Package 'surrosurv'

January 18, 2019

Type Package

Title Evaluation of Failure Time Surrogate Endpoints in Individual Patient Data Meta-Analyses

Version 1.1.25

Date 2019-01-18

Author Federico Rotolo [aut],
Xavier Paoletti [ctr],
Marc Buyse [ctr],
Tomasz Burzykowski [ctr],
Stefan Michiels [ctr, cre]

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

Description Provides functions for the evaluation of surrogate endpoints when both the surrogate and the true endpoint are failure time variables. The approaches implemented are:
(1) the two-step approach (Burzykowski et al, 2001) <DOI:10.1111/1467-9876.00244> with a copula model (Clayton, Plackett, Hougaard) at the first step and either a linear regression of log-hazard ratios at the second step (either adjusted or not for measurement error);
(2) mixed proportional hazard models estimated via mixed Poisson GLM (Rotolo et al, 2019 <DOI:10.1177/0962280217718582>).

Depends R (>= 2.10), stats, optimx, grDevices

Imports copula, eha, lme4, MASS, Matrix, msm, mvmeta, optextras, parallel, parfm, survival, SurvCorr

License GPL-2

VignetteBuilder R.rsp

Suggests R.rsp

Encoding UTF-8

NeedsCompilation no

Repository CRAN

Date/Publication 2019-01-18 21:50:03 UTC

R topics documented:

surrosurv-package 2
convergence 4
gastadj 5
gastadv 6
loocv 7
poissonize 9
simData 12
ste 15
surrosurv 16

Index **20**

`surrosurv-package` *Evaluation of Failure Time Surrogate Endpoints in Individual Patient Data Meta-Analyses*

Description

Provides functions for the evaluation of surrogate endpoints when both the surrogate and the true endpoint are failure time variables. The approaches implemented are: (1) the two-step approach (Burzykowski et al, 2001) <DOI:10.1111/1467-9876.00244> with a copula model (Clayton, Plackett, Hougaard) at the first step and either a linear regression of log-hazard ratios at the second step (either adjusted or not for measurement error); (2) mixed proportional hazard models estimated via mixed Poisson GLM (Rotolo et al, 2019 <DOI:10.1177/0962280217718582>).

Details

The DESCRIPTION file:

Package:	surrosurv
Type:	Package
Title:	Evaluation of Failure Time Surrogate Endpoints in Individual Patient Data Meta-Analyses
Version:	1.1.25
Date:	2019-01-18
Authors@R:	c(person("Federico", "Rotolo", role="aut"), person("Xavier", "Paoletti", role="ctr"), person("Marc", "Buy
Author:	Federico Rotolo [aut], Xavier Paoletti [ctr], Marc Buyse [ctr], Tomasz Burzykowski [ctr], Stefan Michiels
Maintainer:	Stefan Michiels <stefan.michiels@gustaveroussy.fr>
Description:	Provides functions for the evaluation of surrogate endpoints when both the surrogate and the true endpoint
Depends:	R (>= 2.10), stats, optimx, grDevices
Imports:	copula, eha, lme4, MASS, Matrix, msm, mvmeta, optextras, parallel, parfm, survival, SurvCorr
License:	GPL-2
VignetteBuilder:	R.rsp
Suggests:	R.rsp
Encoding:	UTF-8

Index of help topics:

```
convergence             Assesses the convergence of fitted models for
                        surrogacy evaluation
gastadj                 Individual data from the adjuvant GASTRIC
                        meta-analysis
gastadv                 Individual data from the advanced GASTRIC
                        meta-analysis
loocv                   Leave-one-trial-out cross-validation for
                        treatment effect prediction
poissonize              Tranform survival data for fitting a Poisson
                        model
simData.re              Generate survival times for two endpoints in a
                        meta-analysis of randomized trials
ste                     Surrogate threshold effect
surrosurv               Fit and print the models for evaluating the
                        surrogacy strength of a candidate surrogate
                        endpoint
surrosurv-package       Evaluation of Failure Time Surrogate Endpoints
                        in Individual Patient Data Meta-Analyses
```

Author(s)

Federico Rotolo [aut], Xavier Paoletti [ctr], Marc Buyse [ctr], Tomasz Burzykowski [ctr], Stefan Michiels [ctr, cre]

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

References

Rotolo F, Paoletti X, Michiels S. surrosurv: an R Package for the Evaluation of Failure Time Surrogate Endpoints in Individual Patient Data Meta-Analyses of Randomized Clinical Trials. *Computer Methhods and Programs in Biomedicine* 2018; doi: 10.1016/j.cmpb.2017.12.005

Rotolo F, Paoletti X, Burzykowski T, Buyse M, Michiels S. A Poisson approach for the validation of failure time surrogate endpoints in individual patient data meta-analyses. *Statistical Methhods in Medical Research* 2019; **28(1)**. doi: 10.1177/0962280217718582

Burzykowski T, Molenberghs G, Buyse M et al. Validation of surrogate end points in multiple randomized clinical trials with failure time end points. *Journal of the Royal Statistical Society C* 2001; **50**:405–422. doi: 10.1111/1467-9876.00244

Gasparrini A, Armstrong B, Kenward MG. Multivariate meta-analysis for non-linear and other multi-parameter associations. *Statistics in Medicine* 2012; **31**:3821–39. doi: 10.1002/sim.5471

Burzykowski T, Molenberghs G, Buyse M (2005). *The Evaluation of Surrogate Endpoints*. Springer, New York. `http://rd.springer.com/book/10.1007/b138566`

See Also

Surrogate, mvmeta

`convergence` *Assesses the convergence of fitted models for surrogacy evaluation*

Description

This function evaluates whether the fitted models for evaluating the surrogacy of a candidate end-point have converged. Convergence is assessed by checking whether the maximum gradient is small enough, and whether the Hessian matrix and the variance-covariance matrix of random treatment effects are positive definite.

Usage

```
  ## S3 method for class 'surrosurv'
convals(x, ...)
  ## S3 method for class 'surrosurv'
convergence(x, kkttol = 1e-2, kkt2tol = 1e-8, ...)
```

Arguments

`x`	The fitted models, an object of class `surrosurv`.
`kkttol`	The tolerance threshold for the assessing whether the maximum (absolute) scaled gradient is small enough.
`kkt2tol`	The tolerance threshold for checking whether the Hessian matrix and the variance-covariance matrix of random treatment effects are positive definite. The threshold is for the minimum of the eigenvalues.
`...`	Further parameters (not implemented)

Value

The function `convals()` returns a matrix with one row per model and three columans, reporting the values of the maximum scaled gradient (`maxSgrad`), of the minimum eigenvalue of the Hessian matrix (`minHev`), and of the minimum eigenvalue of the estimated variance-covariance matrix of random treatment effects (`minREev`). The function `convergence()` returns a matrix with the same structure as `convals()`, with `TRUE`/`FALSE` values for the test of the results of `convals()` against the given thresholds `kkttol` and `kkt2tol`.

Author(s)

Federico Rotolo [aut], Xavier Paoletti [ctr], Marc Buyse [ctr], Tomasz Burzykowski [ctr], Stefan Michiels [ctr, cre] gastadj *Individual data from the adjuvant GASTRIC meta-analysis*

Description

The `gastadj` dataset contains individual data (overall and disease-free survival) of 3288 patients with resectable gastric cancer from 14 randomized trials of adjuvant chemotherapy.

Usage

```
data(gastadj)
```

Format

A dataframe with variables:

timeT: Overall survival time (days).

statusT: Overall survival indicator (0=censored, 1=death).

timeS: Disease-free survival time (days).

statusS: Disease-free survival indicator (0=censored, 1=prgoression on death).

trialref: Trial indicator

trt: Treatment arm (-0.5 = control, 0.5=chemomtherapy).

id: Patient identifier.

Source

The authors thank the GASTRIC (Global Advanced/Adjuvant Stomach Tumor Research International Collaboration) Group for permission to use their data. The investigators who contributed to GASTRIC are listed in Oba et al (2013) and GASTRIC (2010). The GASTRIC Group data are available within the `surrosurv` package for research purposes, under the conditions that (1) the research be scientifically appropriate, (2) the confidentiality of individual patient data be protected, (3) the results of the analyses be shared with the GASTRIC Group prior to public communication, (4) the source of data be fully acknowledged as above, and (5) resulting data and results be further shared with the research community.

References

Paoletti X, Oba K, Bang Y-J, et al. Disease-free survival as a surrogate for overall survival in adjuvant trials of gastric cancer: a meta-analysis. *J Ntl Cancer Inst*, 105(21):1600-7, 2013. doi:10.1093/jnci/djt270.

The GASTRIC group. Benefit of adjuvant chemotherapy for resectable gastric cancer: a meta-analysis. *JAMA*, 303(17):1729-37, 2010. doi:10.1001/jama.2010.534.

Buyse M, Molenberghs G, Paoletti Xavier et al. Statistical evaluation of surrogate endpoints with examples from cancer clinical trials. *Biom J*, 58(1):104-32, 2016. doi:10.1002/bimj.201400049

Examples

```
## Not run:
  data('gastadj')
  allSurroRes <- surrosurv(gastadj, c('Clayton', 'PoissonTIa'), verbose = TRUE)
  convergence(allSurroRes)
  allSurroRes
  predict(allSurroRes)
  plot(allSurroRes)

## End(Not run)
```

---

gastadv *Individual data from the advanced GASTRIC meta-analysis*

Description

The gastadv dataset contains individual data (overall and progression-free survival) of 4069 patients with advanced/recurrent gastric cancer from 20 randomized trials of chemotherapy.

Usage

```
data(gastadv)
```

Format

A dataframe with variables:

timeT: Overall survival time (days).

statusT: Overall survival indicator (0=censored, 1=death).

timeS: Progression-free survival time (days).

statusS: Progression-free survival indicator (0=censored, 1=prgoression on death).

trialref: Trial indicator

trt: Treatment arm (-0.5 = control, 0.5=chemomtherapy).

id: Patient identifier.

Source

The authors thank the GASTRIC (Global Advanced/Adjuvant Stomach Tumor Research International Collaboration) Group for permission to use their data. The investigators who contributed to GASTRIC are listed in Paoletti et al (2013) and GASTRIC (2013). The GASTRIC Group data are available within the surrosurv package for research purposes, under the conditions that (1) the research be scientifically appropriate, (2) the confidentiality of individual patient data be protected, (3) the results of the analyses be shared with the GASTRIC Group prior to public communication, (4) the source of data be fully acknowledged as above, and (5) resulting data and results be further shared with the research community.

References

Paoletti X, Oba K, Bang Y-J, et al. Progression-free survival as a surrogate for overall survival in advanced/recurrent gastric cancer trials: a meta-analysis. *J Ntl Cancer Inst*, 105(21):1667-70, 2013. doi:10.1093/jnci/djt269.

The GASTRIC group. Role of chemotherapy for advanced/recurrent gastric cancer: An individual-patient-data meta-analysis. *Eur J Cancer*, 49(7):1565-77, 2013. doi:10.1016/j.ejca.2012.12.016.

Buyse M, Molenberghs G, Paoletti Xavier et al. Statistical evaluation of surrogate endpoints with examples from cancer clinical trials. *Biom J*, 58(1):104-32, 2016. doi:10.1002/bimj.201400049

Examples

```
  ## Not run:
    data('gastadv')
    allSurroRes <- surrosurv(gastadv, c('Clayton', 'PoissonTIa'), verbose = TRUE)
    convergence(allSurroRes)
    allSurroRes
    predict(allSurroRes)
    plot(allSurroRes)

## End(Not run)
```

---

loocv — *Leave-one-trial-out cross-validation for treatment effect prediction*

Description

The function `loocv()` computed leave-one-out predcition of the treatment effect on the true end-point for each trial, based on the observed effect on the surrogate endpoint in the trial itself and based on the meta-analytic model fitted on the remaining trials (Michiels et al, 2009).

Usage

```
## S3 method for class 'surrosurv'
loocv(object, models, nCores, parallel = TRUE, ...)

## S3 method for class 'loocvSurrosurv'
print(x, n = min(length(x), 6), silent = FALSE, ...)

## S3 method for class 'loocvSurrosurv'
plot(x, models, exact.models,
                                plot.type = c('classic', 'regression'),
                                main, ylab, xlab, ...)
```

Arguments

`object` Either an object of class `surrosurv` with an attribute `data` of class data.frame or a data.frame with columns
- `trialref`, the trial reference
- `trt`, the treatment arm (-0.5 or 0.5)
- `id`, the patient id
- `timeT`, the value of the true endpoint T
- `statusT`, the censoring/event (0/1) indicator of the true endpoint T
- `timeS`, the value of the surrogate endpoint S
- `statusS`, the censoring/event (0/1) indicator of the surrogate endpoint S

`nCores` The number of cores for parallel computing

`parallel` Should results be computed using parallelization?

`models, exact.models` Which models should be fitted (see `surrosurv()`). By default, the same models fitted in `object` (or `x`).

`x` The fitted models, an object of class `surrosurv`

`n` the number of rows to print

`silent` Should the results be return for storing without printing them?

`plot.type` The type ox x-scale for the loocv plot: either the trial number (`classic`) or the log-HR on the surrogate endpoint (`regression`).

`main, ylab, xlab, ...` Further parameters to be passed to `surrosurv` (for `loocv()`) or to the generics `print()` and `plot()`

Value

An object of class `loocvSurrosurv` containing, for each trial:

`margPars` the observed treatment effects on the surrogate ednpoint (`alpha`) and on the true endpoint (`beta`)

`...` for each method in `models` the predicted value and prediction interval for `beta`.

Author(s)

Federico Rotolo [aut], Xavier Paoletti [ctr], Marc Buyse [ctr], Tomasz Burzykowski [ctr], Stefan Michiels [ctr, cre]

References

Michiels S, Le Maitre A, Buyse M, et al. Surrogate endpoints for overall survival in locally advanced head and neck cancer: meta-analyses of individual patient data. *Lancet Oncol.* 2009;10(4):341-50. doi:10.1016/S1470-2045(09)70023-3

Examples

```
# Possibly long computation time!
data('gastadv')
cvRes <- loocv(gastadv)
cvRes
plot(cvRes)
```

---

poissonize — *Tranform survival data for fitting a Poisson model*

Description

This function tranform survival data into a format compatible with the `glm()` function for fitting an auxiliary Poisson model, providing the parameter estimates of the associated proportional hazard model.

Usage

```
poissonize(data,
           all.breaks = NULL, interval.width = NULL, nInts = 8,
           factors = NULL, compress = TRUE)
plotsson(x, type = c('survival', 'hazard'),
         add = FALSE, xscale = 1, by, col, ...)
```

Arguments

`data` a data frame with columns:

- id : the patient identifyier
- time : the event/censoring time
- status : the event(1) or censoring(0) indicator
- ... : other factors such like the covariables needed in the regression model

`all.breaks` the breakpoints between time intervals

`interval.width` the width of the time intervals on which the risks will be assumed consant, in case of intervals of the same length. This parameter is ignored if `all.breaks` is specified

`nInts` the number of intervals containing the same expected number of events (used only if `is.null(interval.width)`, see Details). This parameter is ignored if either `all.breaks` or `interval.width` is specified

`factors` a vector of characters, containing the names of the factors to be kept in the transformed data set

`compress` a logical, indicating whether the record with the same factor profile should be summarized into one record, i.e. whether the data should be expressed in a short form

x	The fitted Poisson model on the poissonized data
type	the type of plot, either 'haz' for the hazard function or 'Surv', for the survival curve
add	should the plot added to the active device?
xscale	scaling factor for the time (x) axis
by	covariate for which a different curve per level has to be plotted
col, ...	other graphical parameters

Details

If `interval.width` is not null, the study period is diveded into equal-length intervals of length `interval.width`. Otherwise, `nInts` intervals are used, and the location of their bounds is computed based on the empirical quantiles of the survival function.

Note

This code is hugely inspired by original code made publicly available by Stephanie Kovalchik [web link]

Author(s)

Federico Rotolo [aut], Xavier Paoletti [ctr], Marc Buyse [ctr], Tomasz Burzykowski [ctr], Stefan Michiels [ctr, cre]

References

Whitehead, J. Fitting Cox's regression model to survival data using GLIM. *J Roy Stat Soc C Appl Stat* 1980; **29**(3):268-275. `http://www.jstor.org/stable/2346901`.

Crowther MJ, Riley RD, Staessen JA, Wang J, Gueyffier F, Lambert PC. Individual patient data meta-analysis of survival data using Poisson regression models. *BMC Medical Research Methodology* 2012; **12**:34. doi: 10.1186/1471-2288-12-34.

Examples

```
################################################################################
# Example 1 - KIDNEY data                                                      #
################################################################################
library(survival)
data(kidney)
kidney <- kidney[1:(nrow(kidney)/2)*2,]
head(kidney)

par(mfrow=c(1, 3))
for (int in c(50, 20, 10)) {
    head(wdata1 <- poissonize(kidney, interval.width = int,
                              factors = c('disease'), compress = FALSE))
    head(wdata2 <- poissonize(kidney, interval.width = int,
                              factors = c('disease'), compress = TRUE))
```

```
    fitcox <- (coxph(Surv(time, status) ~ disease, data = kidney))
    fitpoi1 <- glm(event ~ -1 + interval + disease + offset(log(time)),
                   data = wdata1, family = 'poisson')
    fitpoi2 <- glm(m ~ -1 + interval + offset(log(Rt)) + disease,
                   data = wdata2, family = 'poisson')
    cox.base <- basehaz(fitcox, centered = FALSE)
    plot(stepfun(cox.base$time[-nrow(cox.base)], exp(-cox.base$hazard)),
         ylim = 0:1, xlim = c(0, max(cox.base$time)),
         do.points = FALSE, verticals = FALSE, xaxs = 'i',
         main = paste0('KIDNEY data set\nInterval width = ', int),
         xlab = 'Time', ylab = 'Survival probability')
    plotsson(fitpoi1, 'Surv', add = TRUE, col = 2, lty = 2)
    plotsson(fitpoi2, 'Surv', add = TRUE, col = 3, lty = 3)
    legend('topright', col = 1:3, lty = 1:3,
           legend = c('Breslow (Cox)', 'Poisson',
                      'Poisson (compressed dataset)'))
}
print(cbind(Cox                = coef(fitcox),
            Poisson            = rev(rev(coef(fitpoi1))[1:3]),
            Poisson_Compressed = rev(rev(coef(fitpoi2))[1:3])), digits = 2)



################################################################################
#  Example 2 - COLON data                                                      #
################################################################################
library(survival)
data(colon)
head(wdata1 <- poissonize(subset(colon, etype == 1), interval.width = 365.25,
                          factors=c('surg', 'sex', 'age'), compress = FALSE))
head(wdata2 <- poissonize(subset(colon, etype == 1), interval.width = 365.25,
                          factors=c('surg', 'sex', 'age'), compress = TRUE))

fitcox <- coxph(Surv(time, status) ~ surg + sex + age,
                data = subset(colon, etype == 1))

system.time({
    fitpoi1 <- glm(event ~ -1 + interval + surg + sex + age + offset(log(time)),
                   data = wdata1, fam = 'poisson')
})
system.time({
    fitpoi2 <- glm(m ~ -1 + interval + offset(log(Rt)) + surg + sex + age,
                   data = wdata2, family = 'poisson')
})
{
    cox.base <- basehaz(fitcox, centered = FALSE)
    par(mfrow = c(1, 1))
    plot(stepfun(cox.base$time[-nrow(cox.base)], exp(-cox.base$hazard)),
         ylim = 0:1, xlim = c(0, max(cox.base$time)),
         do.points = FALSE, verticals = FALSE, xaxs = 'i',
         main = 'COLON data set', xlab = 'Time', ylab = 'Survival probability')
    plotsson(fitpoi1, 'Surv', add = TRUE, col = 2, lty = 2)
    plotsson(fitpoi2, 'Surv', add = TRUE, col = 3, lty = 3)
```

```
    legend('topright', col = 1:3, lty = 1:3,
           legend = c('Cox', 'Poisson', 'Poisson (compressed dataset)'))
}
print(cbind(Cox                = coef(fitcox),
            Poisson            = rev(rev(coef(fitpoi1))[1:3]),
            Poisson_Compressed = rev(rev(coef(fitpoi2))[1:3])), digits = 2)



################################################################################
#  Example 3 - LUNG data                                                       #
################################################################################
library(survival)
data(lung)
lung$status <- lung$status - 1
lung$id <- 1:nrow(lung)
head(wdata1 <- poissonize(lung, interval.width = 365.25/12,
                          factors = c('pat.karno', 'sex', 'age'),
                          compress = FALSE))
head(wdata2 <- poissonize(lung, interval.width = 365.25/12,
                          factors = c('pat.karno', 'sex', 'age'),
                          compress = TRUE))

fitcox <- coxph(Surv(time, status) ~ pat.karno + sex + age, data = lung)

system.time({
    fitpoi1 <- glm(event ~ -1 + interval + pat.karno + sex + age +
                              offset(log(time)),
                   data = wdata1, family = 'poisson')
})
system.time({
    fitpoi2 <- glm(m ~ -1 + interval + pat.karno + sex + age + offset(log(Rt)),
                   data = wdata2, family = 'poisson')
})
{
    cox.base <- basehaz(fitcox, centered = FALSE)
    plot(stepfun(cox.base$time[-nrow(cox.base)], exp(-cox.base$hazard)),
         ylim = 0:1, xlim = c(0, max(cox.base$time)),
         do.points = FALSE, verticals = FALSE, xaxs = 'i',
         main = 'LUNG data set', xlab = 'Time', ylab = 'Survival probability')
    plotsson(fitpoi1, 'Surv', add = TRUE, col = 2, lty = 2)
    plotsson(fitpoi2, 'Surv', add = TRUE, col = 3, lty = 3)
    legend('topright', col = 1:3, lty = 1:3,
           legend = c('Cox', 'Poisson', 'Poisson (compressed dataset)'))
}
print(cbind(Cox                = coef(fitcox),
            Poisson            = rev(rev(coef(fitpoi1))[1:3]),
            Poisson_Compressed = rev(rev(coef(fitpoi2))[1:3])), digits = 2)
```

simData *Generate survival times for two endpoints in a meta-analysis of randomized trials*

Description

Data are generated from a mixed proportional hazard model, a Clayton copula model (Burzykowski and Cortinas Abrahantes, 2005), a Gumbel-Hougaard copula model, or a mixture of half-normal and exponential random variables (Shi et al., 2011).

Usage

```
simData.re(R2 = 0.6, N = 30, ni = 200,
           nifix = TRUE, gammaWei = c(1, 1), censorT, censorA,
           kTau= 0.6, baseCorr = 0.5, baseVars = c(0.2, 0.2),
           alpha = 0, beta = 0,
           alphaVar = 0.1, betaVar = 0.1,
           mstS = 4 * 365.25, mstT = 8 * 365.25)

simData.cc(R2 = 0.6, N = 30, ni = 200,
           nifix = TRUE, gammaWei = c(1, 1), censorT, censorA,
           kTau= 0.6, baseCorr = 0.5, baseVars = c(0.2, 0.2),
           alpha = 0, beta = 0,
           alphaVar = 0.1, betaVar = 0.1,
           mstS = 4 * 365.25, mstT = 8 * 365.25)

simData.gh(R2 = 0.6, N = 30, ni = 200,
           nifix = TRUE, gammaWei = c(1, 1), censorT, censorA,
           kTau= 0.6, baseCorr = 0.5, baseVars = c(0.2, 0.2),
           alpha = 0, beta = 0,
           alphaVar = 0.1, betaVar = 0.1,
           mstS = 4 * 365.25, mstT = 8 * 365.25)

simData.mx(R2 = 0.6, N = 30, ni = 200,
           nifix = TRUE, gammaWei = c(1, 1), censorT, censorA,
           indCorr = TRUE, baseCorr = 0.5, baseVars = c(0.2, 0.2),
           alpha = 0, beta = 0,
           alphaVar = 0.1, betaVar = 0.1,
           mstS = 4 * 365.25, mstT = 8 * 365.25)
```

Arguments

`R2`	The desired trial-level surrogacy R^2
`N`	The number of trials
`ni`	The (fixed or average) number of patients per trial
`nifix`	Should all trials have the same size (if `nifix = TRUE`) of should the `N * ni` patients be randomly assigned to trials with random probabilities (if `nifix = FALSE`)?

gammaWei	The shape parameter(s) of the Weibull distributions. Either one or two values. If one value is provided, it is used for both endpoints
censorT	censoring rate for the true endpoint T (before adding administrative censoring)
censorA	administrative censoring at time censorA
kTau	The desired individual-level dependence between S and T (Kendall's tau)
indCorr	Should S and T be correlated or not? (for .mx method)
baseCorr	correlation between baseline hazards ($\rho_{basehaz}$)
baseVars	variances of baseline random effects (S and T)
alpha	average treatment effect on S
beta	average treatment effect on T
alphaVar	variance of a_i (θ_a^2)
betaVar	variance of b_i (θ_b^2)
mstS	median survival time for S in the control arm
mstT	median survival time for T in the control arm

Details

The function `simData.re` generates data from a proportional hazard model with random effects at individual level and random effects and random treatment effects at trial level. Individual dependence can be tuned in terms of Kendall's tau (`kTau`).

The function `simData.cc` generates data from a Copula function as shown by Burzykowski and Cortinas Abrahantes (2005). Individual dependence can be tuned in terms of Kendall's tau (`kTau`).

The function `simData.mx` implements the simulation method by Shi et al. (2011). This model is based on a mixture of half-normal and exponential random variables. Under this model, individual dependence can be induced by using the same half-normal random variable for S and T. This is obtained by setting `indCorr = TRUE`, but the amount of correlation is not dependent on a single parameter.

Value

A data.frame with columns

trialref	the trial reference
trt	the treatment arm (-0.5 or 0.5)
id	the patient id
timeT	the value of the true endpoint T
statusT	the censoring/event (0/1) indicator of the true endpoint T
timeS	the value of the surrogate endpoint S
statusS	the censoring/event (0/1) indicator of the surrogate endpoint S

Author(s)

Federico Rotolo [aut], Xavier Paoletti [ctr], Marc Buyse [ctr], Tomasz Burzykowski [ctr], Stefan Michiels [ctr, cre]

References

Burzykowski T, Cortinas Abrahantes J (2005). Validation in the case of two failure-time endpoints. In *The Evaluation of Surrogate Endpoints* (pp. 163-194). Springer, New York.

Rotolo F, Paoletti X, Burzykowski T, Buyse M, Michiels S. A Poisson approach for the validation of failure time surrogate endpoints in individual patient data meta-analyses. *Statistical Methhods in Medical Research* 2019; **28(1)**. doi: 10.1177/0962280217718582

Shi Q, Renfro LA, Bot BM, Burzykowski T, Buyse M, Sargent DJ. Comparative assessment of trial-level surrogacy measures for candidate time-to-event surrogate endpoints in clinical trials. *Computational Statistics & Data Analysis* 2011; **55**: 2748–2757.

Examples

```
set.seed(1)
simData.re(N = 2, ni = 5)
simData.cc(N = 2, ni = 5)
simData.mx(N = 2, ni = 5)
```

`ste` *Surrogate threshold effect*

Description

The function `ste()` computes the surrogate threshold effect (STE) of a .

Usage

```
ste(x, models = names(x), exact.models)

## S3 method for class 'steSurrosurv'
print(x, digits = 2, ...)
```

Arguments

`x`	The fitted models, an object of class `surrosurv`
`models, exact.models`	Which models should be fitted (see `surrosurv()`)
`digits`	the number of digits
`...`	Further parameters to be passed to the generic `print()` function

Value

An object of class `steSurrosurv` containing, for each trial:

Author(s)

Federico Rotolo [aut], Xavier Paoletti [ctr], Marc Buyse [ctr], Tomasz Burzykowski [ctr], Stefan Michiels [ctr, cre]

References

Burzykowski T, Buyse M. Surrogate threshold effect: an alternative measure for meta-analytic surrogate endpoint validation. *Pharm Stat*. 2006;5(3):173-86. doi:10.1002/pst.207

Examples

```
# Possibly long computation time!
data('gastadv')
mod <- surrosurv(gastadv, 'Clayton')
ste(mod)
```

---

surrosurv — *Fit and print the models for evaluating the surrogacy strength of a candidate surrogate endpoint*

Description

The function `surrosurv` fits (all or a subset of) statistical models to evaluate a surrogate endpoint S for a given true endpoint T, using individual data from a meta-analysis of randomized controlled trials.

Usage

```
surrosurv(data,
          models = c('Clayton', 'Plackett', 'Hougaard',
                     'Poisson I', 'Poisson T', 'Poisson TI', 'Poisson TIa'),
          intWidth = NULL,  nInts = 8,
          cop.OPTIMIZER = "bobyqa",
          poi.OPTIMIZER = "bobyqa",
          verbose = TRUE,
          twoStage = FALSE,
          keep.data = TRUE)

## S3 method for class 'surrosurv'
predict(object, models = names(object), exact.models, ...)

## S3 method for class 'surrosurv'
print(x, silent = FALSE,
      digits = 2, na.print = "-.--", ...)
```

```
## S3 method for class 'predictSurrosurv'
print(x, n = 6, ...)

## S3 method for class 'surrosurv'
plot(x, ...)

## S3 method for class 'predictSurrosurv'
plot(x, models = names(x), exact.models,
                      pred.ints = TRUE,
                      show.ste = TRUE,
                      surro.stats = TRUE,
                      xlab, ylab,
                      xlim, ylim, mfrow, main, ...)
```

Arguments

`data` A data.frame with columns
- `trialref`, the trial reference
- `trt`, the treatment arm (-0.5 or 0.5)
- `id`, the patient id
- `timeT`, the value of the true endpoint T
- `statusT`, the censoring/event (0/1) indicator of the true endpoint T
- `timeS`, the value of the surrogate endpoint S
- `statusS`, the censoring/event (0/1) indicator of the surrogate endpoint S

`models` For `surrosurv()`, the models should be fitted/plotted/predicted. Possible models are: Clayton copula (unadjusted and adjusted), Plackett copula (unadjusted and adjusted), Hougaard copula (unadjusted and adjusted), Poisson (with individual-level heterogeneity only, with trial-level heterogeneity only, with both individual- and trial-level heterogeneity, with both individual- and trial-level heterogeneity and with random per-trial intercept).

`exact.models` If `TRUE`, plots or predictions are generated only for the elements of `x` which match exactly any of `models`. If `exact.models = TRUE`, partial matching is used. By default, `exact.models = TRUE` if all the `models` match exactly any of the `names(x)` (or `names(object)`) and `exact.models = FALSE` otherwise.

`intWidth` the width of time intervals for data Poissonization (see poissonize)

`nInts` the number of time intervals for data Poissonization (see poissonize)

`cop.OPTIMIZER` the optimizer for copula models (see `optimx`)

`poi.OPTIMIZER` the optimizer for Poisson models (see `optimx`)

`verbose` should the function print out the model being fitted

`twoStage` should the parameters of the baseline hazard functions fixed to their marginal estimates (Shih and Louis, 1995)

`keep.data` should the data object be kept as attribute of the returned results? (this is needed for `confint.surrosurv()`)

`x, object` The fitted models, an object of class `surrosurv`

`silent`	Should the results be return for storing without printing them?
`digits, na.print, xlab, ylab, xlim, ylim, main, ...`	other parameters for `print` or `plot`
`mfrow`	the number of rows and columns for displaying the plots (see `par`). If missing, the default is computed using the function `n2mfrow`
`n`	the number of rows to print
`pred.ints`	Should the prediction intervals be plotted?
`show.ste`	Should the surrogate threshold effect be showed?
`surro.stats`	Should the surrogacy statistics be showed?

Details

Three copula models can be fit: Clayton (1978), Plackett (1965), and Hougaard (1986). For all of them the linear regression at the second step is computed both via simple LS regression and via a linear model adjusted for measurement error of the log-hazard ratios estimated at the first step. This adjusted model is the one described by Burzykowski et al. (2001), which relies on the results by van Houwelingen et al. (2002).

The moxed Poisson models that can be fit are used to estimate parameters of mixed proportional hazard models, as described for instance by Crowther et al (2014). The statistical details are provided in Rotolo et al (2019).

The function `predict()` returns the estimated values of the log-hazard ratios on the true and the surrogate endpoints. The list of the prediction functions (for all the `models`) is available as `attr(predict.surrosurv(...), 'predf')`.

Value

The fitted models, an object of class `surrosurv`.

Author(s)

Federico Rotolo [aut], Xavier Paoletti [ctr], Marc Buyse [ctr], Tomasz Burzykowski [ctr], Stefan Michiels [ctr, cre]

References

Burzykowski T, Molenberghs G, Buyse M et al. Validation of surrogate end points in multiple randomized clinical trials with failure time end points. *Journal of the Royal Statistical Society C* 2001; **50**:405–422. doi: 10.1111/1467-9876.00244

Clayton DG. A model for association in bivariate life tables and its application in epidemiological studies of familial tendency in chronic disease incidence. *Biometrika* 1978; **65**:141–151. doi: 10.1093/biomet/65.1.141

Crowther MJ, Riley RD, Staessen JA, Wang J, Gueyffier F, Lambert PC. Individual patient data meta-analysis of survival data using Poisson regression models. *BMC Medical Research Methodology* 2012; **12**:34. doi: 10.1186/1471-2288-12-34.

Gasparrini A, Armstrong B, Kenward MG. Multivariate meta-analysis for non-linear and other multi-parameter associations. *Statistics in Medicine* 2012; **31**:3821–39. doi: 10.1002/sim.5471

Hougaard P. A class of multivariate failure time distributions. *Biometrika* 1986; **73**:671–678. doi: 10.1093/biomet/73.3.671

Plackett RL. A class of bivariate distributions. *Journal of the America Statistical Association* 1965; **60**:516–522. doi: 10.1080/01621459.1965.10480807

Rotolo F, Paoletti X, Michiels S. surrosurv: an R Package for the Evaluation of Failure Time Surrogate Endpoints in Individual Patient Data Meta-Analyses of Randomized Clinical Trials. *Computer Methhods and Programs in Biomedicine* 2018; doi: 10.1016/j.cmpb.2017.12.005

Rotolo F, Paoletti X, Burzykowski T, Buyse M, Michiels S. A Poisson approach for the validation of failure time surrogate endpoints in individual patient data meta-analyses. *Statistical Methhods in Medical Research* 2019; **28(1)**. doi: 10.1177/0962280217718582

Shih JH, Louis TA. Inferences on the Association Parameter in Copula Models for Bivariate Survival Data. *Biometrics* 1995; **51**:1384–1399. doi: 10.2307/2533269

van Houwelingen HC, Arends LR, Stijnen T. Advanced methods in meta-analysis: multivariate approach and meta-regression. *Statistics in Medicine* 2002; **21**:589–624. doi: 10.1002/sim.1040

Examples

```
set.seed(150)
data <- simData.re(N = 20, ni = 250,
                   R2 = 0.8, kTau = 0.4,
                   alpha = log(0.95), beta = log(0.85),
                   censorA = 15 * 365.25)
library(survival)
par(mfrow = 1:2)
plot(survfit(Surv(timeS, statusS) ~ trt, data = data), lty = 1:2,
     xscale = 365.25, main = 'Progression-Free Survival\n(S)', col = 2)
plot(survfit(Surv(timeT, statusT) ~ trt, data = data), lty = 1:2,
     xscale = 365.25, main = 'Overall Survival\n(T)')


  # Long computation time!
  surrores <- surrosurv(data, verbose = TRUE)
  convergence(surrores)
  surrores


# Advanced GASTRIC data

  # Long computation time!
  data('gastadv')
  allSurroRes <- surrosurv(gastadv, c('Clayton', 'Poisson'), verbose = TRUE)
  convergence(allSurroRes)
  allSurroRes
  predict(allSurroRes)
  plot(allSurroRes)
```

Index

∗Topic **Clayton**
surrosurv, 16
surrosurv-package, 2
∗Topic **Hougaard**
surrosurv, 16
surrosurv-package, 2
∗Topic **KKT**
convergence, 4
∗Topic **Kuhn–Karush–Tucker conditions**
convergence, 4
∗Topic **Plackett**
surrosurv, 16
surrosurv-package, 2
∗Topic **Poisson**
poissonize, 9
surrosurv, 16
surrosurv-package, 2
∗Topic **Survival data**
poissonize, 9
∗Topic **\textasciitildekwd1**
simData, 13
∗Topic **\textasciitildekwd2**
simData, 13
∗Topic **adjuvant**
gastadj, 5
∗Topic **advanced**
gastadv, 6
∗Topic **cancer**
gastadj, 5
gastadv, 6
∗Topic **convergence**
convergence, 4
∗Topic **copula**
surrosurv, 16
surrosurv-package, 2
∗Topic **cross-validation**
loocv, 7
∗Topic **datasets**
gastadj, 5
gastadv, 6
∗Topic **gastadj**
gastadj, 5
∗Topic **gastadv**
gastadv, 6
∗Topic **gastric**
gastadj, 5
gastadv, 6
∗Topic **generalized linear mixed model**
surrosurv, 16
surrosurv-package, 2
∗Topic **leave-one-out**
loocv, 7
∗Topic **meta-analysis**
surrosurv, 16
surrosurv-package, 2
∗Topic **proportional hazard model**
surrosurv, 16
surrosurv-package, 2
∗Topic **randomized controlled trial**
surrosurv, 16
surrosurv-package, 2
∗Topic **ste**
ste, 15
∗Topic **surrogate endpoint**
surrosurv, 16
surrosurv-package, 2
∗Topic **surrogate threshold effect**
ste, 15
∗Topic **surrogate**
gastadj, 5
gastadv, 6
∗Topic **survival**
surrosurv, 16
surrosurv-package, 2

convals *(convergence)*, 4
convergence, 4

data.frame, *8*, *14*, *17*

gastadj, 5
gastadv, 6
glm, *9*
graphical parameters, *10*

loocv, 7

n2mfrow, *18*

optimx, *17*

par, *18*
plot, *18*
plot.loocvSurrosurv *(*loocv*)*, 7
plot.predictSurrosurv *(*surrosurv*)*, 16
plot.surrosurv *(*surrosurv*)*, 16
plotsson *(*poissonize*)*, 9
poissonize, 9, *17*
predict.surrosurv *(*surrosurv*)*, 16
print, *18*
print.loocvSurrosurv *(*loocv*)*, 7
print.predictSurrosurv *(*surrosurv*)*, 16
print.steSurrosurv *(*ste*)*, 15
print.surrosurv *(*surrosurv*)*, 16

simData, 12
ste, 15
surrosurv, *4*, *8*, *15*, 16, *17, 18*
surrosurv-package, 2