Package 'survivALL'

April 25, 2018

Title Continuous Biomarker Assessment by Exhaustive Survival Analysis

Version 0.9.3

Description In routine practice, biomarker performance is calculated by splitting a patient cohort at some arbitrary level, often by median gene expression. The logic behind this is to divide patients into "high" or "low" expression groups that in turn correlate with either good or poor prognosis. However, this median-split approach assumes that the data set composition adheres to a strict 1:1 proportion of high vs. low expression, that for every one "low" there is an equivalent "high". In reality, data sets are often heterogeneous in their composition (Perou, CM et al., 2000 <doi:10.1038/35021093>)- i.e. this 1:1 relationship is unlikely to exist and the true relationship unknown. Given this limitation, it remains difficult to determine where the most significant separation should be made. For example, estrogen receptor (ER) status determined by immunohistochemistry is standard practice in predicting hormone therapy response, where ER is found in an ~1:3 ratio (-:+) in the population (Selli, C et al., 2016 <doi:10.1186/s13058-016-0779-0>). We would expect therefore, upon dividing patients by ER expression, 25% to be classified "low" and 75% "high", and an otherwise 50-50 split to incorrectly classify 25% of our patient cohort, rendering our survival estimate under powered. 'survivALL' is a data-driven approach to calculate the relative survival estimates for all possible points of separation - i.e. at all possible ratios of "high" vs. "low" - allowing a measure's relationship with survival to be more reliably determined and quantified. We see this as a solution to a flaw in common research practice, namely the failure of a true biomarker as part of a meta-analysis.

Depends R (>= 3.3.0)

License MIT + file LICENSE

Encoding UTF-8

LazyData true

biocViews

Imports survival, survcomp, desiR, ggplot2, cowplot, ggthemes, viridis

Suggests testthat, survsim, broom, readr, pander, magrittr, Biobase, GGally, knitr, rmarkdown

RoxygenNote 6.0.1

VignetteBuilder knitr

NeedsCompilation no

Author Dominic Pearce [aut, cre]

Maintainer Dominic Pearce <dominic.pearce@ed.ac.uk>

Repository CRAN

Date/Publication 2018-04-25 10:02:14 UTC

R topics documented:

`allHR` *For all possible separation points for a cohort ordered by a continuous measurement, calculate hazard ratio*

Description

For all possible separation points for a cohort ordered by a continuous measurement, calculate hazard ratio

Usage

```
allHR(measure, srv, time = "Time", event = "Event", log2HR = TRUE,
  remove_outliers = TRUE)
```

Arguments

`measure`	A continuous variable used to order survival data. Samples must be ordered exactly as in srv
`srv`	A dataframe that contains at least two columns, detailing event and time to event information. Samples must be ordered exactly as in measure

time	Column name in srv containing time to event information. Must not contain NAs
event	Column name in srv containing event information coded as 0 (no event) and 1 (event). Must not contain NAs
log2HR	Hazard ratios are returned as log2 values by default
remove_outliers	Large hazard ratios result from statistical disproportion when considering edge cases (e.g. 1 vs 99) and can be automaticall removed

Value

A vector of hazard ratios calculated from srv ordered by measure

Examples

```
library(survivALL)
data(nki_subset)
library(Biobase)
gene_vec <- exprs(nki_subset)["NM_004448", ] #ERBB2 gene id
allHR(measure = gene_vec, srv = pData(nki_subset), time = "t.dmfs",
    event = "e.dmfs", log2HR = TRUE)
```

---

allPvals *For all possible separation points for a cohort ordered by a continuous measurement, perform a uni- or multivariate log-rank test*

Description

For all possible separation points for a cohort ordered by a continuous measurement, perform a uni- or multivariate log-rank test

Usage

```
allPvals(measure, srv, time = "Time", event = "Event", multiv = NULL,
  statistic = "logtest")
```

Arguments

measure	A continuous variable used to order survival data. Samples must be ordered exactly as in srv
srv	A dataframe that contains at least two columns, detailing event and time to event information. Samples must be ordered exactly as in measure
time	Column name in srv containing time to event information. Must not contain NAs
event	Column name in srv containing event information coded as 0 (no event) and 1 (event). Must not contain NAs

multiv	Univariate analysis is performed by default, however a character string specifying a column contained in srv (or a vector of strings specifying multiple columns) detailing additional variables can be included
statistic	the statistical test to be used to compute significance. one of "logtest" (likelihood ratio test), "waldtest" (wald statistic) or "sctest" (log-rank test)

Value

A vector of pvalues calculated from srv ordered by measure

Examples

```
library(survivALL)
data(nki_subset)
library(Biobase)
gene_vec <- exprs(nki_subset)["NM_004448", ] #ERBB2 gene id

allPvals(measure = gene_vec,
    srv = pData(nki_subset),
    time = "t.dmfs",
    event = "e.dmfs",
    statistic = "logtest")


allPvals(measure = gene_vec,
    srv = pData(nki_subset),
    time = "t.dmfs",
    event = "e.dmfs",
    multiv = "grade",
    statistic = "sctest")
```

---

areaBetweenCurves *Calculate the area above the bootstrapped thresholds but below the HR distribution and vice versa for each point of separation*

Description

Calculate the area above the bootstrapped thresholds but below the HR distribution and vice versa for each point of separation

Usage

```
areaBetweenCurves(survivALL_dfr)
```

Arguments

survivALL_dfr	Output of survivALL()

Value

A per-separation point vector equal to the distance beyond or within the bootstrapped thresholds

---

`bootstrapThresholds` *Calculate per-separation point hazard ratio thresholds*

Description

Calculate per-separation point hazard ratio thresholds

Usage

```
bootstrapThresholds(bs_dfr, n_sd = 1.96)
```

Arguments

`bs_dfr`	A matrix of bootstrapped hazard ratio computations as ordered by a random measurement vector. Typically consisting of 5-10,000 repeat samplings
`n_sd`	The number of standard deviations used to define threshold width. 95 deviation of 1.96

Value

A dataframe of per-separation point mean, upper and lower thresholds

Examples

```
data(nki_subset)
library(Biobase)
library(magrittr)
library(ggplot2)

#simulate example HR bootstrapped data
bs_dfr <- matrix(rnorm(150000), ncol = 1000, nrow = 150)

#calculate thresholds
thresholds <- bootstrapThresholds(bs_dfr)
```

checkContSig *Calculate association between survival and a continuous measure#' @inheritParams allPvals*

Description

Calculate association between survival and a continuous measure#' @inheritParams allPvals

Usage

```
checkContSig(measure, time, event)
```

Arguments

measure	A continuous variable used to order survival data. Samples must be ordered exactly as in time and event
time	A numeric vector of sample time-to-event data, ordered exactly as measure. Must not contain NAs
event	A vector of sample event data, ordered exactly as measure. Must not contain NAs

Value

p-value of association between measure and survival

Examples

```
library(survivALL)
library(Biobase)
data(nki_subset)

#Calculate p-value for continuous measure SCUBE2
srv_dfr <- data.frame(measure = exprs(nki_subset)["NM_020974", ],
                      time = nki_subset$t.dmfs,
                      event = nki_subset$e.dmfs
                      )

checkContSig(srv_dfr$measure, srv_dfr$time, srv_dfr$event)
```

compSelect *Given a dataframe of phenotypic information, use a variable (i.e. a single column) to define a patient subset of given proportion*

Description

Given a dataframe of phenotypic information, use a variable (i.e. a single column) to define a patient subset of given proportion

Usage

```
compSelect(pheno, column, values, props)
```

Arguments

pheno	a phenotypic dataframe. Sample IDs must be assigned as rownames
column	the name of the column used to define the subset - e.g. "grade"
values	the values within column that you are defining by These must be categorical - e.g. c("+", "-")
props	The number of how many of each value in column to be returned. - e.g. c(50, 50). Note, be careful not to ask for more samples of a particular value than are available in the dataset

Value

A dataframe, which is the subset of pheno, with a specified proportion of each value found in column

Examples

```
library(survivALL)
data(nki_subset)
library(Biobase)
pheno <- pData(nki_subset)

compSelect(pheno, "grade", values = c(1, 2, 3), props = c(10, 10, 5))

#to manufacture composition from a continuous measure, first translate into a
#categorical equivalent, e.g.
age <- pheno$age
pheno$age_group <- ifelse(age < 40, "<40", ifelse(age < 50, "40-50", ">=50"))
compSelect(pheno,
    "age_group",
    values = c("<40", "40-50", ">=50"),
    props = c(2, 5, 10))
```

hrSignificance *Calculate HR significance using bootstrap results*

Description

Calculate HR significance using bootstrap results

Usage

```
hrSignificance(hratios, bs_dfr)
```

Arguments

hratios	Hazard ratio vector, output of allHR()
bs_dfr	Dataframe of bootstrapped hazard ratios

Value

A per-separation point vector equal to the distance beyond or within the bootstrapped thresholds

nki_subset *NKI breast cancer patients subset with complete t.dmfs and e.dmfs information*

Description

An ExpressionSet of NKI breast cancer patients detailing microarray assay and phenotypic (including survival) information. The data is restricted to the 500 most variable genes only.

Usage

```
nki_subset
```

Format

An ExpressionSet including assayData with 319 columns (samples), 500 rows (features) and phenoData with 5 columns (variables)

Details

A subset of the breastCancerNKI Bioconductor package - for more details see:

http://bioconductor.org/packages/release/data/experiment/html/ breastCancerNKI.html

Source

http://bioconductor.org/packages/release/data/experiment/html/ breastCancerNKI.html

plotALL *Calculate and combine hazard ratio, pvalue, threshold and area-between-curve data and plot*

Description

Calculate and combine hazard ratio, pvalue, threshold and area-between-curve data and plot

Usage

```
plotALL(measure, srv, time = "Time", event = "Event", bs_dfr = c(),
  measure_name = "measure", multiv = NULL, title = "")
```

Arguments

measure	A continuous variable used to order survival data. Samples must be ordered exactly as in srv
srv	A dataframe that contains at least two columns, detailing event and time to event information. Samples must be ordered exactly as in measure
time	Column name in srv containing time to event information. Must not contain NAs
event	Column name in srv containing event information coded as 0 (no event) and 1 (event). Must not contain NAs
bs_dfr	A matrix of bootstrapped hazard ratio computations as ordered by a random measurement vector. Typically consisting of 5-10,000 repeat samplings
measure_name	A descriptive name for the measure used, for example a gene ID
multiv	Univariate analysis is performed by default, however a character string specifying a column contained in srv (or a vector of strings specifying multiple columns) detailing additional variables can be included
title	Plot title; as a character string

Value

Using survival, measure, hazard ratio, pvalue, log10 pvalue, threshold and threshold residual information, plot the measure-event relationship

Examples

```
data(nki_subset)
library(Biobase)

gene_vec <- exprs(nki_subset)["NM_004448", ] #ERBB2 gene id

plotALL(measure = gene_vec,
    srv = pData(nki_subset),
    time = "t.dmfs",
```

```
  event = "e.dmfs",
  title = "ERBB2 Example")
```

`removeOutliers` *Calculate outliers in a numeric vector and then convert those values to NA*

Description

Calculate outliers in a numeric vector and then convert those values to NA

Usage

```
removeOutliers(x, tolerant = TRUE)
```

Arguments

`x`	A numeric vector
`tolerant`	Outlier calculation tolerance. A more tolerant outlier removal is more appropriate when working with hazard ratios

Value

The modified, outlier removed, equivalent of x

Examples

```
set.seed(123); x <- rnorm(100)
sum(is.na(x))
y <- removeOutliers(x)
sum(is.na(y))
```

`survivALL` *Calculate and combine hazard ratio, pvalue, threshold and area-between-curve data as a single dataframe*

Description

Calculate and combine hazard ratio, pvalue, threshold and area-between-curve data as a single dataframe

Usage

```
survivALL(measure, srv, time = "Time", event = "Event", bs_dfr = c(),
  measure_name = "measure", multiv = NULL, n_sd = 1.96,
  remove_outliers = TRUE)
```

Arguments

measure	A continuous variable used to order survival data. Samples must be ordered exactly as in srv
srv	A dataframe that contains at least two columns, detailing event and time to event information. Samples must be ordered exactly as in measure
time	Column name in srv containing time to event information. Must not contain NAs
event	Column name in srv containing event information coded as 0 (no event) and 1 (event). Must not contain NAs
bs_dfr	A matrix of bootstrapped hazard ratio computations as ordered by a random measurement vector. Typically consisting of 5-10,000 repeat samplings
measure_name	A descriptive name for the measure used, for example a gene ID
multiv	Univariate analysis is performed by default, however a character string specifying a column contained in srv (or a vector of strings specifying multiple columns) detailing additional variables can be included
n_sd	The number of standard deviations used to define threshold width. 95 deviation of 1.96
remove_outliers	Large hazard ratios result from statistical disproportion when considering edge cases (e.g. 1 vs 99) and can be automaticall removed

Value

a dataframe detailing survival, measure, hazard ratio, pvalue, log10 pvalue, threshold and threshold residual information

Examples

```
library(survivALL)
data(nki_subset)
library(Biobase)
library(ggplot2)

gene_vec <- exprs(nki_subset)["NM_004448", ] #ERBB2 gene id

survivALL_dfr <- survivALL(measure = gene_vec,
   srv = pData(nki_subset),
   time = "t.dmfs",
   event = "e.dmfs")

ggplot(survivALL_dfr, aes_string(x = 'measure', y = 'p')) +
    geom_hline(yintercept = 0.05, linetype = 3) +
    geom_point()
```

Index