Package ‘RNentropy’

September 28, 2026

Type Package

Title Entropy Based Method for the Detection of Significant Variation in Gene Expression Data

Version 1.3.3

Depends R (>= 2.10)

Date 2026-09-28

Maintainer Federico Zambelli <federico.zambelli@unimi.it>

Description An implementation of a method based on information theory devised for the identification of genes showing a significant variation of expression across multiple conditions. Given expression estimates from any number of RNA-Seq samples and conditions it identifies genes or transcripts with a significant variation of expression across all the conditions studied, together with the samples in which they are over- or under-expressed. It also detects genes whose relative isoform usage changes across samples (isoform switching).
Zambelli et al. (2018) <doi:10.1093/nar/gky055>.

License GPL-3

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Federico Zambelli [aut, cre] (ORCID: <https://orcid.org/0000-0003-3487-4331>), Giulio Pavesi [aut] (ORCID: <https://orcid.org/0000-0001-5705-6249>)

Repository CRAN

Date/Publication 2026-09-28 18:10:08 UTC

Contents

RNentropy-package 2
RNentropy 3
RNentropy_iso_switch 5

RN_BarresLab_design 6
RN_BarresLab_FPKM 7
RN_Brain_Example_design 8
RN_Brain_Example_tpm 8
RN_calc 9
RN_calc_GPV 10
RN_calc_LPV 11
RN_IsoSwitch_Example_S7 11
RN_iso_calc 12
RN_iso_select 14
RN_pmi 15
RN_select 17

Index **19**

`RNentropy-package` *Entropy Based Method for the Detection of Significant Variation in Gene Expression Data*

Description

An implementation of a method based on information theory devised for the identification of genes showing a significant variation of expression across multiple conditions. Given expression estimates from any number of RNA-Seq samples and conditions it identifies genes or transcripts with a significant variation of expression across all the conditions studied, together with the samples in which they are over- or under-expressed. It also detects genes whose relative isoform usage changes across samples (isoform switching). Zambelli et al. (2018) <doi:10.1093/nar/gky055>.

Details

The package provides two analyses.

Variation of expression: `RNentropy` (from a file) or `RN_calc` (from a data.frame) compute a global p-value for each gene or transcript, testing whether its expression varies across all samples, and a local p-value for each sample, testing whether it is over- or under-expressed. `RN_select` selects genes with a significant global p-value and summarises the local p-values of each condition, and `RN_pmi` computes the point mutual information between conditions.

Isoform switching: `RNentropy_iso_switch` (from a file) or `RN_iso_calc` (from a data.frame) compute, for each gene and each sample, a p-value for a change in relative isoform usage with respect to the samples of the other conditions. `RN_iso_select` combines them into a gene-level p-value and selects the genes with a significant change.

Author(s)

Federico Zambelli [aut, cre] (ORCID: <https://orcid.org/0000-0003-3487-4331>), Giulio Pavesi [aut] (ORCID: <https://orcid.org/0000-0001-5705-6249>)

Maintainer: Federico Zambelli <federico.zambelli@unimi.it>

References

Zambelli F, Mastropasqua F, Picardi E, D'Erchia AM, Pesole G, Pavesi G (2018). RNentropy: an entropy-based tool for the detection of significant variation of gene expression across multiple RNA-Seq experiments. *Nucleic Acids Research*, 46(8), e46. doi:10.1093/nar/gky055

Zambelli F, Pavesi G (2021). Using RNentropy to Detect Significant Variation in Gene Expression Across Multiple RNA-Seq or Single-Cell RNA-Seq Samples. In *RNA Bioinformatics*, Methods in Molecular Biology, 2284, 77–96. doi:10.1007/9781071613078_6

Examples

```
#load expression values and experiment design
data("RN_Brain_Example_tpm", "RN_Brain_Example_design")
#compute statistics and p-values (considering only a subset of genes due to
#examples running time limit of CRAN).
Results <- RN_calc(RN_Brain_Example_tpm[1:10000,], RN_Brain_Example_design)
#select only genes with significant changes of expression
Results <- RN_select(Results)
#Compute the Point Mutual information Matrix
Results <- RN_pmi(Results)

#load expression values and experiment design
data("RN_BarresLab_FPKM", "RN_BarresLab_design")
#compute statistics and p-values (considering only a subset of genes due to
#examples running time limit of CRAN)
Results_B <- RN_calc(RN_BarresLab_FPKM[1:10000,], RN_BarresLab_design)
#select only genes with significant changes of expression
Results_B <- RN_select(Results_B)
#Compute the Point Mutual information matrix
Results_B <- RN_pmi(Results_B)

#isoform-switch analysis on a subset of the S7 example
Iso_file <- system.file("extdata", "isoswitch_s7.tsv", package = "RNentropy")
Iso_results <- RNentropy_iso_switch(Iso_figle, tr.col = "TR_ID",
  gene.col = "GENE_ID")
Iso_results <- RN_iso_select(Iso_results)
Iso_results$selected
```

RNentropy *RNentropy*

Description

This function runs RNentropy on a file containing normalized expression values.

Usage

```
RNentropy(file, tr.col, design = NULL, header = TRUE, skip = 0, skip.col = NULL,
col.names = NULL)
```

Arguments

`file` Expression data file. It must contain a column with univocal identifiers for each row (typically transcript or gene ids).

`tr.col` The column # (1 based) in the file containing the univocal identifiers (usually transcript or gene ids).

`design` The RxC design matrix where R (rows) corresponds to the number of numeric columns (samples) in 'file' and C (columns) to the number of conditions. It must be a binary matrix with one and only one '1' for every row, corresponding to the condition (column) for which the sample corresponding to the row has to be considered a biological or technical replicate. See the example 'RN_Brain_Example_design' for the design matrix of 'RN_Brain_Example_tpm' which has three replicates for three conditions (three rows) for a total of nine samples (nine rows). design defaults to a square matrix of independent samples (diagonal = 1, everything else = 0).

`header` Set this to FALSE if 'file' does not have a header row. If header is TRUE the read.table function used by RNentropy will try to assign names to each column using the header. Default = TRUE

`skip` Number of rows to skip at the beginning of 'file'. Rows beginning with a '#' token will be skipped even if skip is set to 0.

`skip.col` Columns that will not be imported from 'file' (1 based) and not included in the subsequent analysis. Useful if you have more than one annotation column, e.g. gene name in the first, transcript ID in the second columns.

`col.names` Assign names to the imported columns. The number of names must correspond to the columns effectively imported (thus, no name for the skipped ones). Also, tr.col is not considered an imported column. Useful to assign sample names to expression columns.

Value

Refer to the help for RN_calc.

Author(s)

Giulio Pavesi - Dep. of Biosciences, University of Milan

Federico Zambelli - Dep. of Biosciences, University of Milan

Examples

```
#synthetic example file: transcript IDs, gene IDs and comments in the first
#three columns, then two replicates for each of three conditions
Example_file <- system.file("extdata", "RNentropy_example.tsv",
  package = "RNentropy")
Sample_names <- c("A_1", "A_2", "B_1", "B_2", "C_1", "C_2")
Example_design <- matrix(
  c(1, 0, 0,
    1, 0, 0,
    0, 1, 0,
```

```
    0, 1, 0,
    0, 0, 1,
    0, 0, 1),
  nrow = 6, byrow = TRUE,
  dimnames = list(Sample_names, c("A", "B", "C"))
)
#compute statistics and p-values
Results <- RNentropy(Example_file, tr.col = 1, design = Example_design,
  header = FALSE, skip.col = c(2, 3), col.names = Sample_names)
#select only genes with significant changes of expression
Results <- RN_select(Results)
Results$selected
```

`RNentropy_iso_switch` *RNentropy_iso_switch*

Description

This function runs the RNentropy isoform-switch analysis on a file containing transcript expression values and the corresponding gene identifiers.

Usage

```
RNentropy_iso_switch(file, tr.col, gene.col, design = NULL,
  header = TRUE, skip = 0, skip.col = NULL, col.names = NULL,
  min_expr = 1, pseudocount = 0.01)
```

Arguments

`file`	Expression data file. It must contain a column with univocal transcript identifiers and a column with the corresponding gene identifiers.
`tr.col`	The column name or number (1 based) in the file containing the univocal transcript identifiers.
`gene.col`	The column name or number (1 based) in the original file containing the gene identifiers.
`design`	The design matrix. Refer to the help for RN_iso_calc for further info.
`header`	Set this to FALSE if `file` does not have a header row. If header is TRUE the read.table function used by RNentropy will try to assign names to each column using the header (Default = TRUE).
`skip`	Number of rows to skip at the beginning of `file`. Rows beginning with a '#' token will be skipped even if skip is set to 0.
`skip.col`	Columns that will not be imported from `file` (1 based) and not included in the subsequent analysis.
`col.names`	Assign names to the imported columns. The number of names must correspond to the columns effectively imported (thus, no name for the skipped ones). Also, tr.col is not considered an imported column.

min_expr	Expression threshold used to determine whether a gene can be tested. Refer to the help for RN_iso_calc for the exact rule (Default = 1).
pseudocount	Positive pseudocount added to the isoform frequencies before re-normalization (Default = 0.01).

Value

Refer to the help for RN_iso_calc.

Author(s)

Giulio Pavesi - Dep. of Biosciences, University of Milan

Federico Zambelli - Dep. of Biosciences, University of Milan

See Also

RN_iso_calc, RN_iso_select

Examples

```
File <- system.file("extdata", "isoswitch_s7.tsv", package = "RNentropy")
Results <- RNentropy_iso_switch(File, tr.col = "TR_ID",
  gene.col = "GENE_ID")
Results$gene_status
```

RN_BarresLab_design *RN_BarresLab_design*

Description

The experiment design matrix for RN_BarresLab_FPKM.

Usage

```
data("RN_BarresLab_design")
```

Format

The format is: int [1:7, 1:7]

Details

A binary matrix where conditions correspond to columns and samples to rows. In this example there are seven conditions (cell types) and seven samples with the expression measured by mean FPKM.

Examples

```
data(RN_BarresLab_design)
```

RN_BarresLab_FPKM *RN_BarresLab_FPKM*

Description

Data expression values (mean FPKM) for different cerebral cortex cell types from Zhang et al: "An RNA-sequencing transcriptome and splicing database of glia, neurons, and vascular cells of the cerebral cortex", J Neurosci 2014 Sep. RN_BarresLab_design is the corresponding design matrix.

Usage

```
data("RN_BarresLab_FPKM")
```

Format

A data frame with 12978 observations on the following 7 variables.

`Astrocyte` a numeric vector

`Neuron` a numeric vector

`OPC` a numeric vector

`NFO` a numeric vector

`MO` a numeric vector

`Microglia` a numeric vector

`Endothelial` a numeric vector

Source

Zhang Y, Chen K, Sloan SA, Bennett ML, Scholze AR, O'Keeffe S, Phatnani HP, Guarnieri P, Caneda C, Ruderisch N, Deng S, Liddelow SA, Zhang C, Daneman R, Maniatis T, Barres BA, Wu JQ. An RNA-sequencing transcriptome and splicing database of glia, neurons, and vascular cells of the cerebral cortex. J Neurosci. 2014 Sep 3;34(36):11929-47. doi: 10.1523/JNEUROSCI.1860-14.2014. Erratum in: J Neurosci. 2015 Jan 14;35(2):846-6. PubMed PMID: 25186741; PubMed Central PMCID: PMC4152602.

Data were downloaded from https://web.stanford.edu/group/barres_lab/brain_rnaseq.html

Examples

```
data(RN_BarresLab_FPKM)
```

RN_Brain_Example_design *RN_Brain_Example_design*

Description

The example design matrix for RN_Brain_Example_tpm

Usage

```
data("RN_Brain_Example_design")
```

Format

The format is: num[1:9,1:3]

Details

A binary matrix where conditions correspond to the columns and samples to the rows. In this example there are three conditions (individuals) and nine samples, with three replicates for each condition.

Examples

```
data(RN_Brain_Example_design)
```

RN_Brain_Example_tpm *RN_Brain_Example_tpm*

Description

Example data expression values from brain of three different individuals, each with three replicates. RN_Brain_Example_design is the corresponding design matrix.

Usage

```
data("RN_Brain_Example_tpm")
```

Format

A data frame with 78699 observations on the following 10 variables.

`GENE` a factor containing gene identifiers

`BRAIN_2_1` a numeric vector

`BRAIN_2_2` a numeric vector

BRAIN_2_3 a numeric vector

BRAIN_3_1 a numeric vector

BRAIN_3_2 a numeric vector

BRAIN_3_3 a numeric vector

BRAIN_1_1 a numeric vector

BRAIN_1_2 a numeric vector

BRAIN_1_3 a numeric vector

Examples

```
data(RN_Brain_Example_tpm)
```

RN_calc *RN_calc*

Description

Computes both global and local p-values, and returns the results in a list containing for each gene the original expression values and the associated global and local p-values (as -log10(p-value)).

Usage

```
RN_calc(X, design = NULL)
```

Arguments

X	data.frame with expression values. It may contain additional non numeric columns (eg. a column with gene names).
design	The RxC design matrix where R (rows) corresponds to the number of numeric columns (samples) in 'X' and C (columns) to the number of conditions. It must be a binary matrix with one and only one '1' for every row, corresponding to the condition (column) for which the sample corresponding to the row has to be considered a biological or technical replicate. See the example 'RN_Brain_Example_design' for the design matrix of 'RN_Brain_Example_tpm' which has three replicates for three conditions (three rows) for a total of nine samples (nine rows). design defaults to a square matrix of independent samples (diagonal = 1, everything else = 0)

Value

gpv	-log10 of the global p-values
lpv	-log10 of the local p-values
c_like	results formatted as in the output of the C++ implementation of RNentropy.
res	The results data.frame with the original expression values and the associated -log10 of global and local p-values.
design	the experimental design matrix

Author(s)

Giulio Pavesi - Dep. of Biosciences, University of Milan

Federico Zambelli - Dep. of Biosciences, University of Milan

Examples

```
data("RN_Brain_Example_tpm", "RN_Brain_Example_design")
#compute statistics and p-values (considering only a subset of genes due to
#examples running time limit of CRAN)
Results <- RN_calc(RN_Brain_Example_tpm[1:10000,], RN_Brain_Example_design)
```

RN_calc_GPV *RN_calc_GPV*

Description

This function calculates global p-values from expression data, represented as -log10(p-values).

Usage

```
RN_calc_GPV(X, bind = TRUE)
```

Arguments

X	data.frame with expression values. It may contain additional non numeric columns (eg. a column with gene names)
bind	See Value (Default: TRUE)

Value

If bind is TRUE the function returns a data.frame with the original expression values from 'X' and an attached column with the -log10() of the global p-value, otherwise only the numeric vector of -log10(p-values) is returned.

Author(s)

Giulio Pavesi - Dep. of Biosciences, University of Milan

Federico Zambelli - Dep. of Biosciences, University of Milan

Examples

```
data("RN_Brain_Example_tpm")
GPV <- RN_calc_GPV(RN_Brain_Example_tpm)
```

RN_calc_LPV — *RN_calc_LPV*

Description

This function calculates local p-values from expression data, output as -log10(p-value).

Usage

```
RN_calc_LPV(X, design = NULL, bind = TRUE)
```

Arguments

X	data.frame with expression values. It may contain additional non numeric columns (eg. a column with gene names)
design	The design matrix. Refer to the help for RN_calc for further info.
bind	See Value (Default = TRUE)

Value

If bind is TRUE the function returns a data.frame with the original expression values from data.frame X and attached columns with the -log10(p-values) computed for each sample, otherwise only the matrix of -log10(p-values) is returned.

Author(s)

Giulio Pavesi - Dep. of Biosciences, University of Milan

Federico Zambelli - Dep. of Biosciences, University of Milan

Examples

```
data("RN_Brain_Example_tpm", "RN_Brain_Example_design")
LPV <- RN_calc_LPV(RN_Brain_Example_tpm, RN_Brain_Example_design)
```

RN_IsoSwitch_Example_S7 — *RN_IsoSwitch_Example_S7*

Description

Transcript expression values from six tissues in the complete historical S7 example used to validate the RNentropy isoform-switch implementation. RefSeq transcript identifiers are used as row names.

Usage

```
data("RN_IsoSwitch_Example_S7")
```

Format

A data frame with 81314 observations on the following 7 variables.

`GENE_ID` a character vector containing gene identifiers

`brain` a numeric vector

`heart` a numeric vector

`kidney` a numeric vector

`liver` a numeric vector

`lung` a numeric vector

`muscle` a numeric vector

Source

Historical S7 benchmark preserved with the independent C++ implementation of RNentropy at https://github.com/Federico77z/RNentropy_cpp.

See Also

`RN_iso_calc`, `RN_iso_select`

Examples

```
data(RN_IsoSwitch_Example_S7)
```

RN_iso_calc *RN_iso_calc*

Description

This function calculates, for each gene and each sample, a p-value for changes in relative isoform usage, output as -log10(p-value). A gene is tested only if at least two of its isoforms have an expression value strictly greater than `min_expr` in at least one sample, and at least two samples have at least one isoform with an expression value strictly greater than `min_expr`.

Usage

```
RN_iso_calc(X, gene.col, design = NULL, min_expr = 1,
  pseudocount = 0.01)
```

Arguments

`X`	data.frame with one row for each transcript, a column containing the corresponding gene identifier, and numeric expression columns. It may contain additional non numeric columns.
`gene.col`	The column name or number in `X` containing the gene identifiers.
`design`	The RxC design matrix where R (rows) corresponds to the number of numeric columns (samples) in `X` and C (columns) to the number of conditions. It must be a binary matrix with one and only one '1' for every row. Replicates of the sample being tested are excluded from the reference distribution. design defaults to a square matrix of independent samples (diagonal = 1, everything else = 0).
`min_expr`	Expression threshold used to determine whether a gene can be tested (see Description). The two isoforms and the two samples above the threshold do not need to coincide: for example, one isoform expressed only in one sample and another isoform expressed only in a different sample is sufficient. The threshold only determines whether a gene is tested; all expression values of a tested gene, including those below the threshold, are used in the test (Default = 1).
`pseudocount`	Positive pseudocount added to the isoform frequencies before re-normalization (Default = 0.01).

Value

A list containing:

`expr`	The original expression data.frame.
`design`	The experimental design matrix.
`pv`	A gene by sample matrix containing -log10(p-values), with columns named `ISO_PV_<sample>`. Values that cannot be calculated are reported as `NA`.
`gene_status`	A named vector reporting `TESTED`, `NOTEST(ISO)`, or `NOTEST(EXP)` for each gene.
`sample_status`	A gene by sample matrix reporting `TESTED`, `NO_CURRENT_EXPRESSION`, or `NO_REFERENCE_EXPRESSION`. The last two values correspond to '-' and '*' in the C++ implementation, respectively.

Author(s)

Giulio Pavesi - Dep. of Biosciences, University of Milan

Federico Zambelli - Dep. of Biosciences, University of Milan

See Also

`RNentropy_iso_switch`, `RN_iso_select`, `RN_IsoSwitch_Example_S7`

Examples

```
Isoforms <- data.frame(
  gene = c("gene_1", "gene_1", "gene_2"),
  sample_1 = c(10, 1, 4),
  sample_2 = c(1, 10, 5)
)
Results <- RN_iso_calc(Isoforms, gene.col = "gene")
Results$gene_status
Results$pv
```

RN_iso_select *Select genes with significant changes in relative isoform usage*

Description

Combine the isoform-switch p-values of each gene into one gene-level p-value and select genes with a gene-level p-value lower than or equal to a user-defined threshold.

Usage

```
RN_iso_select(Results, gene_pv = 0.05)
```

Arguments

`Results`	The output of `RN_iso_calc` or `RNentropy_iso_switch`.
`gene_pv`	Threshold for gene-level p-values, expressed as a raw p-value between 0 and 1 (Default: 0.05).

Details

For each gene with `gene_status` equal to `"TESTED"`, the finite sample p-values $p_{(1)} \leq \cdots \leq p_{(k)}$ are combined with the Simes method into the gene-level p-value $\min_i kp_{(i)}/i$. It tests the null hypothesis that relative isoform usage does not change in any sample, accounting for the multiple sample tests of the gene, and it is valid under independence and under the positive dependence expected between samples of the same gene.

Gene-level p-values are compared directly with `gene_pv`; no correction is applied across genes. Genes that were not tested, and TESTED genes without any finite sample p-value, receive no gene-level p-value and are never selected. Sample p-values are not corrected: the original isoform-switch p-values are reported next to the gene-level value to show in which samples the change occurs.

Value

The original input list containing two additional components:

`gene_pv`	A named vector containing, for each gene, the Simes gene-level p-value as `-log10(p-value)`. Genes without a gene-level p-value are reported as `NA`.

`selected` A data.frame containing TESTED genes with a gene-level p-value less than or equal to `gene_pv`. Gene identifiers are stored as row names. The data.frame contains gene status, the gene-level p-value (`ISO_GENE_PV`), and the original isoform-switch p-values (`ISO_PV_<sample>`), all as `-log10(p-value)`. Rows are ordered by decreasing `ISO_GENE_PV`.

Author(s)

Giulio Pavesi - Dep. of Biosciences, University of Milan

Federico Zambelli - Dep. of Biosciences, University of Milan

References

Simes R. J. (1986) An improved Bonferroni procedure for multiple tests of significance. *Biometrika*, 73(3), 751–754.

See Also

`RN_iso_calc`, `RNentropy_iso_switch`

Examples

```
Isoforms <- data.frame(
  gene = c("gene_1", "gene_1", "gene_2", "gene_2"),
  sample_1 = c(10, 1, 5, 5),
  sample_2 = c(1, 10, 5, 5)
)
Results <- RN_iso_calc(Isoforms, gene.col = "gene")
Results <- RN_iso_select(Results)
Results$gene_pv
Results$selected
```

RN_pmi — *Compute point mutual information matrix for the experimental conditions*

Description

Compute point mutual information for experimental conditions from the over-expressed genes identified by RN_select.

Usage

```
RN_pmi(Results)
```

Arguments

`Results` The output of RNentropy, RN_calc or RN_select. If RN_select has not already run on the results it will be invoked by RN_pmi using default arguments.

Value

The original input containing

gpv	-log10 of the Global p-values
lpv	-log10 of the Local p-values
c_like	a table similar to the one you obtain running the C++ implementation of RNentropy.
res	The results data.frame containing the original expression values together with the -log10 of Global and Local p-values.
design	The experimental design matrix.
selected	Transcripts/genes with a corrected Global p-value lower than gpv_t. Each condition N gets a condition_N column which values can be -1,0,1 or NA. 1 means that all the replicates of this condition seems to be consistently over-expressed w.r.t the overall expression of the transcript in all the conditions (that is, all the replicates of condition N have a positive Local p-value <= lpv_t). -1 means that all the replicates of this condition seems to be consistently under-expressed w.r.t the overall expression of the transcript in all the conditions (that is, all the replicates of condition N have a negative Local p-value and abs(Local p-values) <= lpv_t). 0 means that one or more replicates have an abs(Local p-value) > lpv_t. NA means that the Local p-values of the replicates are not consistent for this condition.

And two new matrices:

pmi	Point mutual information matrix of the conditions.
npmi	Normalized point mutual information matrix of the conditions.

Author(s)

Giulio Pavesi - Dep. of Biosciences, University of Milan

Federico Zambelli - Dep. of Biosciences, University of Milan

Examples

```
data("RN_Brain_Example_tpm", "RN_Brain_Example_design")
#compute statistics and p-values (considering only a subset of genes due to
#examples running time limit of CRAN)
Results <- RN_calc(RN_Brain_Example_tpm[1:10000,], RN_Brain_Example_design)
Results <- RN_select(Results)
Results <- RN_pmi(Results)
```

RN_select *Select transcripts/genes with significant p-values*

Description

Select transcripts with global p-value lower than a user-defined threshold and provide a summary of over- or under-expression according to local p-values.

Usage

```
RN_select(Results, gpv_t = 0.01, lpv_t = 0.01, method = "BH")
```

Arguments

Results	The output of RNentropy or RN_calc.
gpv_t	Threshold for global p-value. (Default: 0.01)
lpv_t	Threshold for local p-value. (Default: 0.01)
method	Multiple test correction method. Available methods are the ones of p.adjust. Type p.adjust.methods to see the list. Default: BH (Benjamini & Hochberg)

Value

The original input containing

gpv	-log10 of the global p-values
lpv	-log10 of the local p-values
c_like	results formatted as in the output of the C++ implementation of RNentropy.
res	The results data.frame containing the original expression values together with the -log10 of global and local p-values.
design	The experimental design matrix.

and a new dataframe

selected	Transcripts/genes with a corrected global p-value lower than gpv_t. For each condition it will contain a column where values can be -1,0,1 or NA. 1 means that all the replicates of this condition have expression value higher than the average and local p-value <= lpv_t (thus the corresponding gene will be over-expressed in this condition). -1 means that all the replicates of this condition have expression value lower than the average and local p-value <= lpv_t (thus the corresponding gene will be under-expressed in this condition). 0 means that at least one of the replicates has a local p-value > lpv_t. NA means that the local p-values of the replicates are not consistent for this condition, that is, at least one replicate results to be over-expressed and at least one results to be under-expressed.

Author(s)

Giulio Pavesi - Dep. of Biosciences, University of Milan

Federico Zambelli - Dep. of Biosciences, University of Milan

Examples

```
data("RN_Brain_Example_tpm", "RN_Brain_Example_design")
#compute statistics and p-values (considering only a subset of genes due to
#examples running time limit of CRAN)
Results <- RN_calc(RN_Brain_Example_tpm[1:10000,], RN_Brain_Example_design)
Results <- RN_select(Results)
```

Index

∗ **PMI**
RN_pmi, 15
∗ **RNentropy**
RN_calc, 9
RN_calc_GPV, 10
RN_calc_LPV, 11
RN_iso_calc, 12
RN_iso_select, 14
RN_pmi, 15
RN_select, 17
RNentropy, 3
RNentropy_iso_switch, 5
∗ **Run**
RN_calc, 9
RN_iso_calc, 12
RNentropy_iso_switch, 5
∗ **Select**
RN_iso_select, 14
RN_select, 17
∗ **datasets**
RN_BarresLab_design, 6
RN_BarresLab_FPKM, 7
RN_Brain_Example_design, 8
RN_Brain_Example_tpm, 8
RN_IsoSwitch_Example_S7, 11
∗ **package**
RNentropy-package, 2

RN_BarresLab_design, 6
RN_BarresLab_FPKM, 7
RN_Brain_Example_design, 8
RN_Brain_Example_tpm, 8
RN_calc, *2*, 9
RN_calc_GPV, 10
RN_calc_LPV, 11
RN_iso_calc, *2*, *6*, *12*, 12, *15*
RN_iso_select, *2*, *6*, *12*, *13*, 14
RN_IsoSwitch_Example_S7, 11, *13*
RN_pmi, *2*, 15
RN_select, *2*, 17
RNentropy, *2*, 3
RNentropy-package, 2
RNentropy_iso_switch, *2*, 5, *13*, *15*