata,
method = c("lasso", "lasso-pcv1"),
inter = FALSE,
folds = 5)
selAcc <- selRes(resBM)
```

```
# Not run:
# Moderate calculation time
set.seed(123456)
sdata <- simdata(
  n = 500, p = 100, q.main = 5, q.inter = 5,
  prob.tt = 0.5, alpha.tt = -0.5,
  beta.main = c(-0.5, -0.2), beta.inter = c(-0.7, -0.4),
b.corr = 0.6, b.corr.by = 10,
m0 = 5, wei.shape = 1, recr = 4, fu = 2,
timefactor = 1,
active.inter = c("bmain03", "bmain021", "bmain044", "bmain049", "bmain097"))
```
simdata

Generation of data sets with survival outcome

Description

This function simulates a data set with survival outcome with given active biomarkers (prognostic and/or interacting with the treatment).

Usage

\[ \text{simdata}(n, p, q.\text{main}, q.\text{inter}, \text{prob.tt}, m0, \alpha.\text{tt}, \beta.\text{main}, \beta.\text{inter}, b.\text{corr}, b.\text{corr.by}, \text{wei.shape}, \text{recr}, \text{fu}, \text{timefactor}, \text{active.main}, \text{active.inter}) \]

\[ \text{simdataV(traindata, Nvalid)} \]

Arguments

- \( n \) - the sample size.
- \( p \) - the number of biomarkers.
- \( q.\text{main} \) - the number of true prognostic biomarkers.
- \( q.\text{inter} \) - the number of true biomarkers interacting with the treatment.
- \( \text{prob.tt} \) - the treatment assignment probability.
- \( m0 \) - the baseline median survival time.
- \( \alpha.\text{tt} \) - the effect of the treatment (in log-scale).
- \( \beta.\text{main} \) - the effect of the prognostic biomarkers (in log-scale).
- \( \beta.\text{inter} \) - the effect of the biomarkers interacting with the treatment (in log-scale).
- \( b.\text{corr} \) - the correlation between biomarker blocks.
- \( b.\text{corr.by} \) - the size of the blocks of correlated biomarkers.
- \( \text{wei.shape} \) - the shape parameter of the Weibull distribution.
- \( \text{recr} \) - the recruitment period duration.
- \( \text{fu} \) - the follow-up period duration.
- \( \text{timefactor} \) - the scale multiplicative factor for times (i.e. 1 = times in years).
active.main  the list of the prognostic biomarkers (not mandatory).
active.inter  the list of the biomarkers interacting with the treatment (not mandatory).
traintdata  the training set returned by simdata, used to generate the new validation data set with the same characteristics.
Nvalid  the sample size of the new validation data set.

Details

The simdata function generates p Gaussian unit-variance (σ = 1) biomarkers including autoregressive correlation (σ_{ij} = b.corr^{|i-j|}) within b.corr by biomarker blocks. The number of active biomarkers and their effect sizes (in log-scale) can be specified using q.main and beta.main for the true prognostic biomarkers and using q.inter and beta.inter for the true treatment-effect modifiers. A total of n patients is generated and randomly assigned to the experimental (coded as +0.5, with probability prob.tt) and control treatment (coded as -0.5). The treatment effect is specified using alpha.tt (in log-scale). Survival times are generated using a Weibull with shape wei.shape (i.e. 1 = exponential distribution) and patient-specific scale depending on the baseline median survival time m0 and the biomarkers values of the patient. Censor status is generated by considering independent censoring from a U(fu, fu + recr) distribution, reflecting a trial with recr years of accrual and fu years of follow-up. Another data set with the same characteristics as the one generated by simdata can be obtained by using the simdataV function.

Value

A simulated data.frame object.

Author(s)

Nils Ternes, Federico Rotolo, and Stefan Michiels
Maintainer: Nils Ternes <nils.ternes@yahoo.com>

Examples

```r
set.seed(123456)
sdata <- simdata(
  n = 500, p = 100, q.main = 5, q.inter = 5,
  prob.tt = 0.5, alpha.tt = -0.5,
  beta.main = c(-0.5, -0.2), beta.inter = c(-0.7, -0.4),
  b.corr = 0.6, b.corr.by = 10,
  m0 = 5, wei.shape = 1, recr = 4, fu = 2,
  timefactor = 1,
  active.inter = c("bm003", "bm021", "bm044", "bm049", "bm097"))

newdata <- simdataV(
  traintdata = sdata,
  Nvalid = 500)
```
Index

*Topic biomarker selection
  BMsel, 2
*Topic breast
  Breast, 6
*Topic cancer
  Breast, 6
*Topic confidence intervals
  expSurv, 7
*Topic dataset
  Breast, 6
*Topic data
  simdata, 16
*Topic estimations
  expSurv, 7
*Topic interactions
  BMsel, 2
*Topic prediction
  predRes, 10
*Topic prognostic
  BMsel, 2
*Topic selection
  selRes, 14
*Topic simulation
  simdata, 16
*Topic surrogate
  Breast, 6
*Topic survival
  BMsel, 2
  expSurv, 7

BMsel, 2, 7, 8, 11, 14
Breast, 6

data.frame, 2, 8, 11

expSurv, 7, 8

plot, 3, 8, 11
plot.predRes (predRes), 10
plot.resexpSurv (expSurv), 7

predict, 8
predict.resexpSurv (expSurv), 7
predRes, 10, 11
selRes, 14
simdata, 14, 16
simdataV (simdata), 16
summary, 3
summary.resBMsel (BMsel), 2