Package 'TSIS'

January 3, 2017

Type Package

Title Time-series isoform switch of alternative splicing

Version 0.1.0

Date 2016-11-10

Author Wenbin Guo <wenbin.guo@hutton.ac.uk>

Maintainer Wenbin Guo <wenbin.guo@hutton.ac.uk>

Description Alternative splicing (AS) is an essential process in plant and animal genomes, which regulates protein abundance and increases the protein biodiversity. In the AS regulation, one of the cases is a pair of isoforms from the same gene reverse their relative expression abundance (usage) at some stress reaction points, denoting as isoform switch event. Isoform switches often play pivotal roles in re-programming of gene regulation in response to biotic and abiotic stimuli, for instance the transcript isoform degradation of nonsense-mediated decay in a surveillance pathway. This package is used to determine, score and visualize the isoform switch patterns in isoform expression time-series data. All the analyses have been integrated into a Shiny App, in which users can implement analysis as easy as mouse click.

License GPL-3

LazyData TRUE

Depends R (>= 3.2.3)

RoxygenNote 5.0.1

Suggests BiocStyle, knitr, rmarkdown

VignetteBuilder knitr

R topics documented:

`AtRTD2` *Example datasets of AtRTD2.*

Description

The data sources include subset of TPM time-series transcript expression data with 26 time points, 3 biological replicates and 3 technical replicates (234 sample points in total) from the AtRTD2 paper (Zhang, et al., 2016), a gene-isoform mapping table of 300 genes (first column) and 766 transcripts (second column) and a subset of 100 isoforms.

Usage

```
data(AtRTD2)
```

Format

An object of data list

References

Zhang, R., et al. AtRTD2: A Reference Transcript Dataset for accurate quantification of alternative splicing and expression changes in Arabidopsis thaliana RNA-seq data. bioRxiv 2016.

Examples

```
names(AtRTD2)
##expression data
AtRTD2$data.exp[1:10,1:5]
##mapping data
AtRTD2$mapping[1:10,]
##subset of isoform names
AtRTD2$sub.isoforms[1:10]
```

`AtRTD2.example` *Save AtRTD2 data into csv files*

Description

Save AtRTD2 data into csv files

Usage

```
AtRTD2.example(dir = "data")
```

Arguments

`dir`	directory to save data. If the directory does not exist, a new folder will be created with the provided name.

Value

Three csv tables of corresponding data in AtRTD2.

Examples

```
AtRTD2.example(dir='data')
```

`data.error` *Calculate standard deviation or standard errors*

Description

Calculate standard deviation or standard errors from a numerical vector.

Usage

```
data.error(x, error.type = "stderr")
```

Arguments

`x`	a numerical vector
`error.type`	method used to calculate data error. Options include "sd" for standard deviation and "stderr" for standard error.

Value

data error

See Also

`sd`

Examples

```
set.seed(2000)
x=rnorm(1000,mean=0,sd=1)
##standard error
data.error(x,error.type = 'stderr')
##standard deviation
data.error(x,error.type = 'sd')
```

`iso.switch` *Isoform switch analysis for time-series data*

Description

This function is used to search and score isoform switch points in time-series isoform expression data.

Usage

```
iso.switch(data.exp, mapping, t.start = 1, t.end = 26, nrep = 9,
  rank = F, min.t.points = 2, min.distance = 1, spline = F,
  spline.df = NULL, verbose = T)
```

Arguments

`data.exp`	isoform expression data frame with row names of isoforms and column names of samples.
`mapping`	gene-isoform mapping data frame with first column of genes and second column of isoforms.
`t.start, t.end`	start and end time points of the time-series data. The time step is assumed to be 1.
`nrep`	number of replicates for each time point.
`rank`	logical, to use rank of isoform expression for each sample (TRUE) or not (FALSE).
`min.t.points`	pre-filtering, if the number of time points in all intervals < `min.t.points`, skip this pair of isoforms.
`min.distance`	pre-filtering, if the sample distances in the time courses (mean expression or splined value) for intersection search all < `min.distance`, skip this pair of isoforms.
`spline`	logical, to use spline method (TRUE) or mean expression (FALSE).
`spline.df`	the degree of freedom used in spline method. See `ns` in `splines` for details.
`verbose`	logical, to track the progressing of runing (TRUE) or not (FALSE).

Details

The detailed steps:

Step 1: search for time course intersection points.

The expression for a pair of isoforms iso_i and iso_j may experience a number isoform switch in the whole time duration. Two methods have been included to search for these switch points where the isoforms reverse relative expression profiles.

- **Method 1:** use average expression values across time points. Taking average values of the replicates for time points in the input isoform expression data.
- **Method 2:** use nature spline curves to fit the time-series data and find intersection points of the fitted curves for each pair of isoforms. See details in `ts.spline`

Step 2: score the isoform switches

We defined 5 parameters to score the quality of isoform switch. The first two are the frequency/probability of switch and the sum of average distance before and after switch, used as Score 1 and Score 2 in iso-kTSP https://bitbucket.org/regulatorygenomicsupf/iso-ktsp method for two condition comparisons (Sebestyen, et al., 2015). To investigate the switches of two isoforms iso_i and iso_j in two conditions c_1 and c_2, **Score 1** is defined as

$$S_1(iso_i, iso_j|c_1, c_2) = |p(iso_1 > iso2|c_1) + p(iso_1 < iso_2|c_2) - 1|$$

where $p(iso_1 > iso2|c_1)$ and $p(iso_1 < iso_2|c_2)$ are the frequencies/probabilities that the samples of one isoform is greater or less than the other in corresponding conditions. **Score 2** is defined as

$$S_2(iso_i, iso_j|c_1, c_2) = |mean.dist(iso_i, iso_2|c_1)| + |mean.dist(ios_1, iso_2|c_2)|$$

where $mean.dist(iso_i, iso_2|c_1)$ and $mean.dist(ios_1, iso_2|c_2)$ are the mean distances of samples in conditions c_1 and c_2, respectively.

However, the time-series for a pair of isoforms may undergo a number of switches in the time duration. The time duration is divided into intervals with the intersection points determined in Step 1. To extend the iso-kTSP to TSIS, the samples in each pair of consecutive intervals before and after switch are assimilated as samples in two conditions to implement the calculation of Score 1 and Score 2.

The time-series isoform switches are more complex than the comparisons over two conditions. In addition to Score 1 and Score 2 for each switch point, we defined other 3 parameters as metrics of switch qualities.

- **Score 3:** p-value of paired t-test for the two isoform sample differences within each interval.
- **Score 4:** Time point number within each interval.
- **Score 5:** Pearson correlation of two isoforms.

Note: since each switch point has a left and right adjoined intervals before and after switch, two p-values and two numbers of time points for the intervals are assigned to each switch point, respectively.

Value

a data frame of scores. The column names:

- iso1,iso2: the isoforms.
- iso1.mean.ratio, iso2.mean.ratio: the ratio of isoforms to their genes.
- left.interval, left.interval: the left (before switch) and right (after switch) intervals of a switch point.
- x.value, y.value: x and y coordinates of switch points.
- left.prob, right.prob: the frequencies/probabilities that the samples of a isoform is greater or less than the other in left and right intervals, respectively.
- left.dist, right.dist: the average sample distances in left and right intervals, respectively.
- left.pval, right.pval: p-values of paired t-test for samples in left and right intervals, respectively.
- left.t.points, right.t.points: number of time points in left and right intervals, respectively.
- prob: Score1.
- dist: Score2.
- cor: Pearson correlation of two isoforms.

References

1. Sebestyen E, Zawisza M, Eyras E: Detection of recurrent alternative splicing switches in tumor samples reveals novel signatures of cancer. Nucleic Acids Res 2015, 43(3):1345-1356.

`plotTSIS` *Plot time-series switches of a pair of isoforms*

Description

Plot time-series switches of a pair of isoforms

Usage

```
plotTSIS(data2plot, scores = NULL, iso1 = NULL, iso2 = NULL,
  gene.name = NULL, y.lab = "Expression", make.plotly = F, t.start = 1,
  t.end = 26, nrep = 9, prob.cutoff = 0.5, x.lower.boundary = 9,
  x.upper.boundary = 17, show.region = T, show.scores = T,
  error.type = "stderr", show.errorbar = T, errorbar.width = 0.2,
  errorbar.size = 0.5, line.width = 1, point.size = 3, spline = F,
  spline.df = NULL, ribbon.plot = F)
```

Arguments

`data2plot`	the expession data frame used to plot the isoforms. The row names of the data frame are the isoforms to plot and the column names are time points and the replicates of time points.The column ordering should be for example t1.replicate1, t1.replicate2, ..., t2.replicate1, t2.replicate2, ...
`scores`	the output scores of functions `iso.switch` and `score.filter`. Default is NULL. To show score labels on the plot, the scores must be provided.
`iso1, iso2`	character string names of the first and second isoforms to plot. If `iso1` and `iso2` are NULL, the input `data2plot` must be a data frame of two rows and the row names are used as isoforms to plot.
`gene.name`	a character string of gene name to show as the title of the plot. If `gene.name=NULL`, the plot title will be "iso1_vs_iso2".
`y.lab`	the y label of the plot, default is "Expression".
`make.plotly`	logical, to plot `plotly` format figures (TRUE) or plain ggplot2 format figures(FALSE)?. See details in `ggplotly` in `plotly` R pacakge.
`t.start, t.end`	start and end time points of the time-series data. The time step is assumed to be 1.
`nrep`	number of replicates.
`prob.cutoff`	the cut-off of switch frequencies/probabilities to label the switch points.
`x.lower.boundary, x.upper.boundary`	the lower and upper boundary of x axis (time points) for the region under investigation. In the analysis, if the isoform pairs have no intersection points in this region, they are filtered.
`show.region`	logical, to highlight the region under investigation (TRUE) or not (FALSE)?
`show.scores`	logical, to show score labels on the plot (TRUE) or not (FALSE)? The `scores` object must be provided.

`error.type` the error type used to show the error bar in the plots. Options are "stderr" for standard error and "sd" for standard deviation. See details in `data.error`.

`show.errorbar` logical, to show error bar (TRUE) or not (FALSE) in the error bar plot.

`errorbar.width, errorbar.size`
the width and size of error bars. See detials in `geom_errorbar` in `ggplot2` R pacakge.

`line.width, point.size`
line width and point marker size of the plots.

`spline` logical, to plot the spline smoothed lines (TRUE) or the lines of mean expression (FALSE).

`spline.df` the degree of freedom used for spline. The value must be the same as in the function `iso.switch`. See spline details in `geom_smooth` in `ggplot2` and `ns` in `splines`.

`ribbon.plot` logcial, to make ribbon plot (TRUE) or error bar plot (FALSE). See ribbon plot details in `geom_smooth` in `ggplot2` R pacakge.

Value

a ggplot2 or `ggplotly` (if `plotly=TRUE`) plot.

See Also

`ggplotly`, `iso.switch`, `score.filter`, `data.error`, `geom_smooth`, `ns`

Examples

```
#generate random datasets
set.seed(100)
data2plot<-data.frame(matrix(abs(rnorm(60)),nrow=2),row.names = c('iso1','iso2'))
#error bar plot
plotTSIS(data2plot=data2plot,scores=NULL,iso1=NULL,iso2=NULL,gene.name=NULL,
      y.lab='Expression',make.plotly=F,t.start=1,t.end=10,nrep=3,prob.cutoff=0.5,x.lower.boundary=3,
        x.upper.boundary=8,show.region=T,error.type='stderr',ribbon.plot = F)

#ribbon plot
plotTSIS(data2plot=data2plot,scores=NULL,iso1=NULL,iso2=NULL,gene.name=NULL,
       y.lab='Expression',make.plotly=F,t.start=1,t.end=10,nrep=3,prob.cutoff=0.5,x.lower.boundary=3,
         x.upper.boundary=8,show.region=T,error.type='stderr',ribbon.plot = T)
```

---

rowmean *Calculate column mean of a matrix or data frame based on a grouping variable*

Description

Compute column means across rows of a numeric matrix-like object for each level of a grouping variable.

Usage

```
rowmean(x, group, reorder = T, na.rm = T)
```

Arguments

x	a matrix or data frame.
group	a vector of factor giving grouping, with one element per row of x.
reorder	if TRUE, then the result will be in order of sort(unique(group)).
na.rm	logical (TRUE or FALSE). Should NA (including NaN) values be replaced by value 0?

Value

rowmean returns a matrix or data frame containing the means. There will be one row per unique value of group.

See Also

rowsum, rowratio

Examples

```
x <- matrix(runif(50), ncol = 5)
group <- sample(1:4, 10, TRUE)
xmean <- rowmean(x, group)
```

---

rowratio *Calculate column raito of a matrix or data frame based on a grouping variable*

Description

Compute column ratio across rows of a numeric matrix-like object for each level of a grouping variable.

Usage

```
rowratio(x, group, reorder = T, na.rm = T)
```

Arguments

x	a matrix or data frame.
group	a vector of factor giving grouping, with one element per row of x.
reorder	if TRUE, then the result will be in order of row names of x. If row names of x is null, the results is not reordered.
na.rm	logical (TRUE or FALSE). Should NA (including NaN) values be replaced by value 0?

Value

rowratio returns a matrix or data frame containing the ratios. There will be one row per unique value of group.

See Also

rowsum, rowmean

Examples

```
x <- matrix(runif(50), ncol = 5)
group <- sample(1:4, 10, TRUE)
xratio <- rowratio(x, group)
```

---

score.filter *Filtering the scores for isoform switch*

Description

Filtering the scores output from iso.switch.

Usage

```
score.filter(scores, prob.cutoff = 0.5, dist.cutoff = 1,
  t.points.cutoff = 2, pval.cutoff = 0.01, cor.cutoff = 0.5,
  data.exp = NULL, mapping = NULL, sub.isoform.list = NULL,
  sub.isoform = F, max.ratio = F, x.value.limit = c(9, 17))
```

Arguments

scores	the scores output from iso.switch.
prob.cutoff, dist.cutoff, t.points.cutoff, pval.cutoff, cor.cutoff	the cut-offs corresponding to switch frequencies/probablities, sum of average distances, p-value and time points cut-offs for both intervals before and after switch and Pearson correlation.
data.exp, mapping	the expression and gene-isoform mapping data.
sub.isoform.list	a vector of isoform names to output subset of the corresponding results.
sub.isoform	logical, to output subset of the results(TRUE) or not (FALSE). If TRUE, sub.isoform.list must be provided.
max.ratio	logical, to show the subset of results with the isoforms of maximum ratios to the genes (TRUE) or not (FALSE). If TRUE, data.exp and mapping data must be provided to calculate the isoform ratios to the genes using rowratio.
x.value.limit	the region of x axis (time) for investigation. If there is no intersection point in this region, the isoform pair is filtered.

Details

A prospective isoform switch should be:

- Have high Score 1 of swtich frequency/probability.
- With proper value of Score 2 the sum of average distances.
- The samples in the intervals before and after switch are statistically different.
- The switch event lasting a few time points in both intervals before and after switch, i.e. the intervals should contain a number of time points.
- For further details, users can investigate the co-expressed isoform pairs with high Pearson correlation. Note: the isoform pairs with high negative correlation may show better switch pattern if look at the time-series plots.

Users may need to investigate subset of isoforms for specific purpose. Three options have been build-in the TSIS package.

- Users can set the lower and upper boundaries of a region in the time duration to study the switches only within this region.
- Users can provide a name list of isoforms to only show the results cantain the isoforms in the list.
- Users can output subset of results with highest ratios (the proportions of isoforms to the genes) isoforms.

`switch.density` *Density plot*

Description

Density plot

Usage

```
switch.density(x, make.plotly = T, ...)
```

Arguments

`x`	a numeric vector to plot th density.
`make.plotly`	logical, to plot `plotly` format figures (TRUE) or general plot (FALSE)?. See details in `ggplotly` in `plotly` R pacakge.
`...`	additional parameters pass to `density`

Value

density plot in general format or `plotly` format if `make.plotly=T`

Examples

```
##random values
set.seed(100)
x<-rnorm(100,mean=0,sd=1)
##geneal format
switch.density(x,make.plotly =F)
##plotly format
switch.density(x,make.plotly =T)
```

`ts.intersection` *Searching for intersection points of two time-series*

Description

Searching for intersection points of two time-series

Usage

```
ts.intersection(x1, x2)
```

Arguments

`x1`	a vector of values for first time-series
`x2`	a vector of values for second time-series

Value

a data frame of intersection points. The first and second columns are x and y coordinates values of intersections, respectively.

`ts.spline` *Fit a time-series with spline curve*

Description

Fit a time-series with spline curve

Usage

```
ts.spline(x, t.start, t.end, nrep, df = 5, ...)
```

Arguments

`x`	a vector of time-series expression
`t.start, t.end`	start and end time points of the time-series data. The time step is assumed to be 1.
`nrep`	number of replicates.
`df`	degree of freedom used in `ns` in `splines` package.
`...`	additional arguments passed to `predict`.

Value

predictions results returned from predict.

See Also

lm, ns, predict

TSIS.app *Isoform switch analysis and visualization with Shiny App*

Description

Isoform switch analysis and visualization with Shiny App

Usage

```
TSIS.app(data.size.max = 100)
```

Arguments

data.size.max	maximum size limit for unload files in Shiny. Default is 100 (MB).

Value

the Shiny App.

See Also

shiny

Examples

```
TSIS.app()
```

Index