Package ‘INSPEcT’

March 22, 2017

Type Package
Title Analysis of 4sU-seq and RNA-seq time-course data
Version 1.4.0
Date 2015-12-15
Author Stefano de Pretis
Maintainer Stefano de Pretis <ste.depo@gmail.com>
Description INSPEcT (INference of Synthesis, Processing and
dEgradation rates in Time-Course experiments) analyses 4sU-seq
and RNA-seq time-course data in order to evaluate synthesis,
processing and degradation rates and assas via modeling the
rates that determines changes in mature mRNA levels.
License GPL-2
Depends R (>= 3.2), methods, Biobase, BiocParallel
Imports pROC, deSolve, rootSolve, compiler, preprocessCore,
GenomicFeatures, GenomicRanges, IRanges, BiocGenerics,
GenomicAlignments, Rsamtools, S4Vectors
Suggests BiocStyle, knitr, TxDb.Mmuscus.UCSC.mm9.knownGene
VignetteBuilder knitr
biocViews Sequencing, RNASeq, GeneRegulation, TimeCourse,
SystemsBiology
NeedsCompilation no
RoxygenNote 5.0.1

R topics documented:

AIC-INSPEcT-method ................................................. 2
chisqmodel ........................................................... 3
chisqtest ............................................................. 4
combine ............................................................... 4
compareSteady ......................................................... 5
dim,INSPEcT-method ............................................... 6
Extract ............................................................... 7
featureNames,INSPEcT-method ...................................... 8
geneClass ........................................................... 8
getModel ............................................................. 9
AIC-INSPEcT-method

Akaike information criterion calculated for the models evaluated by INSPEcT

Description

This method is used to retrieve AIC values for all models tested for all genes.
## chisqmodel

### Usage

```r
## S4 method for signature 'INSPEcT_model'
AIC(object, ..., k = 2)
```

```r
## S4 method for signature 'INSPEcT'
AIC(object, ..., k = 2)
```

### Arguments

- `object`: An object of class `INSPEcT` or `INSPEcT_model`
- `...`: Additional arguments for the generic
- `k`: Additional parameter for the generic

### Value

A matrix of AIC values

### Examples

```r
data('mycerIds10', package='INSPEcT')
AIC(mycerIds10)
```

---

## chisqmodel

Retrieve results of chi-squared test for the selected models

### Description

Retrieve results of chi-squared test for the selected models

This method is used to retrieve the chi-squared test results for the models that have been selected to better represent the behavior of each gene.

### Usage

```r
chisqmodel(object, ...)
```

```r
## S4 method for signature 'INSPEcT_model'
chisqmodel(object, ...)
```

```r
## S4 method for signature 'INSPEcT'
chisqmodel(object, ...)
```

### Arguments

- `object`: An object of class `INSPEcT` or `INSPEcT_model`
- `...`: Additional arguments for the generic

### Value

A vector of chi-squared test results
Examples

data('mycerIds10', package='INSPEcT')
chisqmodel(mycerIds10)

chisqtest <ins>Retrieve all results of chi-squared test</ins>

Description

Retrieve all results of chi-squared test
This method is used to retrieve all the chi-squared test results for all models tested for all genes.

Usage

chisqtest(object, ...)

## S4 method for signature 'INSPEcT_model'
chisqtest(object, ...)

## S4 method for signature 'INSPEcT'
chisqtest(object, ...)

Arguments

object An object of class INSPEcT or INSPEcT_model
...
Additional arguments for the generic

Value

A matrix of chi-squared test results for all the tested models

Examples

data('mycerIds10', package='INSPEcT')
chisqtest(mycerIds10)

combine <ins>Combine different Objects of Class INSPEcT</ins>

Description

This method combines the information coming from different Objects of INSPEcT class. Requirements for two or more object to be combined together are:

• they must be either modeled or either not modeled
• they must have the same time points
• they must have the same modeling parameters
compareSteady

Usage

## S4 method for signature 'INSPEcT,INSPEcT'
combine(x, y, ...)

Arguments

x
An object of class INSPEcT

y
An object of class INSPEcT

... Additional objects of class INSPEcT

Details

In case the same gene is contained in more than one object that the user tries to combine, the
information from one object will be used and a warning will be reported

Value

An Object of class INSPEcT

Examples

data('mycerIds10', package='INSPEcT')
mycerIds_2genes <- mycerIds10[1:2]
mycerIds_5genes <- mycerIds10[6:10]
mycerIds_7genes <- combine(mycerIds_2genes, mycerIds_5genes)

compareSteady  Generate an object of class INSPEcT_diffsteady from two objects of
class INSPEcT

Description

Generate an object of class INSPEcT_diffsteady from two objects of class INSPEcT
This method compares two object of class INSPEcT in order to identify differential usage of syn-
thesis, processing or degradation rates in two different steady-state conditions. The two INSPEcT
objects must have been profiled with replicates in order to provide a statistical significance to the
differences between their rates.

Usage

compareSteady(inspectIds1, inspectIds2)

## S4 method for signature 'INSPEcT,INSPEcT'
compareSteady(inspectIds1, inspectIds2)

Arguments

inspectIds1 An object of class INSPEcT

inspectIds2 A second object of class INSPEcT
**dim,INSPEcT-method**

**Value**

An object of class INSPEcT_diffsteady which contains both the absolute quantification of the rates as well as the comparison with the statistical significance associated for each gene and rate. (See [INSPEcT_diffsteady-class](#))

**Examples**

```r
## load data
data('simRates', package='INSPEcT')
data('simData3rep', package='INSPEcT')
## generate a new data set with 3 replicate to make the comparison
newTpts <- c(0, 1/6)
simData3rep_2 <- makeSimDataset(simRates, newTpts, 3, seed=2)
## compare
diffrates <- compareSteady(simData3rep, simData3rep_2)
## visualize results
diffrates
```

---

**Description**

A method to obtain the dimension of the object of class INSPEcT reported as a vector containing of the genes and the number of time points

**Usage**

```r
## S4 method for signature 'INSPEcT'
dim(x)
```

**Arguments**

- `x` An object of class INSPEcT

**Value**

A numeric that indicates the number of genes within the object and the number of time points contained the object

**See Also**

`nGenes, nTpts`
Extract Parts of an INSPEcT or an INSPEcT_model Object

Description

Operators acting on INSPEcT or INSPEcT_model objects to extract parts. INSPEcT_model objects can be subsetted only by gene. INSPEcT objects can be subsetted either by gene id or time point. In case of subsetting an INSPEcT object by time point, the model should be empty.

Usage

## S4 method for signature 'INSPEcT_model,ANY,ANY'

x[i]

## S4 method for signature 'INSPEcT,ANY,ANY'

x[i, j]

Arguments

- **x**: An object of class INSPEcT or INSPEcT_model
- **i**: A numeric, a vector of logicals or a vector of names indicating the features to be extracted
- **j**: A numeric, a vector of logicals indicating the time points to be extracted

Value

An Object of class INSPEcT

See Also

removeModel

Examples

data('mycerIds10', package='INSPEcT')
mycerIds_5genes <- mycerIds10[1:5]
## Not run:
## This will turn out into an error:
mycerIds_5genes_5tpts <- mycerIds10[1:5, 1:5]

## End(Not run)
## Before subsetting time points, the model should be removed:
mycerIds_5genes_5tpts <- removeModel(mycerIds10)[1:5, 1:5]
## Description

A method to visualize gene names associated with the object of class INSPEcT

### Usage

```r
## S4 method for signature 'INSPEcT'
featureNames(object)

## S4 replacement method for signature 'INSPEcT'
featureNames(object) <- value
```

### Arguments

- **object**: An object of class INSPEcT
- **value**: A character that will replace the current feature names

### Value

A character that contains gene names associated with the object of class INSPEcT

## geneClass

*Retrieve the regulatory class for each gene*

### Description

Retrieve the regulatory class for each gene

This method returns a factor that summarises the gene class (transcriptional regulatory mechanism) that INSPEcT has assigned to each gene. The classification depends on the chi-squared and Brown’s method thresholds, that can be both provided as arguments. If the user decides a different thresholding respect to the default, these new values can be permanently set within the object.

### Usage

```r
geneClass(object, bTsh = NULL, cTsh = NULL)

## S4 method for signature 'INSPEcT_model'
geneClass(object, bTsh = NULL, cTsh = NULL)

## S4 method for signature 'INSPEcT'
geneClass(object, bTsh = NULL, cTsh = NULL)
```
getModel

Arguments

object An object of class INSPEcT or INSPEcT_model
bTsh A numeric representing the p-value threshold for considering a rate as variable. P-values are calculated through ratePvals
cTsh A numeric representing the threshold for the chi-squared test to consider a model as valid

Value

A character containing the regulatory class for each gene

See Also

ratePvals

Examples

data('mycerIds10', package='INSPEcT')
geneClass(mycerIds10)
# see the classification with another threshold for chi-squared test
geneClass(mycerIds10, cTsh=.2)
# set the new threshold permanently within the object
thresholds(mycerIds10)$chisquare <- .2

getModel

Get or replace INSPEcT_model object within INSPEcT object

Description

Get or replace INSPEcT_model object within INSPEcT object

A method to get or set the INSPEcT_model object within an INSPEcT object. This method is particularly useful to get and set testing parameters of the INSPEcT_model object within the INSPEcT object.

Usage

getModel(object)

getModel(object) <- value

## S4 method for signature 'INSPEcT'
getModel(object)

## S4 replacement method for signature 'INSPEcT'
getModel(object) <- value

Arguments

object An object of class INSPEcT
value An object of class INSPEcT model
inHeatmap

Description

Heatmap that represent the fold changes of all the five features

A method to see as an heatmap the logRatios of synthesis, degradation and processing rates and pre-mRNA and total mRNA concentration of a population of genes, either at the level of estimated or modeled rates.

Usage

```r
inHeatmap(object, type = "pre-model", breaks = seq(-1, 1, length.out = 51),
palette = colorRampPalette(c("green", "black", "firebrick3")),
plot_matureRNA = FALSE, absoluteExpression = TRUE, rowLabels = NULL,
clustering = TRUE, clustIdx = 3:5)
```

Arguments

- **object**: An object of class INSPEcT
- **type**: Either "pre-model" or "model" to switch between pre-modeled or modeled features
- **breaks**: A vector of breaks for the heatmap
- **palette**: A color generating function, output of colorRampPalette
- **plot_matureRNA**: A logical. If set to TRUE, mature-mRNA is displayed instead of total-mRNA (default: FALSE)
- **absoluteExpression**: A logical. If set to FALSE, the plot representing the intensity of expression is omitted. (default=TRUE)
- **rowLabels**: A character that represent the label names that will be shown on the y-axis of the heatmap. If NULL featureNames(object) will be shown (default is NULL)
clustering A logical. If set to FALSE, it displays genes the order they are, with no clustering (default: TRUE)

clustIdx A numeric. Indicates which of the features are used for the clustering. 0=absoluteExpression; 1=total-mRNA/mature-mRNA; 2=preMRNA; 3=synthesis; 4=degradation; 5=processing (default=3;5, meaning that synthesis, degradation and processing are used for the clustering)

Value
A list of matrices containing the logRatios for total mRNA levels, pre-mRNA levels, synthesis rates, degradation rates and processing rates. Matrices are ordered according to the clustering.

Examples
data('mycerIds10', package='INSPEcT')
inHeatmap(mycerIds10, 'pre-model')
inHeatmap(mycerIds10, 'model')

Description
INSPEcT (INference of Synthesis, Processing and dEgradation rates in Time-cousre experiments) is a package that analyse 4sU-seq and RNA-seq time-course data in order to evaluate synthesis, processing and degradation rates and asses via modeling the rates that determines changes in mature mRNA levels.

It implements two classes (INSPEcT_model and INSPEcT) and their corresponding methods. To have a detailed description of how the two classes are structured and which methods apply on, type:

?'INSPEcT-class'
?'INSPEcT_model-class'
?'INSPEcT_diffsteady-class'

To see how the typical workflow of INSPEcT works, type:
vignette('INSPEcT')

Last but not least, to obtain a citation, type:
citation('INSPEcT')

Description
INSPEcT is a class able to store all the estimated rates and concentrations (slot ratesFirstGuess), the modeled rates and concentrations (slot modelRates) and the model themselves (slot model). Within the class INSPEcT other information regarding the experimental design are stored, such as the time points where experiments were collected (slot tpts), the labeling time (slot tL) and the normalization scale fators used for RNA- (totalSF) and 4sU-seq libraries (labeledSF). A list of parameters that will be used during the modeling process is stored within the slot params and can be accessed by modelingParams. A new instance of the class INSPEcT can be generated by the constructor function newINSPEcT.
Usage

```
## S4 method for signature 'INSPEcT'
show(object)
```

Arguments

- **object**: An object of class INSPEcT

Details

Methods that apply to INSPEcT class are

- `[]`
- `AIC`
- chisqmodel
- chisqtest
- combine
- dim
- featureNames
- geneClass
- getModel<-
- getModel
- inHeatmap
- labeledSF llrtests
- logLik
- makeModelRates
- makeSimModel
- modelingParams<-
- modelingParams
- modelRates
- modelSelection
- nGenes
- nTpts
- plotGene
- ratePvals
- ratesFirstGuessVar
- ratesFirstGuess
- removeModel
- sfPlot
- thresholds
- totalSF
- tpts
- viewModelRates

Value

Method show for objects of class INSPEcT displays the main features of the slots ratesFirstGuess, model and modelRates

Slots

- `params`: A list of parameters of the modeling part
ratesFirstGuess An object of class ExpressionSet that contains all the rates and concentrations guessed from the first part of INSPEcT analysis (before modeling)

precision A matrix that contains the estimated precision of the rates. Rows represent genes, Columns represent time points.

model1 An object of class INSPEcT_model that contains the output of the modeling

modelRates An object of class ExpressionSet that contains all modeled the rates and concentrations

tpts A numeric vector of the time-points

labeledSF A numeric vector of the scaling factor used for inter time-point normalization of 4sU-seq libraries

totalsF A numeric vector of the scaling factor used for inter time-point normalization of RNA-seq libraries

tl A numeric containing the length of the 4sU pulse

INSPEcT_diffsteady-class

An S4 class to represent comparisons between two steady-state conditions

Description

INSPEcT_diffsteady is a class able to store the results of the comparisons between two steady states. An object of class INSPEcT_diffsteady is created with the method "compareSteady" applied on two "INSPEcT" objects (see compareSteady).

Usage

synthesis(object)

processing(object)

degradation(object)

## S4 method for signature 'INSPEcT_diffsteady'
show(object)

## S4 method for signature 'INSPEcT_diffsteady'
synthesis(object)

## S4 method for signature 'INSPEcT_diffsteady'
processing(object)

## S4 method for signature 'INSPEcT_diffsteady'
degradation(object)

Arguments

object An object of class INSPEcT_model
Details

Methods associated to the class INSPEcT_diffsteady are:

- synthesis: Accessor to the synthesis rates and their comparisons.
- degradation: Accessor to the degradation rates and their comparisons.
- processing: Accessor to the processing rates and their comparisons.
- plotMA: visualization function for rates comparisons, see plotMA

Value

Method show for objects of class INSPEcT_model returns the number of the genes that have been modeled

Slots

- synthesis: A data.frame which contains both input data and comparisons results regarding synthesis rates
- degradation: A data.frame which contains both input data and comparisons results regarding degradation rates
- processing: A data.frame which contains both input data and comparisons results regarding processing rates

Examples

data('simData3rep', package='INSPEcT')
data('simData3rep_2', package='INSPEcT')
diffrates <- compareSteady(simData3rep, simData3rep_2)
head(synthesis(diffrates))
head(processing(diffrates))
head(degradation(diffrates))

Description

INSPEcT_model is a class able to store all the results of the modeling of synthesis, processing and degradation rates made via the method modelRates (slot ratesSpecs). It also stores the criteria (slot parameter) to choose between the many models tested for each gene the one that better describes the data and the results. The slot simple is a flag that distinguish whether the model contains the information of the introns or not. In case not, the flag simple is set to TRUE. Also the method makeSimModel of class INSPEcT-class creates an object of class INSPEcT_model. This object will be used by makeSimDataset to generate a complete simulated data-set, whose classification performance can be tested.

Usage

```r
## S4 method for signature 'INSPEcT_model'
show(object)
```
**Arguments**

object An object of class INSPEcT_model

**Details**

Methods that apply to INSPEcT_model class are

- \[AIC\]
- chisqmodel
- chisqtest
- geneClass
- llrtests
- logLik
- makeModelRates
- makeSimDataset
- modelSelection
- ratePvals
- rocCurve
- rocThresholds
- thresholds

**Value**

Method show for objects of class INSPEcT_model returns the number of the genes that have been modeled

**Slots**

- \[params\] A list that defines thresholds and how to perform log likelihood ratio tests
- \[ratesSpecs\] A list containing the modeling output
- \[simple\] A logical that indicates whether the mode of INSPEcT is simple (no pre-mRNA and degradation rates) or not.

**Description**

Accessor to the slot labeledSF of an INSPEcT object

**Usage**

```r
labeledSF(object)
```

## S4 method for signature 'INSPEcT'
labeledSF(object)
logLik

Arguments

object An object of class INSPEcT

Value

A numeric that indicates the scaling factors applied between time points of the data coming from 4sU-seq library (applies directly to synthesis rates and indirectly to degradation rates)

Examples

data('mycerIds10')
labeledSF(mycerIds10)

logLik Retrieve results of log likelihood test

Description

Retrieve results of log likelihood test

This method is used to retrieve all the log likelihood ratio test results for all pairs tested for all genes.

Usage

logLik(object, ...)

## S4 method for signature 'INSPEcT_model'
logLik(object, ...)

## S4 method for signature 'INSPEcT'
logLik(object, ...)

Arguments

object An object of class INSPEcT or INSPEcT_model
...

Additional arguments for the generic

Value

A matrix of log likelihood test results for all the tested model comparisons

Examples

data('mycerIds10', package='INSPEcT')
logLik(mycerIds10)
Calculate modeled rates and concentrations

Description

Calculate modeled rates and concentrations

This function is used to evaluate rates and concentrations after modeling of the rates has been run with `modelRates`. The modeled rates are in functional form and can be evaluated at any time points.

This method can be used to regenerate the rates associated to the modeling, in case some testing parameters has changed.

Usage

```r
makeModelRates(object, ...)
```

## S4 method for signature 'INSPEcT_model'

```r
makeModelRates(object, ...)
```

## S4 method for signature 'INSPEcT'

```r
makeModelRates(object, ...)
```

Arguments

- `object`: An object of class `INSPEcT_model`
- `...`: additional arguments `tpts`: A vector of time points where rates and concentrations have to be evaluated

Value

An object of class `ExpressionSet` containing the modeled rates and concentrations

Examples

```r
data('mycerIds10', package='INSPEcT')
tpts <- c(0, 1/6, 1/3, 1/2, 1, 2, 4, 8, 16)
eSet <- makeModelRates(getModel(mycerIds10), tpts=tpts)
exprs(eSet)
```
```r
data('mycerIds10', package='INSPEcT')
viewModelRates(mycerIds10, 'degradation')
## force every degradation rate to be accepted as variable
thresholds(getModel(mycerIds10))$brown <- c(synthesis=.01, degradation=1, processing=.01)
mycerIds10 <- makeModelRates(mycerIds10)
viewModelRates(mycerIds10, 'degradation')
```
makeRPKMs

Calculate RPKM and count values on introns and exons from bam/sam files

Description

Given a TranscriptDb object and a list of bam/sam files for 4su and total RNA experiments, "makeRPKMs" function calculates read counts and RPKM on exonic and intronic features per each gene. Reads that fall where intronic and exonic features overlaps are univocally assigned to exons.

Usage

makeRPKMs(txdb, paths_fourSU, paths_total, by = c("gene", "tx"),
          countMultiMappingReads = FALSE, allowMultiOverlap = FALSE,
          strandSpecific = FALSE, isPairedEnd = FALSE)

Arguments

txdb       A TranscriptDB object
paths_fourSU A vector of paths of 4sU-seq sam files
paths_total A vector of paths of RNA-seq sam files
by         A character, either "gene" or "tx", indicating if rpkms and counts should be summarized at the levels of genes or transcripts. "gene" by default

countMultiMappingReads A logical, if multimapping reads should be counted, FALSE by default. Multimap reads are identified using the tag "NH" in the bam/sam file.

allowMultiOverlap A logical, indicating if a read is allowed to be assigned to more than one feature, FALSE by default

strandSpecific A logical, if strand-specific read counting should be performed, FALSE by default

isPairedEnd A logical, if paired-end reads are used, FALSE by default

Value

A list containing rpkms, counts and the annotation extracted from TxDB for exons and introns

Examples

require(TxDb.Musculus.UCSC.mm9.knownGene)
txdb <- TxDb.Musculus.UCSC.mm9.knownGene
filesFourSU <- system.file('extdata', '4sRNA_0h.bam', package='INSPEcT')
filesTotal <- system.file('extdata', 'totalRNA_0h.bam', package='INSPEcT')
makeRPKMsOut <- makeRPKMs(txdb, filesFourSU, filesTotal)
rpkms <- makeRPKMsOut$rpkms
counts <- makeRPKMsOut$counts
annotation <- makeRPKMsOut$annotation
makeSimDataset

Generate synthetic rates and concentrations

Description

Generate synthetic rates and concentrations

This method generates rates and concentrations where noise is added according to the desired number of replicates that the user set as an arguments from the INSPEcT_model object that has been created by the method of the class INSPEcT makeSimModel. Rates and concentrations can be generated at the time-points of interest. This method generates an INSPEcT object that can be modeled and the performance of the modeling can be tested directly against the INSPEcT_model object created by makeSimModel.

Usage

makeSimDataset(object, tpts, nRep, seed = NULL)

## S4 method for signature 'INSPEcT_model'
makeSimDataset(object, tpts, nRep, seed = NULL)

Arguments

object An object of class INSPEcT_model, usually the output of makeSimModel
tpts A numeric vector of time points where rates and concentrations have to be evaluated
nRep Number of replicates to simulate
seed A numeric to obtain reproducible results

Value

An object of the class ExpressionSet containing rates and concentrations

See Also

makeSimModel

Examples

## generate a synhtic data-set of 10 genes based on the real data-set
data('rpkms', package='INSPEcT')
tpts <- c(0, 1/6, 1/3, 1/2, 1, 2, 4, 8, 16)
tL <- 1/6
mycerIds <- newINSPEcT(tpts, tL, rpkms$foursu_exons, rpkms$total_exons, rpkms$foursu_introns, rpkms$total_introns, BPPARAM=SerialParam())
simRates <- makeSimModel(mycerIds, 10)
simData <- makeSimDataset(simRates, tpts, 1)
## load simulated datasets
data('simRates', package='INSPEcT')
data('simData3rep', package='INSPEcT')
## measure sensitivity/sensibility of synthesis, degradation and processing
## rates identification
Description

Build the synthetic rates shaped on a dataset

This method allow the creation of synthesis, degradation and processing rates for a certain number of genes. The rates are created according to the distributions of the real data-set which is given as an input of the method. Different proportions of constant varying rates can be set and a new vector of time points can be provided. This method has to be used before the `makeSimDataset` method.

Usage

```r
makeSimModel(object, nGenes, newTpts = NULL, probs = c(constant = 0.5, sigmoid = 0.3, impulse = 0.2), na.rm = TRUE, seed = NULL)
```

## S4 method for signature 'INSPEcT'

```r
makeSimModel(object, nGenes, newTpts = NULL, 
probs = c(constant = 0.5, sigmoid = 0.3, impulse = 0.2), na.rm = TRUE, 
seed = NULL)
```

Arguments

- `object`: An object of class INSPEcT
- `nGenes`: A numeric with the number of synthetic genes to be created
- `newTpts`: A numeric vector with time points of the synthetic dataset, if NULL the time points of the real dataset will be used
- `probs`: A numeric vector wich describes the probability of a rate to be constant, shaped like a sigmoid or like an impulse model
- `na.rm`: A logical that set whether missing values in the real dataset should be removed
- `seed`: A numeric to obtain reproducible results

Details

The method `makeSimModel` generates an object of class INSPEcT_model that stores the parametric functions to generate clean rates of a time-course. To any of the rates also a noise variance is associate but not used yet. In a typical workflow the output of `makeSimModel` is the input of the method `makeSimDataset`, that build the noisy rates and concentrations, given a specified number of replicates.

Value

An object of class INSPEcT_model with synthetic rates
**modelingParams**

Get and set number parameters for the modeling

**Description**

Get and set number parameters for the modeling

A method to get and set the parameters that will be used in the modeling of estimated rates and concentrations by the method `modelRates`

**Usage**

```r
modelingParams(object)
modelingParams(object) <- value
```

## S4 method for signature 'INSpecT'

```r
modelingParams(object)
```

## S4 replacement method for signature 'INSpecT'

```r
modelingParams(object) <- value
```
**modelRates**

**Arguments**

- `object`: An object of class `INSPEcT`
- `value`: A list with new parameters

**Value**

List of parameters and their values

- `nInit`: number of optimization to find the best functional representation of each rate (by default 10)
- `nIter`: number of max iteration during optimization (default is 300)
- `na.rm`: A logical whether missing values should be removed from estimated rates (default is `TRUE`)
- `verbose`: A logical whether to be verbose or not (default is `TRUE`)
- `estimateRatesWith`: Either "int" or "der". With "int" the degradation and processing rates are estimated integrating the system between one time point and the following. With "der" degradation and processing rates are estimated using the derivative of total and pre mRNA. (default is "int")
- `useSigmoidFun`: A logical, whether to choose between sigmoid and impulse function to fit rates and concentrations. In case not, always impulse function is used. (default is `TRUE`)
- `testOnSmooth`: A logical, whether models should be tested on smoothed pre-mRNA, total mRNA and synthesis rates or not. (default is `TRUE`)

**See Also**

- `modelRates`

**Examples**

```r
data('mycerIds10', package='INSPEcT')
modelingParams(mycerIds10)
data('mycerIds10', package='INSPEcT')
mycerIds10 <- removeModel(mycerIds10)
modelingParams(mycerIds10)$useSigmoidFun <- FALSE
```

**Description**

Launch the modeling process with parameters set with `modelingParams`

This method model the synthesis, degradation and processsing rates after their estimation by the constructor function `newINSPEcT`. Estimated rates are not guaranteed to optimally describes provided input data yet. To this purpose, modeled rates can be generated and genes can be assigned to a transcriptional regulatory mechanism. Modeled rates can be accessed via the method `viewModelRates` and gene classification according to the regulatory mechanism can be accessed by `geneClass`. The modeling procedure can be set by the user by modifying the parameters via `modelingParams`
modelRates

Usage

modelRates(object, seed = NULL, BPPARAM = bpparam(), verbose = NULL)

## S4 method for signature 'INSPEcT'
modelRates(object, seed = NULL, BPPARAM = bpparam(),
verbose = NULL)

Arguments

object An object of class INSPEcT
seed A numeric, indicating the seed to be set for reproducible results
BPPARAM Parallelization parameters for bplapply. By default bpparam()
verbose Either NULL or logical. If logical indicates whether to output some text during computation or not, if NULL it takes the information from the object (see modelingParams) (Default: NULL) (Default: NULL)

Details

When modeling many genes, parallelization is strongly suggested to reduce computational time. Since all genes run independently, the computational time is divided by the number of cores used/available. However, when modeling more than 500 genes, it may happen that a single gene returns an error that escapes the try/catch controls of INSPEcT. With the parallel mode, the error will propagate on all genes that have been computed with the same processor (or core). To avoid this, the computation could be splitted in chunks and the whole data set can be obtained by combining the chunks (see Examples).

Value

An object of class INSPEcT with modeled rates

See Also

viewModelRates, geneClass, modelingParams

Examples

data('rpkms', package='INSPEcT')
tpts <- c(0, 1/6, 1/3, 1/2, 1, 2, 4, 8, 16)
tL <- 1/6
mycerIds <- newINSPEcT(tpts, tL, rpkms$foursu_exons, rpkms$total_exons,
rpkms$foursu_introns, rpkms$total_introns, BPPARAM=SerialParam())
mycerIdsOneGene <- mycerIds[5]
## View modeling parameters
modelingParams(mycerIdsOneGene)
## Run the modeling in a reproducible way (setting seed)
mycerIdsOneGene <- modelRates(mycerIdsOneGene, seed=1, BPPARAM=SerialParam())
## view modeled synthesis rates
viewModelRates(mycerIdsOneGene, 'synthesis')
## view gene classes
geneClass(mycerIdsOneGene)
## Divide a parallel computation into chunks

```r
nCores(mycerIds) <- parallel::detectCores()
chunkSize <- 100
splitIdx <- ceiling(c(1:nGenes(mycerIds))/chunkSize)
chunks <- lapply(split(mycerIds, splitIdx), modelRates)
mycerIdsModeled <- do.call('combine', chunks)
```

## End(Not run)

---

### Description

With this method, the user can personalize the criteria by which INSPEcT selects a rate to be variable or constant. In particular, the model selection criteria can be selected between log-likelihood ratio test and Akaike’s information criterion. In case log-likelihood ratio test is selected, the thresholds of chi-squared and Brown’s method can be set (see Details section).

### Usage

```r
modelSelection(object)
modelSelection(object) <- value
thresholds(object)
thresholds(object) <- value
llrtests(object)
llrtests(object) <- value
```

---

```r
## S4 method for signature 'INSPEcT'
modelSelection(object)
```

```r
## S4 replacement method for signature 'INSPEcT'
modelSelection(object) <- value
```

```r
## S4 method for signature 'INSPEcT_model'
modelSelection(object)
```

```r
## S4 replacement method for signature 'INSPEcT_model'
modelSelection(object) <- value
```

```r
## S4 method for signature 'INSPEcT'
thresholds(object)
```

```r
## S4 replacement method for signature 'INSPEcT'
thresholds(object) <- value
```
## S4 method for signature 'INSPEcT_model'
thresholds(object)

## S4 replacement method for signature 'INSPEcT_model'
thresholds(object) <- value

## S4 method for signature 'INSPEcT'
llrtests(object)

## S4 replacement method for signature 'INSPEcT'
llrtests(object) <- value

## S4 method for signature 'INSPEcT_model'
llrtests(object)

## S4 replacement method for signature 'INSPEcT_model'
llrtests(object) <- value

### Arguments

- **object**: An object of class INSPEcT or INSPEcT_model
- **value**: A list or a character that will substitute the set of parameters
  - **modelSelection**: A character, either "llr" to test whether a rate is varying using log-likelihood testing framework or "aic" to choose the best model via Akaike Information Criterion (Default: "llr").
  - **thresholds**: A named list containing the threshold that is used to consider a model as accepted in terms of the chi-squared test and three thresholds (one per each rate) that are used to consider a rate as variable using the Brown's method on the log-likelihood ratio tests
  - **llrtests**: A list of three elements that represent, for each rate, the pairs of models that will be compared via log likelihood ratio test to assess whether the rate is variable or not

### Details

When log-likelihood is chosen as a criterion for model selection, different nested models can be compared to assess whether a single rate is varying or constant. For example, in case we want to establish whether synthesis rate is constant or not we can test the null hypothesis "all the rates are constant" against the alternative hypothesis "synthesis rate is changing". The null hypothesis is a special case of the alternative hypothesis, therefore the models are nested. We can also assess whether synthesis rate is constant or not by comparing the null hypothesis "degradation rate is changing" against the alternative hypothesis "degradation and synthesis are changing". The method `llrtests` set the models that are compared to assess the variability of each rate. Different comparisons will be combined using Brown’s method for combining p-values. Models are named with a short notation where synthesis is "a", degradation is "b" and processing is "c". "0" is the model where all genes are kept constant and "ab", for example is the model where synthesis rate and degradation rate are changing. The user can also set the thresholds for Brown’s p-value and chi-squared p-value. While the former set the threshold to assess whether a rate is variable or not over time, the latter set the chi-squared threshold for a pair of model to be used via the log-likelihood ratio test. In order for a pair to be used, at least one model of the pair should have a chi-squared p-value (goodness of fit) below the threshold. The construction of a synthetic data-set can help in the choice of the correct parameters for the test (`makeSimModel`, `makeSimDataset`).

newINSPEcT

Value

See "value"

See Also

makeSimModel, makeSimDataset

Examples

data('mycerIds10', package='INSPEcT')
modelSelection(mycerIds10) <- 'aic'
thresholds(mycerIds10)
thresholds(mycerIds10)$chisquare <- 1e-3
thresholds(mycerIds10)$brown['synthesis'] <- 1e-3
llrtests(mycerIds10)
llrtests(mycerIds10)$synthesis <- list(c('0','a'), c('b','ab'))

mycerIds10  An INSPEcT object with evaluated and modeled rates and concentrations

Description

This INSPEcT object contains the evaluated and modeled rates and concentrations of the very first 10 genes of the dataset rpkms

Format

An INSPEcT object

newINSPEcT  Create a new INSPEcT object

Description

The function newINSPEcT creates a new instance of the class INSPEcT, provided the intronic and exonic RPKMs of RNA- and 4sU-seq experiments at different time points, the time points, the labeling time and optionally the scaling factor to normalize the libraries. The scaling factor to normalize the 4sU-seq libraries can be calculated by the function itself and this way of calculating the scaling factors gives robustness to the estimation of synthesis, degradation and processing rates. In case only exonic RPKMs are provided, only synthesis and degradation rates can be estimated by the function newINSPEcT and the scaling factors to normalize 4sU-seq libraries cannot be calculated. To provide robustness to the procedure the argument labeledSF is strongly suggested in this case.
newINSPEcT

Usage

newINSPEcT(tpts, labeling_time, rpkms_4su_exons, rpkms_total_exons,
     rpkms_4su_introns = NULL, rpkms_total_introns = NULL,
     BPPARAM = bpparam(), totalMedianNorm = TRUE, labeledMedianNorm = FALSE,
     totalSF = NULL, labeledSF = NULL, totalQuantileNorm = FALSE,
     labeledQuantileNorm = FALSE, simulatedData = FALSE,
     degDuringPulse = FALSE)

Arguments

tpts A vector of time points
labeling_time A number, length of the 4sU pulse
rpkms_4su_exons A matrix containing expression levels of 4su exons
rpkms_total_exons A matrix containing expression levels of total exons
rpkms_4su_introns A matrix containing expression levels of 4su introns
rpkms_total_introns A matrix containing expression levels of total introns
BPPARAM Configuration for BiocParallel parallelization. By default is set to bpparam()
totalMedianNorm A logical to perform median normalization over total RNA exons rpkms, it will apply also on introns
labeledMedianNorm A logical to perform median normalization over 4sU RNA exons rpkms, it will apply also on introns
totalSF A vector storing user defined normalization scale over Total RNA exons and introns rpkms
labeledSF A vector storing user defined normalization scale over 4sU RNA exons and introns rpkms
totalQuantileNorm A logical to perform to perform median normalization over total RNA exons rpkms, it will apply also on introns
labeledQuantileNorm A logical to perform to perform median normalization over 4sU RNA exons rpkms, it will apply also on introns
simulatedData A logical, set to TRUE in case the analysis is on simulated data
degDuringPulse A logical, set to TRUE in case of a long labelling time. Also degradation of newly synthesized transcripts will be taken into account

Value

An object of class INSPEcT with rates guessed, rates can be accessed by ratesFirstGuess method.
Examples

```r
data('rpkms', package='INSPEcT')
tpts <- c(0, 1/6, 1/3, 1/2, 1, 2, 4, 8, 16)
tL <- 1/6
mycerIds <- newINSPEcT(tpts, tL, rpkms$foursu_exons, rpkms$total_exons, rpkms$foursu_introns, rpkms$total_introns, BPPARAM=SerialParam())
```

---

## nGenes

*Get the number of genes within the INSPEcT object*

**Description**

Get the number of genes within the INSPEcT object

A method to obtain the number of the genes associated with the object of class INSPEcT

**Usage**

```r
nGenes(object)
```

---

**Arguments**

- `object` An object of class INSPEcT

**Value**

A numeric that indicates the number of genes within the object

**Examples**

```r
data('mycerIds10')
nGenes(mycerIds10)
```

---

## nTpts

*Get the number of time points within the INSPEcT object*

**Description**

Get the number of time points within the INSPEcT object

A method to obtain the number of the tpts associated with the object of class INSPEcT

**Usage**

```r
nTpts(object)
```

---

**Arguments**

- `object` An object of class INSPEcT

**Value**

A numeric that indicates the number of time points within the object

**Examples**

```r
data('mycerIds10')
nTpts(mycerIds10)
```
**plotGene**

**Arguments**

object  
An object of class INSPEcT

**Value**

A numeric that indicates the number of time points contained the object

**Examples**

data('mycerIds10')
nTpts(mycerIds10)

---

**Description**

Plot the pre-modeled and modeled profiles for one gene. A method to see the shapes of the estimated synthesis, degradation and processing rates, pre-mRNA and total mRNA concentrations (solid thin lines) their variances (dashed lines) and the modeled rates and concentrations (ticker solid line) of a single gene.

**Usage**

plotGene(object, ix, fix.yaxis = FALSE)

## S4 method for signature 'INSPEcT'
plotGene(object, ix, fix.yaxis = FALSE)

**Arguments**

object  
An object of class INSPEcT

ix  
Either a rowname or a row number to select one single gene

fix.yaxis  
A logical, indicating whether the limits for y-axis of degradation and processing rates should be fixed relative to their distributions

**Value**

A list containing total mRNA levels and their confidence interval (levels plus and minus one standard deviation), pre-mRNA lelevs and their confidence intervals, synthesis rates and their confidence intervals, degradation rates and processing rates of the selected gene.

**Examples**

data('mycerIds10', package='INSPEcT')
plotGene(mycerIds10, 1)
Description

Visualize the comparison between the rates calculated from two different INSPEcT objects profiled in steady-state conditions.

Usage

```r
## S4 method for signature 'INSPEcT_diffsteady'
plotMA(object, ...)```

Arguments

- `object`: An object of class `INSPEcT_diffsteady`
- `...`: Additional parameters, see Details section

Details

Possible arguments to "plotMA":

- "rate" - A character, which represent the rate to be visualized, either "synthesis", "processing" or "degradation". By default, "synthesis" is chosen.
- "alpha" - A numeric, The confidence interval for significance (FDR), by default 0.1
- "xlim" - A numeric vector of length 2, limits of x-axis, by default the range of the data.
- "xlab" - A character, the label of x-axis, by default "log2 geometric mean"
- "ylim" - A numeric vector of length 2, limits of y-axis, by default the range of the data.
- "ylab" - A character, the label of y-axis, by default "log2 fold change"
- "main" - A character, the title of the plot, by default the name of the visualized rate.

See Also


Examples

```r
data('simData3rep', package='INSPEcT')
data('simData3rep_2', package='INSPEcT')
diffrates <- compareSteady(simData3rep, simData3rep_2)
plotMA(diffrates, alpha=.5)```
ratePvals

Retrieve a single p-value for each rate

Description

Retrieve a single p-value for each rate

This method is used to retrieve all the p-values combined with Brown’s method that combines the results of the log likelihood ratio test results for all pairs tested for each rate and all genes. P-values will change according to the threshold set for the chi-squared test because it influences the model that will be taken into consideration to perform log likelihood ratio tests. To have a sense of the best parameter to choose, a synthetic data-set can be built and tested (makeSimModel, makeSimDataset)

In case 'aic' has been selected via modelSelection method, this method assigns the chi-squared test result of the model selected by AIC to the respective variable rates

Usage

ratePvals(object, cTsh = NULL)

## S4 method for signature 'INSPEcT_model'
ratePvals(object, cTsh = NULL)

## S4 method for signature 'INSPEcT'
ratePvals(object, cTsh = NULL)

Arguments

object An object of class INSPEcT or INSPEcT_model
cTsh A numeric representing the threshold for the chi-squared test to consider a model as valid

Details

ratePvals retrieve a single p-value for each rate thanks to multiple log likelihood tests performed on nested models that has a chi-squared test below the selected threshold. Among the many p-values that log likelihood ratio test calculate, a single p-value is obtained applying Brown’s method for combining dependent p-values.

Value

A matrix containing p-values calculated for each rate

See Also

makeSimModel, makeSimDataset

Examples

data('mycerIds10', package='INSPEcT')
ratePvals(mycerIds10)

# calculate again the p-values with Brown with a different threshold
# for considering a model valid for the log likelihood ratio test
ratesFirstGuess

ratePvals(mycerIds10, cTsh=.2)
# Set permanently the chi-squared threshold at .2 for mycerIds10 object
thresholds(mycerIds10)$chisquare <- .2

ratesFirstGuess

Retrieve pre-modeling rates and concentrations

Description

Retrieve pre-modeling rates and concentrations

This method allows to access to the estimated synthesis, degradation, processing rates and pre mRNA and total mRNA concentrations the way they were calculated by the constructor function newINSPEcT.

Usage

ratesFirstGuess(object, feature)

## S4 method for signature 'INSPEcT'
ratesFirstGuess(object, feature)

Arguments

object

An object of class INSPEcT

feature

A character indicating the feature to retrieve, "synthesis", "degradation", "processing" for rates, "total" for total mRNA concentrations or "preMRNA" for premature mRNA concentrations

Value

A numeric matrix containing the values for the selected feature

See Also

newINSPEcT, ratesFirstGuessVar

Examples

data('rpkmws', package='INSPEcT')
tpts <- c(0, 1/6, 1/3, 1/2, 1, 2, 4, 8, 16)
tL <- 1/6
mycerIds <- newINSPEcT(tpts, tL, rpkmws$foursu_exons, rpkmws$total_exons, rpkmws$foursu_introns, rpkmws$total_introns, BPPARAM=SerialParam())
# get estimated synthesis rates
ratesFirstGuess(mycerIds, 'synthesis')
ratesFirstGuessVar

Retrieve pre-modeling rates and concentrations variance

Description
Retrieve pre-modeling rates and concentrations variance
This method allows to access to the estimated variance of synthesis rates and pre mRNA and total mRNA concentrations the way they were calculated by the constructor function `newINSPEcT`.

Usage
`ratesFirstGuessVar(object, feature)`

## S4 method for signature 'INSPEcT'
ratesFirstGuessVar(object, feature)

Arguments
- `object`: An object of class INSPEcT
- `feature`: A character indicating the feature to retrieve, "synthesis", "degradation", "processing" for rates, "total" for total mRNA concentrations or "preMRNA" for premature mRNA concentrations

Value
A numeric vector containing the values for the selected feature

See Also
`newINSPEcT, ratesFirstGuess`

Examples
```r
data('rpkm$s', package='INSPEcT')
tpts <- c(0, 1/6, 1/3, 1/2, 1, 2, 4, 8, 16)
tL <- 1/6
mycerIds <- newINSPEcT(tpts, tL, rpkm$s$foursu_exons, rpkm$s$total_exons, rpkm$s$foursu_introns, rpkm$s$total_introns, BPPARAM=SerialParam())
ratesFirstGuessVar(mycerIds, 'synthesis')
```

removeModel
remove modelling information from INSPEcT object

Description
remove modelling information from INSPEcT object
Remove the model from an INSPEcT object. It is required when subsetting an INSPEcT object per time points because when removing time points the modeling is not valid anymore.
rocCurve

Usage
removeModel(object)

### S4 method for signature 'INSPeCT'
removeModel(object)

Arguments
object An Object of class INSPeCT

Value
An Object of class INSPeCT

Examples

```r
data('mycerIds10', package='INSPeCT')
mycerIds_5genes <- mycerIds10[1:5]

## This will turn out into an error:
## Not run: mycerIds_5genes_5tpts <- mycerIds10[1:5, 1:5]

## Before subsetting time points, the model should be removed:
mycerIds_5genes_5tpts <- removeModel(mycerIds10)[1:5, 1:5]

## This will turn out into an error:
## Not run: mycerIds10 <- modelRates(mycerIds10)

## Before running the model again, or changing modeling parameters,
## the previous model should be removed:
mycerIds10_old <- mycerIds10
mycerIds10_new <- removeModel(mycerIds10)
modelingParams(mycerIds10_new)$useSigmoidFun <- FALSE
## Not run: mycerIds10_new <- modelRates(mycerIds10_new)
```

Description

Display rate classification performance

A method to visualize the performance in the classification of synthesis, degradation and processing rates based on the comparison of the original simulated rates and the one obtained by the function `modelRates`. For each rate, classification performance is measured in terms of sensitivity and specificity using a ROC curve analysis. False negatives (FN) represent cases where the rate is identified as constant while it was simulated as varying. False positives (FP) represent cases where INSPeCT identified a rate as varying while it was simulated as constant. On the contrary, true positives (TP) and negatives (TN) are cases of correct classification of varying and constant rates, respectively. Consequently, sensitivity and specificity are computed using increasing thresholds for the brown p-values, and the ability of correctly classifying a rate is measured through the area under the curve (AUC) for each rate.
rocThresholds

Usage

rocCurve(object, object2, cTsh = NULL, plot = TRUE)

## S4 method for signature 'INSPEcT_model,INSPEcT_model'
rocCurve(object, object2, cTsh = NULL, plot = TRUE)

## S4 method for signature 'INSPEcT_model,INSPEcT'
rocCurve(object, object2, cTsh = NULL, plot = TRUE)

Arguments

- object: An object of class INSPEcT_model, with true rates
- object2: An object of class INSPEcT or INSPEcT_model, with modeled rates
- cTsh: A numeric representing the threshold for the chi-squared test to consider a model as valid; if NULL the value is taken from the INSPEcT_model object
- plot: A logical indicating whether ROC curves should be plotted or not

Value

A list of objects of class pROC with summary of each roc curve

See Also

makeSimModel, makeSimDataset, rocThresholds

Examples

data('simRates', package='INSPEcT')
data('simData3rep', package='INSPEcT')
rocCurve(simRates, simData3rep)

---

rocThresholds  Display rate classification performance with thresholds visible at x-axis

Description

Display rate classification performance with thresholds visible at x-axis

A method to visualize the performance in the classification of synthesis, degradation and processing rates based on the comparison of the original simulated rates and the one obtained by the function modelRates. For each rate, classification performance is measured in terms of sensitivity and specificity using a ROC curve analysis. False negatives (FN) represent cases where the rate is identified as constant while it was simulated as varying. False positives (FP) represent cases where INSPEcT identified a rate as varying while it was simulated as constant. On the contrary, true positives (TP) and negatives (TN) are cases of correct classification of varying and constant rates, respectively. Consequently, at increasing brown p-values different sensitivity and specificity can be achieved.
Usage

rocThresholds(object, object2, cTsh = NULL, bTsh = NULL, xlim = c(1e-05, 1))

## S4 method for signature 'INSPEcT_model,INSPEcT_model'
rocThresholds(object, object2,
               cTsh = NULL, bTsh = NULL, xlim = c(1e-05, 1))

## S4 method for signature 'INSPEcT_model,INSPEcT'
rocThresholds(object, object2, cTsh = NULL,
               bTsh = NULL, xlim = c(1e-05, 1))

Arguments

object  An object of class INSPEcT_model, with true rates
object2 An object of class INSPEcT or INSPEcT_model, with modeled rates
cTsh    A numeric representing the threshold for the chi-squared test to consider a model as valid
bTsh    A numeric representing the threshold for the Brown’s method to consider a rate as varying
xlim    A numeric representing limits for the x-axis (default is c(1-e-5,1))

Value

None

See Also

makeSimModel, makeSimDataset, rocCurve

Examples

data('simRates', package='INSPEcT')
data('simData3rep', package='INSPEcT')
rocThresholds(simRates, simData3rep)
# Increase the Brown threshold for all rates (be more relaxed)
thresholds(simData3rep)$brown <- c(alpha=.05, beta=.05, gamma=.05)
rocThresholds(simRates, simData3rep)

rpkms  intron, exons RPKM values of 500 genes from 4sU-seq and RNA-seq experiments

Description

A dataset containing the values of exonic and intronic RPKMs of 500 genes both in 4sU-seq and RNA-seq experiments. The variables are as follows:

Format

A list of 4 matrices with 500 rows and 9 columns
Details

- rpkms_4su_exons
- rpkms_4su_introns
- rpkms_total_exons
- rpkms_total_introns

Description

A nice plot to see scaling factors used for RNA-seq and 4sU-seq libraries

This method generates a plot that immediately shows the scaling factors used to scale RNA- and 4sU-seq libraries and the possible relations between them. The ratio between the RNA- and the 4sU-seq scaling can be in fact considered as a yield of the synthesis within the cells.

Usage

sfPlot(object)

## S4 method for signature 'INSPEcT'
sfPlot(object)

Arguments

object

An object of class INSPEcT

Value

None

Examples

data('rpkms', package='INSPEcT')
tpts <- c(0, 1/6, 1/3, 1/2, 1, 2, 4, 8, 16)
tL <- 1/6
mycerIds <- newINSPEcT(tpts, tL, rpkms$foursu_exons, rpkms$total_exons, rpkms$foursu_introns, rpkms$total_introns, BPPARAM=SerialParam())
sfPlot(mycerIds)
<table>
<thead>
<tr>
<th>simData1rep</th>
<th>An INSPEcT object with 1000 simulated rates and concentration and their modeled rates</th>
</tr>
</thead>
</table>

**Description**

A dataset containing the rates and concentrations obtained from the dataset simRates with 1 replicates and time points corresponding to: 0, 1/6, 1/3, 1/2, 1, 2, 4, 8, 16 hours. On this dataset rates and concentrations have been modeled with the method modelRates

**Format**

An INSPEcT object

<table>
<thead>
<tr>
<th>simData3rep</th>
<th>An INSPEcT object with 1000 simulated rates and concentration and their modeled rates</th>
</tr>
</thead>
</table>

**Description**

A dataset containing the rates and concentrations obtained from the dataset simRates with 3 replicates and time points corresponding to: 0, 1/6, 1/3, 1/2, 1, 1.5, 2, 4, 8, 12, 16 and 24 hours. On this dataset rates and concentrations have been modeled with the method modelRates

**Format**

An INSPEcT object

<table>
<thead>
<tr>
<th>simRates</th>
<th>An INSPEcT_model object of 1000 simulated rates</th>
</tr>
</thead>
</table>

**Description**

The INSPEcT_model object contains 1000 simulated rates that were obtained using the dataset rpkms as reference

**Format**

An INSPEcT_model object
**split**

*Divide an INSPEcT Object into groups*

**Description**

Divides the INSPEcT object into the groups defined by `f`.

**Usage**

```r
## S4 method for signature 'INSPEcT,ANY'
split(x, f, drop = FALSE, ...)
```

**Arguments**

- `x`: An object of class INSPEcT
- `f`: A vector of length equal to the number of genes in `x` which defines the groups
- `drop`: A logical belonging to the generic function, useless in this context.
- `...`: Additional arguments to match the generic function

**Value**

A list containing objects of class INSPEcT

**Examples**

```r
data('mycerIds10')
splitIdx <- c(1,1,1,2,2,2,3,3,3,4)
mycerIds10Split <- split(mycerIds10, splitIdx)
```

---

**totalSF**

*Accessor to the slot totalSF of an INSPEcT object*

**Description**

Accessor to the slot totalSF of an INSPEcT object

**Usage**

```r
totalSF(object)
```

**Arguments**

- `object`: An object of class INSPEcT
Value
A numeric that indicates the scaling factors applied between time points of the data coming from RNA-seq library (applies to total-mRNAs and pre-mRNAs concentrations)

Examples
```r
data('mycerIds10')
totalSF(mycerIds10)
```

---

**tpts**  
*Accessor to the slot tpts of an INSPEcT object*

---

Description
Accessor to the slot tpts of an INSPEcT object
Accessor to obtain the tpts associated with the object of class INSPEcT

Usage
```
tpts(object)
```

## S4 method for signature 'INSPEcT'
tpts(object)

Arguments
- **object**  
  An object of class INSPEcT

Value
A numeric that indicates time points contained the object

Examples
```r
data('mycerIds10')
tpts(mycerIds10)
```

---

UCSC_mm9_genes_exons  
*A Gtf file containing exons definition of 100 genes*

Description
A Gtf file containing exons definition of 100 genes

Format
A tab separated file
Description

A Gtf file containing introns definition of 100 genes

Format

A tab separated file

viewModelRates

Retrieve the modeled rates and concentrations

Description

Retrieve the modeled rates and concentrations

A method to access the modeled rates via the method `modelRates`

Usage

`viewModelRates(object, feature)`

```
## S4 method for signature 'INSPEcT'
viewModelRates(object, feature)
```

Arguments

- `object` An object of class INSPEcT
- `feature` A character indicating the feature to retrieve, "synthesis", "degradation", "processing" for rates, "total" for total mRNA concentrations or "preMRNA" for premature mRNA concentrations

Value

A numeric matrix containing the values for the selected feature

Examples

```r
data('mycerIds10', package='INSPEcT')
viewModelRates(mycerIds10, 'synthesis')
```
makeSimModel, 12, 14, 19, 20, 20, 25, 26, 31, 35, 36
makeSimModel, INSPEcT-method
(makeSimModel), 20
modelingParams, 11, 12, 21, 22, 23
modelingParams, INSPEcT-method
(modelingParams), 21
modelingParams <- (modelingParams), 21
modelingParams <-, INSPEcT-method
(modelingParams), 21
modelRates, 12, 14, 17, 21, 22, 23, 34, 35, 41
modelRates, INSPEcT-method (modelRates), 22
modelSelection, 12, 15, 24, 31
modelSelection, INSPEcT-method
(modelSelection), 24
modelSelection, INSPEcT_model-method
(modelSelection), 24
modelSelection <- (modelSelection), 24
modelSelection <-, INSPEcT-method
(modelSelection), 24
modelSelection <-, INSPEcT_model-method
(modelSelection), 24
mycerIds10, 26
newINSPEcT, 11, 22, 26, 32, 33
nGenes, 6, 12, 28
nGenes, INSPEcT-method (nGenes), 28
nTpts, 6, 12, 28
nTpts, INSPEcT-method (nTpts), 28
plotGene, 12, 29
plotGene, INSPEcT-method (plotGene), 29
plotMA, 14, 30
plotMA, INSPEcT_diffsteady-method
(plotMA), 30
processing (INSPEcT_diffsteady-class), 13
processing, INSPEcT_diffsteady-method
(INSPEcT_diffsteady-class), 13
ratePvals, 9, 12, 15, 31
ratePvals, INSPEcT-method (ratePvals), 31
ratePvals, INSPEcT_model-method
(ratePvals), 31
ratesFirstGuess, 12, 27, 32, 33
ratesFirstGuess, INSPEcT-method
(ratesFirstGuess), 32
ratesFirstGuessVar, 12, 32, 33
ratesFirstGuessVar, INSPEcT-method
(ratesFirstGuessVar), 33
removeModel, 12, 33
removeModel, INSPEcT-method
(removeModel), 33
rocCurve, 15, 34, 36
rocCurve, INSPEcT_model, INSPEcT-method
(rocCurve), 34
rocCurve, INSPEcT_model, INSPEcT_model-method
(rocCurve), 34
rocThresholds, 15, 35, 35
rocThresholds, INSPEcT_model, INSPEcT-method
(rocThresholds), 35
rocThresholds, INSPEcT_model, INSPEcT_model-method
(rocThresholds), 35
rpkms, 36
sfPlot, 12, 37
sfPlot, INSPEcT-method (sfPlot), 37
show, INSPEcT-method (INSPEcT-class), 11
show, INSPEcT_diffsteady-method
(INSPEcT_diffsteady-class), 13
show, INSPEcT_model-method
(INSPEcT_model-class), 14
simData1rep, 38
simData3rep, 38
simRates, 38
split, 39
split, INSPEcT, ANY-method (split), 39
synthesis (INSPEcT_diffsteady-class), 13
synthesis, INSPEcT_diffsteady-method
(INSPEcT_diffsteady-class), 13
testingParams, 10
testingParams (modelSelection), 24
thresholds, 12, 15
thresholds (modelSelection), 24
thresholds, INSPEcT-method
(modelSelection), 24
thresholds, INSPEcT_model-method
(modelSelection), 24
thresholds <- (modelSelection), 24
thresholds <-, INSPEcT-method
(modelSelection), 24
thresholds <-, INSPEcT_model-method
(modelSelection), 24
totalSF, 12, 39
totalSF, INSPEcT-method (totalSF), 39
tpts, 12, 40
tpts, INSPEcT-method (tpts), 40
UCSC_mm9_genes_exons, 40
UCSC_mm9_genes_introns, 41
viewModelRates, 12, 22, 23, 41
viewModelRates, INSPEcT-method
(viewModelRates), 41