

Package ‘RTNsurvival’

April 12, 2018

Type Package

Title Survival analysis using transcriptional networks inferred by the RTN package

Version 1.2.0

Author Clarice S. Groeneveld, Vinicius S. Chagas, Mauro A. A. Castro

Maintainer Clarice Groeneveld <clari.groeneveld@gmail.com>, Mauro A. A. Castro <mauro.a.castro@gmail.com>

Depends R (>= 3.3.3), methods, RTN

Imports survival, RColorBrewer, grDevices, graphics, stats, RTNduals, png, utils

Suggests Fletcher2013b, pheatmap, knitr, rmarkdown, BiocStyle, RUnit, BiocGenerics

Description RTNsurvival is a tool for integrating regulons generated by the RTN package with survival information. For a given regulon, the 2-tailed GSEA approach computes a differential Enrichment Score (dES) for each individual sample, and the dES distribution of all samples is then used to assess the survival statistics for the cohort. There are two main survival analysis workflows: a Cox Proportional Hazards approach used to model regulons as predictors of survival time, and a Kaplan-Meier analysis assessing the stratification of a cohort based on the regulon activity. All plots can be fine-tuned to the user's specifications.

License Artistic-2.0

biocViews NetworkEnrichment, Survival, GeneRegulation, GeneSetEnrichment, NetworkInference, GraphAndNetwork

LazyData TRUE

VignetteBuilder knitr

RoxygenNote 6.0.1

NeedsCompilation no

R topics documented:

RTNsurvival-package	2
dualCoxTable	3
dualSurvivalPanel,MBR-method	4
TNS-class	6
TNS.data	6
tnsCox,TNS-method	7

tnsGet,TNS-method	8
tnsGSEA2,TNS-method	9
tnsKM,TNS-method	10
tnsPlotGSEA2	11
tnsPreprocess,TNI-method	12

Index	14
--------------	-----------

RTNsurvival-package	<i>Performs survival analysis using transcriptional networks inferred by the RTN package.</i>
---------------------	---

Description

This package provides classes and methods to perform survival analysis using transcriptional networks inferred by the RTN package, including Kaplan-Meier and multivariate survival analysis using Cox's regression model.

Details

Package: RTNsurvival
 Type: Package
 Depends: R (>= 3.3.0), methods
 Imports: RTN, survival, RColorBrewer
 Suggests: Fletcher2013b, knitr, rmarkdown, BiocStyle, RUnit, BiocGenerics
 License: Artistic-2.0
 biocViews: NetworkInference, NetworkEnrichment, GeneRegulation, GeneExpression, GraphAndNetwork, GeneSetEn

Index

TNS-class: an S4 class for survival survival analysis using RTN transcriptional networks.
tnsPreprocess: a preprocessing method for objects of class TNS.
tnsGSEA2: two-tailed gene set enrichment analysis (GSEA) over a list of regulons.
tnsPlotGSEA2: plot results from the two-tailed GSEA.
tnsKM: Kaplan-Meier analysis for TNS class objects.
tnsCox: Cox regression analysis for TNS class objects.

Further information is available in the vignettes by typing `vignette("RTNsurvival")`. Documented topics are also available in HTML by typing `help.start()` and selecting the RTNsurvival package from the menu.

Author(s)

Clarice S. Groeneveld, Vinicius S. Chagas, Gordon Robertson, ..., Kerstin Meyer, Mauro A. A. Castro

References

Fletcher M.N.C. et al., *Master regulators of FGFR2 signalling and breast cancer risk*. Nature Communications, 4:2464, 2013.

Castro M.A.A. et al., *Regulators of genetic risk of breast cancer identified by integrative network analysis*. Nature Genetics, 48:12-21, 2016.

dualCoxTable	<i>Cox regression analysis for dual regulons</i>
--------------	--

Description

Returns a data.frame with Cox regression results for the interaction between regulons identified as duals.

Usage

```
dualCoxTable(mbr, tns1, tns2 = NULL, duals = NULL, verbose = TRUE,
             excludeMid = FALSE)
```

Arguments

mbr	an MBR object computed from the same TNI objects that were used to make the TNS objects.
tns1	a 'TNS' object with regulons used to compute the duals.
tns2	another 'TNS' object computed with the other regulons used to compute the duals. It's optional if all duals are from the same 'TNI' object.
duals	an optional vector containing duals to create the table. If NULL, the regression is performed for all duals.
verbose	a logical value. If TRUE, prints status of function while executing.
excludeMid	if TRUE, doesn't use the samples that fall in mid section of the stratification for computations.

Value

A matrix containing the Cox regression results for all given duals

See Also

[tni2mbrPreprocess](#) for all plot parameters

Examples

```
# load survival data
data(survival.data)

# load TNI-object
data(stni, package = "RTN")
stni <- upgradeTNI(stni)

# perform survival analysis
```

```

stns <- tnsPreprocess(stni, survival.data, keycovar = c('Grade','Age'),
                    time = 1, event = 2)
stns <- tnsGSEA2(stns, verbose=FALSE)

# create MBR-object using TF-TF duals
library(RTNduals)
tni <- tnsGet(stns, "TNI")
mbr <- tni2mbrPreprocess(tni, tni, verbose = FALSE)
mbr <- mbrAssociation(mbr, prob = 0.75)
mbr <- mbrDuals(mbr)
results <- mbrGet(mbr, what="dualsInformation")

# create dual Cox regression
dualCox <- dualCoxTable(mbr, stns, verbose = FALSE)

```

dualSurvivalPanel,MBR-method

Panel plot of regulons in a inferred dual

Description

Returns a folder with 5 plots that form a panel: [1] sample stratification based on activity of regulon 1, [2] sample stratification based on activity of regulon 2, [3] a scatter plot of samples's rank in both stratifications based on regulon activity [4] Kaplan-Meier curves based on both regulon stratifications and interaction of regulons in dual, [5] Cox regression for both regulons and the interaction between the activity information for the dual. The function also returns a PNG panel with all plots aligned.

Usage

```

## S4 method for signature 'MBR'
dualSurvivalPanel(mbr, tns1, tns2 = NULL, dual,
  attribs = NULL, endpoint = 60, path = NULL, nSections = 2,
  pal = "BrBG", excludeMid = FALSE, sectionsLegend = NULL,
  panelWidths = c(2, 3), png.res = 150, attribs.cex = 1)

```

Arguments

mbr	an MBR object computed from the same TNI objects that were used to make the TNS objects.
tns1	a 'TNS' object with regulons used to compute the duals.
tns2	another 'TNS' object computed with the other regulons used to compute the duals. It's optional if all duals are from the same 'TNI' object.
dual	a character string with the name of a dual
attribs	a numeric vector. Contains the columns of the survival data.frame which will be plotted as sample status in first and second plots.
endpoint	a numeric value. It represents the cut-off point for the 'time', if any.
path	a string. The path to the directory where the plot will be saved.

nSections	numeric value for the stratification of the sample. The larger the number, the more subdivisions will be created for the Kaplan-Meier and sample stratification analysis.
pal	a palette name to be passed to brewer.pal .
excludeMid	a logical value. if TRUE, doesn't use the samples that fall in mid section of the stratification for computations.
sectionsLegend	a string vector. It must contain names for the sections of the stratification on the interaction Kaplan-Meier. If NULL, the sections are simply numbered in the legend.
panelWidths	a numeric vector of length 2. It must contain a proportion for the two panels in plots [1] and [2].
png.res	a numeric value. It represents the resolution in ppi for the png panel.
attribs.cex	a numeric value. Represents the expansion factor for the attribs legend.

Value

A folder containing 5 pdf survival plots and one png survival panel with all plots

See Also

[tnsKM](#), [tnsCox](#)

Examples

```
# load survival data
data(survival.data)

# load TNI-object
data(stni, package = "RTN")
stni <- upgradeTNI(stni)

# perform survival analysis
stns <- tnsPreprocess(stni, survival.data, keycovar = c('Grade', 'Age'),
                    time = 1, event = 2)
stns <- tnsGSEA2(stns, verbose=FALSE)

# create MBR-object using TF-TF duals
library(RTNduals)
tni <- tnsGet(stns, "TNI")
mbr <- tni2mbrPreprocess(tni, tni, verbose = FALSE)
mbr <- mbrAssociation(mbr, prob = 0.75)
mbr <- mbrDuals(mbr)
duals <- mbrGet(mbr, what="dualRegulons")

# create panel plot

dualSurvivalPanel(mbr, stns, dual = duals[1],
                 attribs = c("ER+", "ER-", "PR+", "PR-"))
```

TNS-class	<i>TNS: An S4 class for survival survival analysis using transcriptional networks inferred by the RTN package.</i>
-----------	--

Description

TNS: An S4 class for survival survival analysis using transcriptional networks inferred by the RTN package.

Slots

`survivalData` a data frame containing the survival data for all samples. Samples must be in the rows and the survival variables in the columns. Time of last update and event in last update (0 for alive, 1 for deceased).

`EScores` a list created by `tnsGSEA2`, containing the enrichment scores for all samples

`keycovar` a string vector containing the key covariables used to compute the Cox regression. They must be present in the `survivalData` table.

`tni` a [TNI](#)-class object, previously computed. It is added to the TNS via `tnsGSEA2`

`status` a vector containing the processing status of the TNS object.

`para` a list with the parameters used to compute the GSEA2 analysis.

Constructor

`tnsPreprocess(tni, survivalData, keycovar, time = 1, event = 2, samples = NULL)`

- `tni` - a 'TNI' object.
- `survivalData` - a data.frame containing at least 2 columns
- `keycovar` - A character vector or NULL
- `time` - A numeric value or character vector
- `event` - A numeric value or character vector
- `samples` - A character vector

TNS.data	<i>A pre-processed dataset for demonstration purposes only.</i>
----------	---

Description

A minimum dataset used to demonstrate RTNsurvival main features.

Usage

```
data(survival.data)
```

Format

`survival.data` A data.frame with a subset of samples in the Fletcher2013b package.

Details

The dataset consists of data.frame with survival and clinical variables used in the RTNsurvival vignettes. It should be regarded as a toy example for demonstration purposes only, despite being extracted, pre-processed and size-reduced from Fletcher et al. (2013) and Curtis et al. (2012).

Value

a data.frame.

References

Fletcher M.N.C. et al., *Master regulators of FGFR2 signalling and breast cancer risk*. Nature Communications, 4:2464, 2013.

Curtis C. et al., *The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups*. Nature 486, 7403. 2012.

Examples

```
data(survival.data)
```

tnsCox, TNS-method	<i>Cox regression analysis for TNS class objects.</i>
--------------------	---

Description

Run Cox multivariate regression for regulons and key covariables.

Usage

```
## S4 method for signature 'TNS'
tnsCox(tns, regs = NULL, endpoint = 60, fname = "coxplot",
      fpath = ".", ylab = "Regulons and key covariates",
      xlab = "Hazard Ratio (95% CI)", qqkeycovar = FALSE, excludeMid = FALSE,
      width = 5, height = 5, xlim = c(0.2, 10), sortregs = TRUE,
      plotpdf = TRUE)
```

Arguments

tns	a TNS object, which must have passed GSEA2 analysis.
regs	a string vector. Contains the regulons which will be used to compute the Cox multivariate model. If left NULL, all regulons will be used.
endpoint	a numeric value. The final point in time for the samples. All time values larger than endpoint will be set at endpoint.
fname	a string. The name of the PDF file which will contain the plot.
fpath	a string. The directory where the file will be saved.
ylab	a string. The label of the y-axis, describing what is represented.
xlab	a string. The label of the x-axis.
qqkeycovar	a logical value. If TRUE, only the samples in the 2nd and 3rd quarters of dES are used to compute. If FALSE, all samples are used.

excludeMid	a logical value. If TRUE, inconclusive dES values will not be considered in the survival analysis.
width	a numeric value. The width of the plot.
height	a numeric value. The height of the plot.
xlim	a vector with 2 values. The first value represents the lowest value in the x-axis, the second value is the highest.
sortregs	a logical value. If TRUE, regulons are sorted from most negatively associated with hazard to most positively associated with hazard.
plotpdf	a logical value.

Value

A Cox hazard model plot. If TRUE, generates a pdf plot.

Examples

```
# load survival data
data(survival.data)

# load TNI-object
data(stni, package = "RTN")

stns <- tnsPreprocess(stni, survival.data, keycovar = c('Grade', 'Age'),
time = 1, event = 2)
stns <- tnsGSEA2(stns, verbose=FALSE)
tnsCox(stns, regs = c('PTTG1', 'E2F2', 'FOX1'), sortregs = TRUE,
plotpdf = FALSE)
```

tnsGet, TNS-method	<i>Get information from slots in a TNS object</i>
--------------------	---

Description

Get information from individual slots in a TNS object and any available results from a previous analysis.

Usage

```
## S4 method for signature 'TNS'
tnsGet(object, what)
```

Arguments

object	a TNS object
what	a character vector specifying what should be retrieved from the object. Options: 'survivalData', 'EScores', 'TNI', 'keycovar'

Value

A plot containing the 2-tailed GSEA analysis for a phenotype.

Examples

```
# load survival data
data(survival.data)

# load TNI-object
data(stni, package = "RTN")

stns <- tnsPreprocess(stni, survival.data, keycovar = c('Grade', 'Age'),
time = 1, event = 2)
stns <- tnsGSEA2(stns, verbose=FALSE)
enrichmentScores <- tnsGet(stns, 'EScores')
```

tnsGSEA2, TNS-method *2-tailed Gene Set Enrichment Analysis on Transcriptional Networks.*

Description

Works as a wrapper for [tni.gsea2](#), performing a 2-tailed GSEA analysis on a [TNI](#) class object and integrating the results into the [TNS](#) class object.

Usage

```
## S4 method for signature 'TNS'
tnsGSEA2(tns, ...)
```

Arguments

tns	A TNS class, which has been preprocessed
...	Parameters passed to the tni.gsea2 function.

Value

A [TNS](#) class, with added Enrichment Scores.

See Also

[tni.gsea2](#) for information on all parameters.

Examples

```
# load survival data
data(survival.data)

# load TNI-object
data(stni, package = "RTN")

stns <- tnsPreprocess(stni, survival.data, keycovar = c('Grade', 'Age'), time = 1, event = 2)
stns <- tnsGSEA2(stns, verbose=FALSE)
```

tnsKM, TNS-method

*Kaplan-Meier analysis for TNS class objects.***Description**

Makes a 2 or 3 panel plot for survival analysis. The first panel shows the differential Enrichment score (dES) for all samples, ranked by expression in their sections. The second (optional) panel shows the status of other attributes which may be present in the survival data.frame for all samples. The third panel shows a Kaplan-Meier plot computed for the given survival data, with a curve for each section.

Usage

```
## S4 method for signature 'TNS'
tnsKM(tns, regs = NULL, attribs = NULL, nSections = 2,
      endpoint = 60, fname = "survplot", fpath = ".",
      ylab = "Survival probability", xlab = "Months", pal = "redblue",
      excludeMid = FALSE, flipcols = FALSE, plotpdf = TRUE,
      plotbatch = FALSE, width = 6.3, height = 3.6, panelWidths = c(3, 2,
      4), dES.ylab = "Samples", show.KMlegend = TRUE,
      KMlegend.pos = "bottomleft", KMlegend.cex = 1, show.pval = TRUE,
      pval.cex = 1, pval.pos = "topright")
```

Arguments

tns	a TNS object, which must have passed GSEA2 analysis.
regs	a string vector. Contains all the regulons which are going to be plotted.
attribs	a numeric vector. Contains the columns of the survival data.frame which will be plotted for the second panel.
nSections	A numeric value for the stratification of the sample. The larger the number, the more subdivisions will be created for the Kaplan-Meier analysis.
endpoint	a numeric value. It represents the cut-off point for the 'time', if any.
fname	a string. The name of the file in which the plot will be saved
fpath	a string. The path to the directory where the plot will be saved
ylab	a string. The label for the y axis on the third panel
xlab	a string. The label for the x axis on the third panel. This should be the measure of time shown in the survival data.frame after the last check-up.
pal	a string, which can be 'red', 'blue' or 'redblue'. Represents the colors used in the first and third panels. Alternatively, it can contains the hex values.
excludeMid	a logical value. If TRUE, inconclusive dES values will not be consired in the survival analysis.
flipcols	a logical value. If TRUE, flips the order of the samples to lowest expression on top, highest on the bottom.
plotpdf	a logical value. If TRUE, the plot is saved as a pdf file. If false, it is plotted in the plotting area.
plotbatch	a logical value. If TRUE, plots for all regs are saved in the same file. If FALSE, each plot for each reg is saved in a different file.

<code>width</code>	a numeric value. Represents the width of the plot.
<code>height</code>	a numeric value. Represents the height of the plot.
<code>panelWidths</code>	a numeric vector of length=3 specifying the relative width of the internal panels.
<code>dES.ylab</code>	a string. The label for the y axis of the first panel.
<code>show.KMlegend</code>	a logical value. If TRUE, shows the sample stratification information on the third panel.
<code>KMlegend.pos</code>	a string. Provides the location of the sample stratification legend on the third panel. One of: 'bottomright', 'bottom', 'bottomleft', 'left', 'topleft', 'top', 'topright', 'right' and 'center'.
<code>KMlegend.cex</code>	a numeric value. Provides the character expansion factor for the stratification legend, which alters the size and spacing of the font in the legend.
<code>show.pval</code>	a logical value. If TRUE, shows the Logrank P-value for the Kaplan-Meier plot on the third panel.
<code>pval.cex</code>	a numeric value. Provides the character expansion factor for the pvalue legend, which alters the size and spacing of the font in the legend.
<code>pval.pos</code>	a string. Provides the location of the Logrank P-value on the third panel. One of: 'bottomright', 'bottom', 'bottomleft', 'left', 'topleft', 'top', 'topright', 'right' and 'center'.

Value

A plot, showing the graphical analysis of provided survival data.

Examples

```
# load survival data
data(survival.data)

# load TNI-object
data(stni, package = "RTN")

stns <- tnsPreprocess(stni, survival.data, keycovar = c('Grade', 'Age'),
  time = 1, event = 2)
stns <- tnsGSEA2(stns, verbose=FALSE)
tnsKM(stns, regs='FOXM1', attribs = list(c('ER+', 'ER-'), c('G1', 'G2', 'G3'))),
  plotpdf = FALSE)
```

tnsPlotGSEA2

Plot 2-tailed GSEA for a sample from a TNS

Description

Makes a 2-tailed GSEA plot for a certain phenotype (sample) present in a TNS. A wrapper of [tna.plot.gsea2](#)

Usage

```
tnsPlotGSEA2(object, aSample, regs = NULL, refsamp = NULL, log = FALSE,
  ntop = NULL, pValueCutoff = 0.05, pAdjustMethod = "BH",
  verbose = TRUE, plotpdf = TRUE, ...)
```

Arguments

object	a TNS object
aSample	a string specifying a given sample number present in the 'survivalData' table.
regs	an optional string vector specifying regulons to make the plot.
refsamp	a character vector.
log	a logical value. If TRUE, gexp values are transformed into log space.
ntop	an optional integer value. The number of regulons for which the GSEA2 will be plotted.
pValueCutoff	an integer. The p cutoff value for the analysis.
pAdjustMethod	a character. Specifies the adjustment method for the pvalue. See p.adjust
verbose	a logical value. If TRUE, prints status of function while executing.
plotpdf	a single logical value.
...	parameters which will be passed to tna.plot.gsea2 , such as ylimPanels, heightPanels, width, height, ylabPanels, xlab...

Value

A plot containing the 2-tailed GSEA analysis for a phenotype.

See Also

[tna.plot.gsea2](#) for all plot parameters

Examples

```
# load survival data
data(survival.data)

# load TNI-object
data(stni, package = "RTN")

stns <- tnsPreprocess(stni, survival.data, keycovar = c('Grade', 'Age'),
  time = 1, event = 2)
stns <- tnsGSEA2(stns, verbose=FALSE)
tnsPlotGSEA2(stns, 'MB-5115', regs = 'FOXM1')
```

tnsPreprocess, TNI-method

Preprocessing of TNS class objects.

Description

Creates TNS class onbjects for regulons an survival data.

Usage

```
## S4 method for signature 'TNI'
tnsPreprocess(tni, survivalData, keycovar = NULL, time = 1,
  event = 2, samples = NULL)
```

Arguments

tni	A TNI class, already processed with the same samples listed in the survival data.frame.
survivalData	A named data.frame with samples in rows and survival data in the columns.
keycovar	A character vector of the 'keycovars' listed in the data.frame columns.
time	A numeric or character value corresponding to the column of the data.frame where the time of last observation is given.
event	A numeric or character value, corresponding to the column of the data.frame where the 'event' information is given.
samples	An optional character vector listing samples to be analyzed.

Value

A preprocessed [TNS](#) class

See Also

[tni.preprocess](#) for similar preprocessing.

Examples

```
# load survival data
data(survival.data)

# load TNI-object
data(stni, package = "RTN")

# create a new TNS object
stns <- tnsPreprocess(stni, survival.data, keycovar = c('Grade', 'Age'),
time = 1, event = 2)
```

Index

*Topic **dataset**

TNS.data, 6

*Topic **package**

RTNsurvival-package, 2

brewer.pal, 5

dualCoxTable, 3

dualSurvivalPanel

(dualSurvivalPanel, MBR-method),

4

dualSurvivalPanel, MBR-method, 4

MBR, 3, 4

p.adjust, 12

RTNsurvival (RTNsurvival-package), 2

RTNsurvival-package, 2

survival.data (TNS.data), 6

tna.plot.gsea2, 11, 12

TNI, 3, 4, 6, 9, 13

tni.gsea2, 9

tni.preprocess, 13

tni2mbrPreprocess, 3

TNS, 3, 4, 7, 9, 10, 13

TNS (TNS-class), 6

TNS-class, 2, 6

TNS.data, 6

tnsCox, 2, 5

tnsCox (tnsCox, TNS-method), 7

tnsCox, TNS-method, 7

tnsGet (tnsGet, TNS-method), 8

tnsGet, TNS-method, 8

tnsGSEA2, 2

tnsGSEA2 (tnsGSEA2, TNS-method), 9

tnsGSEA2, TNS-method, 9

tnsKM, 2, 5

tnsKM (tnsKM, TNS-method), 10

tnsKM, TNS-method, 10

tnsPlotGSEA2, 2, 11

tnsPreprocess, 2

tnsPreprocess

(tnsPreprocess, TNI-method), 12

tnsPreprocess, TNI-method, 12