

Package ‘INSPEcT’

April 16, 2019

Type Package

Title Analysis of 4sU-seq and RNA-seq time-course data

Version 1.12.2

Date 2018-12-06

Author Stefano de Pretis

Maintainer Stefano de Pretis <ste.depo@gmail.com>

Description INSPEcT (INference of Synthesis, Processing and dEgradation rates from Transcriptomic data) RNA-seq data in time-course experiments or different conditions, with or without the support of nascent RNA data.

License GPL-2

Depends R (>= 3.2), methods, Biobase, BiocParallel

Imports pROC, deSolve, rootSolve, compiler, preprocessCore, GenomicFeatures, GenomicRanges, IRanges, BiocGenerics, GenomicAlignments, Rsamtools, S4Vectors, DESeq2, plgem, SummarizedExperiment, TxDb.Mmusculus.UCSC.mm9.knownGene, shiny

Suggests BiocStyle, knitr

VignetteBuilder knitr

biocViews Sequencing, RNASeq, GeneRegulation, TimeCourse, SystemsBiology

NeedsCompilation no

RoxygenNote 6.1.1

git_url <https://git.bioconductor.org/packages/INSPEcT>

git_branch RELEASE_3_8

git_last_commit 4bd62ca

git_last_commit_date 2019-04-04

Date/Publication 2019-04-15

R topics documented:

AIC-INSPEcT-method	3
allcounts	3
chisqmodel	4

chisqtest	4
combine	5
compareSteady	6
compareSteadyNoNascent	7
dim,INSPEcT-method	8
Extract	8
featureNames,INSPEcT-method	9
featureWidths	10
geneClass	10
getModel	11
inHeatmap	12
INSPEcT	13
INSPEcT-class	13
INSPEcT-shinyApps	15
INSPEcT_diffsteady-class	15
INSPEcT_model-class	17
labeledSF	18
libsizes	19
logLik	19
makeModelRates	20
makeSimDataset	21
makeSimModel	22
modelingParams	23
modelRates	24
modelSelection	25
newINSPEcT	27
nGenes	28
nTpts	29
plotGene	29
plotMA	30
processingDelay	31
quantifyExpressionsFromBAMs	33
quantifyExpressionsFromTrAbundance	34
quantifyExpressionsFromTrCounts	35
ratePvals	36
ratesFirstGuess	37
ratesFirstGuess Var	38
removeModel	39
rocCurve	40
rocThresholds	41
sfPlot	42
simData3rep_Nascent	43
simData3rep_NoNascent	43
simData4rep_Nascent	44
simData4rep_NoNascent	44
split	44
tpts	45
viewModelRates	45
Index	47

AIC-INSPEcT-method	<i>Akaike information criterion calculated for the models evaluated by INSPEcT</i>
--------------------	------------------------------------------------------------------------------------

Description

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

Usage

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

## 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

```
nascentInspObj10 <- readRDS(system.file(package='INSPEcT', 'nascentInspObj10.rds'))
AIC(nascentInspObj10)
```

allcounts	<i>A list containing mature and nascent counts for exons and introns, three replicates and 11 time points: 0,1/6,1/3,1/2,1,1.5,2,4,8,12,16 hours.</i>
-----------	-------------------------------------------------------------------------------------------------------------------------------------------------------

Description

A list containing mature and nascent counts for exons and introns, three replicates and 11 time points: 0,1/6,1/3,1/2,1,1.5,2,4,8,12,16 hours.

Format

A list of 4 matrices 500 x 33

chisqmodel	<i>Retrieve results of chi-squared test for the selected models</i>
------------	---------------------------------------------------------------------

Description

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

```
chisqmodel(object, ...)

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

## 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

```
nascentInspObj10 <- readRDS(system.file(package='INSPEcT', 'nascentInspObj10.rds'))
chisqmodel(nascentInspObj10)
```

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

Description

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

```
nascentInspObj10 <- readRDS(system.file(package='INSPEcT', 'nascentInspObj10.rds'))
chisqtest(nascentInspObj10)
```

combine

Combine different Objects of Class INSPEcT

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

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

```
nascentInspObj10 <- readRDS(system.file(package='INSPEcT', 'nascentInspObj10.rds'))
nascentInspObj10_2genes <- nascentInspObj10[1:2]
nascentInspObj10_5genes <- nascentInspObj10[6:10]
nascentInspObj10_7genes <- combine(nascentInspObj10_2genes, nascentInspObj10_5genes)
```

compareSteady	<i>Generate an object of class <code>INSPEcT_diffsteady</code> from an object of class <code>INSPEcT</code></i>
---------------	-----------------------------------------------------------------------------------------------------------------

Description

This method compares two object of class `INSPEcT` in order to identify differential usage of synthesis, 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(inspectIds, BPPARAM = bpparam())
```

```
## S4 method for signature 'INSPEcT'
compareSteady(inspectIds, BPPARAM = bpparam())
```

Arguments

<code>inspectIds</code>	An object of calss <code>INSPEcT</code> with two conditions
<code>BPPARAM</code>	Configuration for <code>BiocParallel</code> parallelization. By default is set to <code>bpparam()</code>

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

```
if( Sys.info()["sysname"] != "Windows" ) {
  data('allcounts', package='INSPEcT')
  data('featureWidths', package='INSPEcT')
  data('libsizes', package='INSPEcT')

  nascentCounts<-allcounts$nascent
  matureCounts<-allcounts$mature
  conditions<-letters[1:11]
  expDes<-rep(conditions,3)
  tL<-1/6

  nasExp_DESeq2<-quantifyExpressionsFromTrCounts(
    allcounts=nascentCounts
    ,libsize=nascentLS
    ,exonsWidths=exWdths
    ,intronsWidths=intWdths
    ,experimentalDesign=expDes)

  matExp_DESeq2<-quantifyExpressionsFromTrCounts(
    allcounts=matureCounts
    ,libsize=totalLS
    ,exonsWidths=exWdths
```

```

      ,intronsWidths=intWdths
      ,experimentalDesign=expDes)

nasFullObj <- newINSPECT(
  tpts=conditions
  ,labeling_time=tL
  ,nascentExpressions=nasExp_DESeq2
  ,matureExpressions=matExp_DESeq2)

diffrates = compareSteady(nasFullObj[,c(1,11)])
}

```

```
compareSteadyNoNascent
```

Compare mature RNA steady state data from an object of class INSPECT

Description

This function compare exons and introns expression level matrices, from two up to an arbitrary number of samples, in order to identify genes which are oddly regulated, compared to an expected standard behaviour, from the post transcriptional point of view.

Usage

```
compareSteadyNoNascent(inspectIds, expressionThreshold = 0.25,
  log2FCThreshold = 2)
```

```
## S4 method for signature 'INSPECT'
compareSteadyNoNascent(inspectIds,
  expressionThreshold = 0.25, log2FCThreshold = 2)
```

Arguments

```
inspectIds      An object of class INSPECT.
expressionThreshold
                A parameter which sets how many log2 fold changes of distance from the median behaviour are imputable to noise.
log2FCThreshold
                A parameter which sets the log2 fold change distance from the median behaviour that is imputable to noise.
```

Examples

```

data('allcounts', package='INSPECT')
data('featureWidths', package='INSPECT')
data('libsizes', package='INSPECT')

nascentCounts<-allcounts$nascent
matureCounts<-allcounts$mature
conditions<-letters[1:11]
expDes<-rep(conditions,3)

```

```

matExp_DESeq2<-quantifyExpressionsFromTrCounts(
  allcounts=matureCounts
  ,libsize=totalLS
  ,exonsWidths=exWdths
  ,intronsWidths=intWdths
  ,experimentalDesign=expDes)

matureInspObj <- newINSPEcT(tpts=conditions,matureExpressions=matExp_DESeq2)

regGenes<-compareSteadyNoNascent(inspectIds=matureInspObj
  ,expressionThreshold=0.25
  ,log2FCThreshold=2.)
head(regGenes)
table(regGenes)

```

dim,INSPEcT-method *Dimensions of an Object of Class INSPEcT*

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

```
## 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 *Extract Parts of an INSPEcT or an INSPEcT_model Object*

Description

Operators acting on INSPEcT, INSPEcT_model or INSPEcT_diffsteady 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,ANY'
x[i]

## S4 method for signature 'INSPEcT,ANY,ANY,ANY'
x[i, j]

## S4 method for signature 'INSPEcT_diffsteady,ANY,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

```
nascentInspObj10 <- readRDS(system.file(package='INSPEcT', 'nascentInspObj10.rds'))
nascentInspObj10_5genes <- nascentInspObj10[1:5]
## Not run:
## This will turn out into an error:
nascentInspObj10_5genes_5tpts <- nascentInspObj10[1:5, 1:5]

## End(Not run)
## Before subsetting time points, the model should be removed:
nascentInspObj10_5genes_5tpts <- removeModel(nascentInspObj10)[1:5, 1:5]
```

featureNames,INSPEcT-method

Gene Names Associated with an Object of Class INSPEcT

Description

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

Usage

```
## 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

featureWidths	<i>Contains two variables: "exWdths" and "intWdths" containing the length of the exons and introns, respectively, relative to the genes in "allcounts"</i>
---------------	------------------------------------------------------------------------------------------------------------------------------------------------------------

Description

Contains two variables: "exWdths" and "intWdths" containing the length of the exons and introns, respectively, relative to the genes in "allcounts"

Format

numeric vector of length 500

geneClass	<i>Retrieve the regulatory class for each gene</i>
-----------	----------------------------------------------------

Description

This method returns a factor that summarise 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

```
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)

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

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

```
nascentInspObj10 <- readRDS(system.file(package='INSPEcT', 'nascentInspObj10.rds'))
geneClass(nascentInspObj10)
# see the classification with another threshold for chi-squared test
geneClass(nascentInspObj10, cTsh=.2)
# set the new threshold permanently within the object
modelSelection(nascentInspObj10)$thresholds$chisquare <- .2
```

getModel

Get or replace INSPEcT_model object within INSPEcT object

Description

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

Value

An object of class INSPEcT model

Examples

```
nascentInspObj10 <- readRDS(system.file(package='INSPEcT', 'nascentInspObj10.rds'))
getModel(nascentInspObj10)
```

inHeatmap

Heatmap that represent the fold changes of all the five features

Description

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

Usage

```
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)
```

```
## S4 method for signature 'INSPEcT'
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-RNA is displayed instead of total-RNA (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-RNA/mature-RNA; 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 RNA levels, pre-RNA levels, synthesis rates, degradation rates and processing rates. Matrices are ordered according to the clustering.

Examples

```
nascentInspObj10 <- readRDS(system.file(package='INSPEcT', 'nascentInspObj10.rds'))
inHeatmap(nascentInspObj10, 'pre-model')
inHeatmap(nascentInspObj10, 'model')
```

INSPEcT

*INSPEcT package***Description**

INSPEcT (INference of Synthesis, Processing and dEgradation rates from Transcriptome data), is a package that analyse RNA-seq data in order to evaluate synthesis, processing and degradation rates and asses via modeling the rates that determines changes in RNA 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')
```

INSPEcT-class

*An S4 class to contain all rates and models generated by 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 performed (slot `tpts`) and, if provided, the nascent RNA collecting time (slot `tL`) and the normalization scale factors used for nascent (labeledSF) RNA-seq libraries. 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 logLik
makeModelRates
makeSimModel
modelingParams<-
modelingParams
modelRates
modelSelection
nGenes
nTpts
plotGene
ratePvals
ratesFirstGuessVar
ratesFirstGuess
removeModel
sfPlot
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)

ratesFirstGuessVar An object of class ExpressionSet that contains the variances related to rates and concentrations guessed from the first part of INSPEcT analysis (before modeling)

ratesFirstGuessP An object containing the rates first guess part of INSPEcT analysis (before modeling)

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

model 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 Nascent-seq libraries.

tL A numeric containing the length of the Nascent pulse.

NoNascent A logical indicating if the nascent RNA was included into the analysis.

degDuringPulse A logical indicating if degradation of RNA during the 4sU pulse was considered.

INSPEcT-shinyApps

Run shiny applications contained in the package INSPEcT

Description

Two shiny apps are encoded into the package inspect: - runProcessingRateDelay: plots single genes as well as genome wide plots associated to the processing induced delay loading the data from an INSPEcT object. - runRNAdynamics: is a way to visualize and interact with the RNAdynamics at the level of a single gene, either loading the data from an INSPEcT object or from scratch.

Usage

```
runProcessingRateDelay()

runRNAdynamics()
```

Examples

```
# runProcessingRateDelay()
# runRNAdynamics()
```

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)

## S4 method for signature 'INSPEcT_diffsteady'
featureNames(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 fuction 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

`modeling_res` A `data.frame` which contains modeling results

Examples

```

if( Sys.info()["sysname"] != "Windows" ) {
  data('allcounts', package='INSPEcT')
  data('featureWidths', package='INSPEcT')
  data('libsizes', package='INSPEcT')

  nascentCounts<-allcounts$nascent
  matureCounts<-allcounts$mature
  conditions<-letters[1:11]
  expDes<-rep(conditions,3)
  tL<-1/6

  nasExp_DESeq2<-quantifyExpressionsFromTrCounts(
    allcounts=nascentCounts
    ,libsize=nascentLS
    ,exonsWidths=exWdths
    ,intronsWidths=intWdths
    ,experimentalDesign=expDes)

  matExp_DESeq2<-quantifyExpressionsFromTrCounts(
    allcounts=matureCounts
    ,libsize=totalLS
    ,exonsWidths=exWdths
    ,intronsWidths=intWdths
    ,experimentalDesign=expDes)

  nasFullObj <- newINSPEcT(tpts=conditions,labeling_time=tL
    ,nascentExpressions=nasExp_DESeq2,matureExpressions=matExp_DESeq2)

  diffrates = compareSteady(nasFullObj[,c(1,11)])
  head(synthesis(diffrates))
}
if( Sys.info()["sysname"] != "Windows" ) {
  head(processing(diffrates))
}
if( Sys.info()["sysname"] != "Windows" ) {
  head(degradation(diffrates))
}
if( Sys.info()["sysname"] != "Windows" ) {
  featureNames(diffrates)
}

```

INSPEcT_model-class *An S4 class to represent models generated by INSPEcT*

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 wheter 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

```
## 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
logLik
makeModelRates
makeSimDataset
modelSelection
ratePvals
rocCurve
rocThresholds
```

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.

labeledSF

Accessor to the slot labeledSF of an INSPEcT object

Description

Accessor to obtain the labeledSF slot associated with the object of class INSPEcT

Usage

```
labeledSF(object)

## S4 method for signature 'INSPEcT'
labeledSF(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 Nascent-seq library (applies directly to synthesis rates and indirectly to degradation rates)

Examples

```
nascentInspObj10 <- readRDS(system.file(package='INSPEcT', 'nascentInspObj10.rds'))
labeledSF(nascentInspObj10)
```

libsizes	<i>Contains two variables: "nascentLS" and "totalLS" containing the sequencing depth of nascent and total libraries respectively, relative to the experiments in "allcounts"</i>
----------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Description

Contains two variables: "nascentLS" and "totalLS" containing the sequencing depth of nascent and total libraries respectively, relative to the experiments in "allcounts"

Format

numeric vector of length 33

logLik	<i>Retrieve results of log likelihood test</i>
--------	------------------------------------------------

Description

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

```
nascentInspObj10 <- readRDS(system.file(package='INSPEcT', 'nascentInspObj10.rds'))
logLik(nascentInspObj10)
```

makeModelRates

Calculate modeled rates and concentrations

Description

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

```
makeModelRates(object, ...)

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

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

Arguments

object	An object of class <code>INSPEcT_model</code>
...	additional arguments <code>tpts</code> : 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

```
nascentInspObj10 <- readRDS(system.file(package='INSPEcT', 'nascentInspObj10.rds'))
tpts <- tpts(nascentInspObj10)
eSet <- makeModelRates(getModel(nascentInspObj10), tpts=tpts)
exprs(eSet)
nascentInspObj10 <- readRDS(system.file(package='INSPEcT', 'nascentInspObj10.rds'))
viewModelRates(nascentInspObj10, 'degradation')
## force every degradation rate to be accepted as variable
modelSelection(getModel(nascentInspObj10))$thresholds$brown <- c(synthesis=.01, degradation=1, processing=.01)
nascentInspObj10 <- makeModelRates(nascentInspObj10)
viewModelRates(nascentInspObj10, 'degradation')
```

makeSimDataset	<i>Generate synthetic rates and concentrations</i>
----------------	----------------------------------------------------

Description

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 within the original time window. 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, NoNascent = FALSE, seed = NULL)

## S4 method for signature 'INSPECT_model'
makeSimDataset(object, tpts, nRep,
  NoNascent = FALSE, 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
NoNascent	A logical which, if true, makes the output of the method suitable for an analysis without Nascent
seed	A numeric to obtain reproducible results

Value

An object of the class ExpressionSet containing rates and concentrations

See Also

[makeSimModel](#)

Examples

```
if( Sys.info()["sysname"] != "Windows" ) {
  nascentInspObj <- readRDS(system.file(package='INSPECT', 'nascentInspObj.rds'))
  simRates<-makeSimModel(nascentInspObj, 1000, seed=1)
  tpts <- tpts(nascentInspObj)
  nascentSim2replicates <- makeSimDataset(object=simRates,tpts=tpts,nRep=3,NoNascent=FALSE,seed=1)
}
```

makeSimModel *Build the synthetic rates shaped on a dataset*

Description

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

```
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'
makeSimModel(object, nGenes, 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 synthtic genes to be created
newTpts	A numeric verctor with time points of the synthtic 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 genrate 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

See Also

[makeSimDataset](#)

Examples

```
nascentInspObj <- readRDS(system.file(package='INSPEcT', 'nascentInspObj.rds'))
simRates<-makeSimModel(nascentInspObj, 1000, seed=1)
table(geneClass(simRates))
```

modelingParams	<i>Get and set number parameters for the modeling</i>
----------------	-------------------------------------------------------

Description

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

```
modelingParams(object)

modelingParams(object) <- value

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

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

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 wheter missing values should be removed from estimated rates (default is TRUE)
- verbose A logical wheter 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, wheter models should be tested on smoothed pre-mRNA, total mRNA and eventually synthesis rates or not. (default is TRUE)

See Also

[modelRates](#)

Examples

```
nascentInspObj10 <- readRDS(system.file(package='INSPECT', 'nascentInspObj10.rds'))
modelingParams(nascentInspObj10)
nascentInspObj10 <- readRDS(system.file(package='INSPECT', 'nascentInspObj10.rds'))
nascentInspObj10 <- removeModel(nascentInspObj10)
modelingParams(nascentInspObj10)$useSigmoidFun <- FALSE
```

modelRates

Launch the modeling process

Description

Launch the modeling process with parameters set with [modelingParams](#)

This method model the synthesis, degradation and processing 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 changing the parameters via [modelingParams](#)

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

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

```
if( Sys.info()["sysname"] != "Windows" ) {
nascentInspObj10 <- readRDS(system.file(package='INSPEcT', 'nascentInspObj10.rds'))
## models removal
nascentInspObjThreeGenes <- removeModel(nascentInspObj10[1:3])
nascentInspObjThreeGenes <- modelRates(nascentInspObjThreeGenes, seed=1, BPPARAM=SerialParam())
## view modeled synthesis rates
viewModelRates(nascentInspObjThreeGenes, 'synthesis')
## view gene classes
geneClass(nascentInspObjThreeGenes)
}
```

modelSelection

Get or set parameters for model test and selection

Description

With this methods 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 (AIC). In case log-likelihood ratio test is selected, the thresholds of chi-squared and Brown's method can be set (see Details section).

Usage

```
modelSelection(object)

modelSelection(object) <- value

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

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

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

## S4 replacement method for signature 'INSPEcT_model'
modelSelection(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 <ul style="list-style-type: none"> • 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 thresholds for the goodness of fit (chisquare) and variability (brown) tests. Both must be comprised from 0 to 1. The goodness of fit threshold defines which models are considered valid or not (0 most stringent, 1 most permissive). The variability threshold (one per each rate) defines the stringency of the call for the variability of each rate (0 the most stringent, 1 the most permissive). When set to 0, the specific rate is excluded from the hypothesis of variability.
- **preferPValue**: when model selection is "llr", preferPValue means that if the selected model has a goodness of fit below threshold the model with the best goodness of fit is returned in place of it. When the model selection is "aic", with preferPValue the best model is tested against the closest nested models to test the hypothesis of variability and only when this pvalue is below threshold the rate is considered as variable. Otherwise the model selection is just based on the lowest AIC (Default: TRUE).
- **padj**: whether to correct pvalues with the Benjamini-Hochberg procedure or not. (Default: TRUE).
- **limitModelComplexity**: limit the complexity of the models associated to variable rates to the length of the time courses. Particularly helpful for short time series. (Default:FALSE)

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". 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](#)).

Value

See "value"

See Also

[makeSimModel](#), [makeSimDataset](#)

Examples

```
nascentInspObj10 <- readRDS(system.file(package='INSPECT', 'nascentInspObj10.rds'))
modelSelection(nascentInspObj10)
modelSelection(nascentInspObj10)$modelSelection <- 'aic'
```

newINSPEcT *Create a new INSPEcT object*

Description

The function `newINSPEcT` creates a new instance of the class `INSPEcT` provided the experimental time points, expression data (like RPKMs) of mature and eventually nascent RNA. For the nascent analysis, it is also requires a collecting time and the scaling factor to normalize the nascent RNA-seq libraries. This latter parameter can also be calculated by the function itself if both exonic and intronic expression data are provided; otherwise it must be given as an input and it is essential to guarantee the robustness of the analysis.

Usage

```
newINSPEcT(tpts, labeling_time = NULL, nascentExpressions = NULL,
  matureExpressions, preexisting = FALSE, BPPARAM = bpparam(),
  labeledSF = NULL, simulatedData = FALSE, degDuringPulse = FALSE,
  Dmin = 1e-06, Dmax = 10, genesFilter = TRUE)
```

Arguments

<code>tpts</code>	A vector of time points, one for each sample
<code>labeling_time</code>	A number, length of the Nascent pulse
<code>nascentExpressions</code>	A list which contains exons and introns expression matrices and variances for the nascent RNA
<code>matureExpressions</code>	A list which contains exons and introns expression matrices and variances for the mature RNA
<code>preexisting</code>	A logical, indicating if the mature expression refers to the pre-existing (unlabeled) population. Not implemented yet for the "degDuringPulse" mode.
<code>BPPARAM</code>	Configuration for BiocParallel parallelization. By default is set to <code>bpparam()</code>
<code>labeledSF</code>	A vector storing user defined normalization scale over Nascent RNA exons and introns quantifications
<code>simulatedData</code>	A logical, set to <code>TRUE</code> in case the analysis is on simulated data
<code>degDuringPulse</code>	A logical, set to <code>TRUE</code> in case of a long labelling time. Also degradation of newly synthesized transcripts will be taken into account
<code>Dmin</code>	A numerical, it is the lower bound of the degradation rate domain for the prior optimization
<code>Dmax</code>	A numerical, it is the upper bound of the degradation rate domain for the prior optimization
<code>genesFilter</code>	A logical, if <code>TRUE</code> , filters out genes which have no signal in at least 2/3 of the observations

Value

An object of class `INSPEcT` with a first estimation of the rates which can be accessed by the method [ratesFirstGuess](#)

Examples

```

data('allcounts', package='INSPEcT')
data('featureWidths', package='INSPEcT')
data('libsizes', package='INSPEcT')

matureCounts<-allcounts$mature
tpts <- c(0,1/6,1/3,1/2,1,1.5,2,4,8,12,16)
expDes<-rep(tpts,3)

matExp_DESeq2<-quantifyExpressionsFromTrCounts(
  allcounts=matureCounts
  ,libsize=totalLS
  ,exonsWidths=exWdths
  ,intronsWidths=intWdths
  ,experimentalDesign=expDes)

matureInspObj<-newINSPEcT(tpts=tpts
  ,labeling_time=NULL
  ,nascentExpressions=NULL
  ,matureExpressions=matExp_DESeq2)

```

nGenes

Get the number of genes within the INSPEcT object

Description

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

Usage

```

nGenes(object)

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

```

Arguments

object An object of class INSPEcT

Value

A numeric that indicates the number of genes within the object

Examples

```

nascentInspObj10 <- readRDS(system.file(package='INSPEcT', 'nascentInspObj10.rds'))
nGenes(nascentInspObj10)

```

nTpts	<i>Get the number of time points within the INSPEcT object</i>
-------	----------------------------------------------------------------

Description

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

Usage

```
nTpts(object)

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

Arguments

object An object of class INSPEcT

Value

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

Examples

```
nascentInspObj10 <- readRDS(system.file(package='INSPEcT', 'nascentInspObj10.rds'))
nTpts(nascentInspObj10)
```

plotGene	<i>Plot the pre-modeled and modeled profiles for one gene</i>
----------	---------------------------------------------------------------

Description

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

Usage

```
plotGene(object, ix, fix.yaxis = FALSE, priors = TRUE)

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

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
priors A logical, if true the priors of the rates are plotted

Value

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

Examples

```
nascentInspObj10 <- readRDS(system.file(package='INSPECT', 'nascentInspObj10.rds'))
plotGene(nascentInspObj10, 1)
```

plotMA

MA-plot from base means and log fold changes

Description

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

Usage

```
## 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.
- "padj" - A numeric, The p-adjusted threshold for significance. Genes with p-adjusted lower than the threshold will be depicted as orange triangles. By default set to -Inf, meaning that no genes will be highlighted.
- "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

http://en.wikipedia.org/wiki/MA_plot

Examples

```

if( Sys.info()["sysname"] != "Windows" ) {
  data('allcounts', package='INSPEcT')
  data('featureWidths', package='INSPEcT')
  data('libsizes', package='INSPEcT')

  nascentCounts<-allcounts$nascent
  matureCounts<-allcounts$mature
  conditions<-letters[1:11]
  expDes<-rep(conditions,3)
  tL<-1/6

  nasExp_DESeq2<-quantifyExpressionsFromTrCounts(
    allcounts=nascentCounts
    ,libsize=nascentLS
    ,exonsWidths=exWdths
    ,intronsWidths=intWdths
    ,experimentalDesign=expDes)

  matExp_DESeq2<-quantifyExpressionsFromTrCounts(
    allcounts=matureCounts
    ,libsize=totalLS
    ,exonsWidths=exWdths
    ,intronsWidths=intWdths
    ,experimentalDesign=expDes)

  nasFullObj <- newINSPEcT(tpts=conditions
    ,labeling_time=tL
    ,nascentExpressions=nasExp_DESeq2
    ,matureExpressions=matExp_DESeq2)

  diffrates = compareSteady(nasFullObj[,c(1,11)])

  plotMA(diffrates, padj=.01)
}

```

processingDelay

Classify genes as delayed by the processing using the delta and tau metrics

Description

These functions calculates the tau and delta metrics for all genes with introns and exons in an object of class INSPEcT. If the INSPEcT dataset was obtained with nascent RNA the metrics are calculated using RNA dynamics and solving numerically the system of equations. If the INSPEcT dataset was obtained without nascent RNA the metrics are approximated using premature and mature levels.

Usage

```
processingDelay(inspectIds, tauThreshold = 1.2, deltaThreshold = 1,
  silent = TRUE)
```

```
calculateDelta(inspectIds, silent = FALSE)
```

```

calculateTau(inspectIds, silent = FALSE)

## S4 method for signature 'INSPEcT'
processingDelay(inspectIds, tauThreshold = 1.2,
  deltaThreshold = 1, silent = TRUE)

## S4 method for signature 'INSPEcT'
calculateTau(inspectIds, silent = FALSE)

## S4 method for signature 'INSPEcT'
calculateDelta(inspectIds, silent = FALSE)

```

Arguments

<code>inspectIds</code>	An object of class <code>INSPEcT</code> .
<code>tauThreshold</code>	A numeric representing the tau threshold to define a gene affected by processing. Default: 1.2
<code>deltaThreshold</code>	A numeric representing the delta threshold to define a gene affected by processing. Default: 1.0
<code>silent</code>	A logical indicating whether information about the procedure should be printed or not.

Examples

```

data('allcounts', package='INSPEcT')
data('featureWidths', package='INSPEcT')
data('libsizes', package='INSPEcT')

nascentCounts<-allcounts$nascent
matureCounts<-allcounts$mature
conditions<-c(0,1/6,1/3,1/2,1,1.5,2,4,8,12,16)
expDes<-rep(conditions,3)
tL <- 1/6

nasExp_DESeq2<-quantifyExpressionsFromTrCounts(
  allcounts=matureCounts
  ,libsize=totalLS
  ,exonsWidths=exWdths
  ,intronsWidths=intWdths
  ,experimentalDesign=expDes)

matExp_DESeq2<-quantifyExpressionsFromTrCounts(
  allcounts=matureCounts
  ,libsize=totalLS
  ,exonsWidths=exWdths
  ,intronsWidths=intWdths
  ,experimentalDesign=expDes)

matureInspObj <- newINSPEcT(
  tpts=conditions
  ,labeling_time=tL
  ,nascentExpressions=nasExp_DESeq2
  ,matureExpressions=matExp_DESeq2)

```



```

procDelay<- processingDelay(inspectIds=matureInspObj
,tauThreshold=1.2
,deltaThreshold=1.0)

head(procDelay)
table(procDelay)

head(calculateTau(matureInspObj))

head(calculateDelta(matureInspObj))

```

quantifyExpressionsFromBAMs

Evaluate introns and exons expressions from BAM or SAM files

Description

Given a TranscriptDb object and a list of BAM or SAM files "quantifyExpressionsFormBAMs" evaluates exons and introns expressions and the associated variances per each gene.

Usage

```

quantifyExpressionsFromBAMs(txdb, BAMfiles, experimentalDesign,
by = c("gene", "tx"), countMultiMappingReads = FALSE,
allowMultiOverlap = FALSE, prioritizeExons = TRUE,
libsize = c("assigned", "all"), strandSpecific = 0,
isPairedEnd = FALSE, DESeq2 = TRUE, varSamplingCondition = NULL)

```

Arguments

txdb	A TranscriptDB object
BAMfiles	A vector of paths
experimentalDesign	A numerical which reports the desing of the experiment in terms of time points and replicates. Time points must be ordered according to the sequence of files submitted for the analysis, these labels characterize different files as replicates of a given condition.
by	A character, either "gene" or "tx", indicating if expressions and counts should be summarized at the levels of genes or transcripts. "gene" by default. In case "tx" is selected, we suggest to set argument "allowMultiOverlap" to TRUE, otherwise the reads mapping to overlapping transcripts of the same gene will remain unassigned.
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

prioritizeExons	A logical, indicating whether reads assigned to exon should not be accounted for intron counts. If set to FALSE, reads with shared overlap between an exon and the following intron will be assigned to the intron. This could improve intronic quantification in experimental settings (including polyA library preparation) or compact genomes where intronic reads are sampled at a very low rate compared to exonic reads. By default, TRUE.
libsize	A character, either "assigned" or "all", indicating whether the libsize for expression normalization should include all mapped reads or only the reads assigned to any of the features. By default, "assigned" is selected.
strandSpecific	Numeric, 0 if no strand-specific read counting should be performed, 1 stranded, 2 reversely-stranded. 0 by default
isPairedEnd	A logical, if paired-end reads are used, FALSE by default
DESeq2	A logical, if TRUE exons and introns variances are evaluated through the package DESeq2, if FALSE through plgem
varSamplingCondition	A character reporting which experimental condition should be used to sample the variance if DESeq2 = FALSE. By default, the first element of "experimentalDesign" with replicates.

Value

A list containing expressions and associated variances for exons and introns.

Examples

```
if( Sys.info()["sysname"] != "Windows" ) {
  require(TxDb.Mmusculus.UCSC.mm9.knownGene)
  txdb<-TxDb.Mmusculus.UCSC.mm9.knownGene
  expDes<-c(0,0,1,1)

  paths_total<-system.file('extdata/', c('bamRep1.bam'
                                         , 'bamRep2.bam'
                                         , 'bamRep3.bam'
                                         , 'bamRep4.bam')
                           ,package='INSPECT')

  matExp<-quantifyExpressionsFromBAMs(txdb=txdb
                                     ,BAMfiles=paths_total
                                     ,experimentalDesign=expDes)
}
```

quantifyExpressionsFromTrAbundance

Given introns and exons abundances (for example RPKMs) this method returns their variances evaluated through plgem.

Description

Given introns and exons abundances (for example RPKMs) this method returns their variances evaluated through plgem.

Usage

```
quantifyExpressionsFromTrAbundance(trAbundances, experimentalDesign,
  varSamplingCondition = NULL, simulatedData = FALSE)
```

Arguments

`trAbundances` A a list with elements "exonsAbundances" and "intronsAbundances".

`experimentalDesign` A numerical which reports the desing of the experiment in terms of time points and replicates. The time points must be ordered according to the columns of the count matrices submitted for the analysis; these labels define conditions and replicates.

`varSamplingCondition` A character reporting which experimental condition should be used to sample the variance if DESeq2 = FALSE.

`simulatedData` A boolean which is TRUE if the data under analysis are simulated.

Value

A list containing RPKMs and associated variances for exons and introns.

quantifyExpressionsFromTrCounts

Evaluates introns and exons RPKMs, per gene, from counts data.

Description

Evaluates introns and exons RPKMs, per gene, from counts data.

Usage

```
quantifyExpressionsFromTrCounts(allcounts, experimentalDesign, exonsWidths,
  intronsWidths, libsize = NULL, DESeq2 = TRUE,
  varSamplingCondition = NULL)
```

Arguments

`allcounts` A named list containing "exonsCounts" and "intronsCounts".

`experimentalDesign` A numerical which reports the desing of the experiment in terms of time points and replicates. Time points must be ordered according to the sequence of files submitted for the analysis, these labels characterize different files as replicates of a given condition.

`exonsWidths` A numeric containing the exons widths.

`intronsWidths` A numeric containing the introns widths.

`libsize` A numeric containing the library size.

`DESeq2` A logical, if TRUE the RPKMs variances are evaluated through the package DESeq2, if FALSE plgem is used.

`varSamplingCondition` A character reporting which experimental condition should be used to sample the variance if DESeq2 = FALSE. By default, the first element of "experimentalDesign" with replicates.

Value

A list containing RPKMs and associated variances for exons and introns.

Examples

```
data('allcounts', package='INSPECT')
data('featureWidths', package='INSPECT')
data('libsizes', package='INSPECT')

nascentCounts<-allcounts$nascent
matureCounts<-allcounts$mature

expDes<-rep(c(0,1/6,1/3,1/2,1,1.5,2,4,8,12,16),3)

nasExp_DESeq2<-quantifyExpressionsFromTrCounts(libsize=nascentLS
,exonsWidths=exWdths
,intronsWidths=intWdths
,allcounts=nascentCounts
,experimentalDesign=expDes)

matExp_DESeq2<-quantifyExpressionsFromTrCounts(libsize=totalLS
,exonsWidths=exWdths
,intronsWidths=intWdths
,allcounts=matureCounts
,experimentalDesign=expDes)

nasExp_plgem<-quantifyExpressionsFromTrCounts(libsize=nascentLS
,exonsWidths=exWdths
,intronsWidths=intWdths
,allcounts=nascentCounts
,DESeq2=FALSE
,experimentalDesign=expDes)

matExp_plgem<-quantifyExpressionsFromTrCounts(libsize=totalLS
,exonsWidths=exWdths
,intronsWidths=intWdths
,allcounts=matureCounts
,DESeq2=FALSE
,experimentalDesign=expDes)
```

ratePvals

Retrieve a single p-value for each rate

Description

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 sythetic 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, bTsh = NULL, cTsh = NULL)

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

## S4 method for signature 'INSPECT'
ratePvals(object, bTsh = NULL, cTsh = NULL)
```

Arguments

object	An object of class <code>INSPECT</code> or <code>INSPECT_model</code>
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

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

```
nascentInspObj10 <- readRDS(system.file(package='INSPECT', 'nascentInspObj10.rds'))
ratePvals(nascentInspObj10)
# calculate again the p-values with Brown with a different threshold
# for considering a model valid for the log likelihood ratio test
ratePvals(nascentInspObj10, cTsh=.2)
# Set permanently the chi-squared threshold at .2 for nascentInspObj10 object
modelSelection(nascentInspObj10)$thresholds$chisquare <- .2
```

ratesFirstGuess

Retrieve pre-modeling rates and concentrations

Description

This method allow 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

```
nascentInspObj10 <- readRDS(system.file(package='INSPEcT', 'nascentInspObj10.rds'))

ratesFirstGuess(nascentInspObj10, 'total')
ratesFirstGuess(nascentInspObj10, 'preMRNA')
ratesFirstGuess(nascentInspObj10, 'synthesis')
```

ratesFirstGuessVar	<i>Retrieve pre-modeling rates and concentrations variance</i>
--------------------	----------------------------------------------------------------

Description

This method allow 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

```
nascentInspObj10 <- readRDS(system.file(package='INSPECT', 'nascentInspObj10.rds'))

ratesFirstGuessVar(nascentInspObj10, 'total')
ratesFirstGuessVar(nascentInspObj10, 'preMRNA')
ratesFirstGuessVar(nascentInspObj10, 'synthesis')
```

removeModel

remove modelling information from INSPECT object

Description

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.

Usage

```
removeModel(object)

## S4 method for signature 'INSPECT'
removeModel(object)
```

Arguments

object An Object of class INSPECT

Value

An Object of class INSPECT

Examples

```
nascentInspObj10 <- readRDS(system.file(package='INSPECT', 'nascentInspObj10.rds'))
nascentInspObj10_5genes <- nascentInspObj10[1:5]

## This will turn out into an error:
## Not run: nascentInspObj10_5genes_5tpts <- nascentInspObj10[1:5, 1:5]

## Before subsetting time points, the model should be removed:
nascentInspObj10_5genes_5tpts <- removeModel(nascentInspObj10)[1:5, 1:5]

## Also this will turn out into an error:
## Not run: nascentInspObj10 <- modelRates(nascentInspObj10)
```

```
## Before running the model again, or changing modeling parameters,
## the previous model should be removed:
nascentInspObj10_old <- nascentInspObj10
nascentInspObj10_new <- removeModel(nascentInspObj10)
modelingParams(nascentInspObj10_new)$useSigmoidFun <- FALSE
## Not run: nascentInspObj10_new <- modelRates(nascentInspObj10_new)
```

rocCurve

*Display rate classification performance***Description**

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.

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

```

if( Sys.info()["sysname"] != "Windows" ) {
  nascentInspObj <- readRDS(system.file(package='INSPEcT', 'nascentInspObj.rds'))

  simRates<-makeSimModel(nascentInspObj, 1000, seed=1)

  # newTpts<-simRates@params$tpts
  # nascentSim2replicates<-makeSimDataset(object=simRates
  #                                     ,tpts=newTpts
  #                                     ,nRep=3
  #                                     ,NoNascent=FALSE
  #                                     ,seed=1)
  # nascentSim2replicates<-modelRates(nascentSim2replicates[1:100]
  #                                   ,seed=1)
  # (not evaluated to save computational time)

  data("nascentSim2replicates",package='INSPEcT')

  rocCurve(simRates[1:100],nascentSim2replicates)
  title("3rep. 11t.p. Total and nascent RNA", line=3)
}

```

rocThresholds	<i>Display rate classification performance with thresholds visible at x-axis</i>
---------------	----------------------------------------------------------------------------------

Description

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

```

if( Sys.info()["sysname"] != "Windows" ) {
  nascentInspObj <- readRDS(system.file(package='INSPEcT', 'nascentInspObj.rds'))

  simRates<-makeSimModel(nascentInspObj, 1000, seed=1)

  # newTpts<-simRates@params$tpts
  # nascentSim2replicates<-makeSimDataset(object=simRates
  #                                     ,tpts=newTpts
  #                                     ,nRep=3
  #                                     ,NoNascent=FALSE
  #                                     ,seed=1)
  # nascentSim2replicates<-modelRates(nascentSim2replicates[1:100]
  #                                   ,seed=1)
  # (not evaluated to save computational time)

  data("nascentSim2replicates",package='INSPEcT')

  rocThresholds(simRates[1:100],nascentSim2replicates,bTsh=c(.01, .01, .05),cTsh=.1)
}

```

sfPlot

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

Description

This method generates a plot that immediately shows the scaling factors used to scale RNA- and Nascent-seq libraries and the possible relations between them. The ratio between the RNA- and the Nascent-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 nascentInspObj10 <- readRDS(system.file(package='INSPEcT', 'nascentInspObj10.rds')) sf-
Plot(nascentInspObj10)
```

simData3rep_Nascent *An INSPEcT object with 1000 simulated rates and concentration and their modeled rates*

Description

A dataset containing the rates and concentrations obtained from the dataset simRates; 3 replicates and time points corresponding to: 0, 1/6, 1/3, 1/2, 1, 1.5, 2, 4, 8, 12, 16 hours.

Format

An INSPEcT object

simData3rep_NoNascent *An INSPEcT object with 1000 simulated rates and concentration and their modeled rates*

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

simData4rep_Nascent	<i>An INSPEcT object with 1000 simulated rates and concentration and their modeled rates</i>
---------------------	----------------------------------------------------------------------------------------------

Description

A dataset containing the rates and concentrations obtained from the dataset simRates; 3 replicates and time points corresponding to: 0,1/6,1/3,1/2,1,1.25,1.5,2,3,4,6,8,10,12,16 hours.

Format

An INSPEcT object

simData4rep_NoNascent	<i>An INSPEcT object with 1000 simulated rates and concentration and their modeled rates</i>
-----------------------	----------------------------------------------------------------------------------------------

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

split	<i>Divide an INSPEcT Object into groups</i>
-------	---------------------------------------------

Description

Divides the INSPEcT object into the groups defined by 'f',

Usage

```
## 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

```
nascentInspObj10 <- readRDS(system.file(package='INSPEcT', 'nascentInspObj10.rds'))
splitIdx <- c(1,1,1,2,2,2,3,3,3,4)
nascentInspObj10Split <- split(nascentInspObj10, splitIdx)
```

tpts

Accessor to the slot tpts of an INSPEcT object

Description

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

```
nascentInspObj10 <- readRDS(system.file(package='INSPEcT', 'nascentInspObj10.rds'))
tpts(nascentInspObj10)
```

viewModelRates

Retrieve the modeled rates and concentrations

Description

A method to access the modeld 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

```
nascentInspObj10 <- readRDS(system.file(package='INSPEcT', 'nascentInspObj10.rds'))  
viewModelRates(nascentInspObj10, 'synthesis')
```

Index

- [, [14](#), [18](#)
- [, INSPECT, ANY, ANY, ANY-method (Extract), [8](#)
- [, INSPECT_diffsteady, ANY, ANY, ANY-method (Extract), [8](#)
- [, INSPECT_model, ANY, ANY, ANY-method (Extract), [8](#)

- AIC, [14](#), [18](#)
- AIC, INSPECT-method (AIC-INSPECT-method), [3](#)
- AIC, INSPECT_model-method (AIC-INSPECT-method), [3](#)
- AIC-INSPECT-method, [3](#)
- allcounts, [3](#)

- calculateDelta (processingDelay), [31](#)
- calculateDelta, INSPECT-method (processingDelay), [31](#)
- calculateTau (processingDelay), [31](#)
- calculateTau, INSPECT-method (processingDelay), [31](#)
- chisqmodel, [4](#), [14](#), [18](#)
- chisqmodel, INSPECT-method (chisqmodel), [4](#)
- chisqmodel, INSPECT_model-method (chisqmodel), [4](#)
- chisqtest, [4](#), [14](#), [18](#)
- chisqtest, INSPECT-method (chisqtest), [4](#)
- chisqtest, INSPECT_model-method (chisqtest), [4](#)
- combine, [5](#), [14](#)
- combine, INSPECT, INSPECT-method (combine), [5](#)
- compareSteady, [6](#), [15](#)
- compareSteady, INSPECT-method (compareSteady), [6](#)
- compareSteadyNoNascent, [7](#)
- compareSteadyNoNascent, INSPECT-method (compareSteadyNoNascent), [7](#)

- degradation (INSPECT_diffsteady-class), [15](#)

- degradation, INSPECT_diffsteady-method (INSPECT_diffsteady-class), [15](#)
- dim, [14](#)
- dim, INSPECT-method, [8](#)
- Extract, [8](#)

- featureNames, [14](#)
- featureNames, INSPECT-method, [9](#)
- featureNames, INSPECT_diffsteady-method (INSPECT_diffsteady-class), [15](#)
- featureNames<-, INSPECT-method (featureNames, INSPECT-method), [9](#)
- featureWidths, [10](#)

- geneClass, [10](#), [14](#), [18](#), [24](#), [25](#)
- geneClass, INSPECT-method (geneClass), [10](#)
- geneClass, INSPECT_diffsteady-method (geneClass), [10](#)
- geneClass, INSPECT_model-method (geneClass), [10](#)
- getModel, [11](#), [14](#)
- getModel, INSPECT-method (getModel), [11](#)
- getModel<- (getModel), [11](#)
- getModel<-, INSPECT-method (getModel), [11](#)

- inHeatmap, [12](#), [14](#)
- inHeatmap, INSPECT-method (inHeatmap), [12](#)
- INSPECT, [13](#), [13](#)
- INSPECT-class, [13](#)
- INSPECT-package (INSPECT), [13](#)
- INSPECT-shinyApps, [15](#)
- INSPECT_diffsteady-class, [15](#)
- INSPECT_model, [13](#)
- INSPECT_model-class, [17](#)

- labeledSF, [14](#), [18](#)
- labeledSF, INSPECT-method (labeledSF), [18](#)
- libsizes, [19](#)
- logLik, [14](#), [18](#), [19](#)
- logLik, INSPECT-method (logLik), [19](#)
- logLik, INSPECT_model-method (logLik), [19](#)

- makeModelRates, [14](#), [18](#), [20](#)

- makeModelRates, INSPEcT-method
(makeModelRates), 20
- makeModelRates, INSPEcT_model-method
(makeModelRates), 20
- makeSimDataset, 17, 18, 21, 22, 26, 36, 37,
40, 42
- makeSimDataset, INSPEcT_model-method
(makeSimDataset), 21
- makeSimModel, 14, 17, 21, 22, 22, 26, 36, 37,
40, 42
- makeSimModel, INSPEcT-method
(makeSimModel), 22
- modelingParams, 13, 14, 23, 24, 25
- modelingParams, INSPEcT-method
(modelingParams), 23
- modelingParams<- (modelingParams), 23
- modelingParams<-, INSPEcT-method
(modelingParams), 23
- modelRates, 14, 17, 20, 23, 24, 40, 41, 45
- modelRates, INSPEcT-method (modelRates),
24
- modelSelection, 14, 18, 25, 36
- modelSelection, INSPEcT-method
(modelSelection), 25
- modelSelection, INSPEcT_model-method
(modelSelection), 25
- modelSelection<- (modelSelection), 25
- modelSelection<-, INSPEcT-method
(modelSelection), 25
- modelSelection<-, INSPEcT_model-method
(modelSelection), 25

- newINSPEcT, 13, 24, 27, 37–39
- nGenes, 8, 14, 28
- nGenes, INSPEcT-method (nGenes), 28
- nTpts, 8, 14, 29
- nTpts, INSPEcT-method (nTpts), 29

- plotGene, 14, 29
- plotGene, INSPEcT-method (plotGene), 29
- plotMA, 16, 30
- plotMA, INSPEcT_diffsteady-method
(plotMA), 30
- processing (INSPEcT_diffsteady-class),
15
- processing, INSPEcT_diffsteady-method
(INSPEcT_diffsteady-class), 15
- processingDelay, 31
- processingDelay, INSPEcT-method
(processingDelay), 31

- quantifyExpressionsFromBAMs, 33
- quantifyExpressionsFromTrAbundance, 34

- quantifyExpressionsFromTrCounts, 35

- ratePvals, 11, 14, 18, 36, 37
- ratePvals, INSPEcT-method (ratePvals), 36
- ratePvals, INSPEcT_model-method
(ratePvals), 36
- ratesFirstGuess, 14, 27, 37, 39
- ratesFirstGuess, INSPEcT-method
(ratesFirstGuess), 37
- ratesFirstGuessVar, 14, 38, 38
- ratesFirstGuessVar, INSPEcT-method
(ratesFirstGuessVar), 38
- removeModel, 14, 39
- removeModel, INSPEcT-method
(removeModel), 39
- rocCurve, 18, 40, 42
- rocCurve, INSPEcT_model, INSPEcT-method
(rocCurve), 40
- rocCurve, INSPEcT_model, INSPEcT_model-method
(rocCurve), 40
- rocThresholds, 18, 40, 41
- rocThresholds, INSPEcT_model, INSPEcT-method
(rocThresholds), 41
- rocThresholds, INSPEcT_model, INSPEcT_model-method
(rocThresholds), 41
- runProcessingRateDelay
(INSPEcT-shinyApps), 15
- runRNAdynamics (INSPEcT-shinyApps), 15

- sfPlot, 14, 42
- sfPlot, INSPEcT-method (sfPlot), 42
- show, INSPEcT-method (INSPEcT-class), 13
- show, INSPEcT_diffsteady-method
(INSPEcT_diffsteady-class), 15
- show, INSPEcT_model-method
(INSPEcT_model-class), 17
- simData3rep_Nascent, 43
- simData3rep_NoNascent, 43
- simData4rep_Nascent, 44
- simData4rep_NoNascent, 44
- split, 44
- split, INSPEcT, ANY-method (split), 44
- synthesis (INSPEcT_diffsteady-class), 15
- synthesis, INSPEcT_diffsteady-method
(INSPEcT_diffsteady-class), 15

- tpts, 14, 45
- tpts, INSPEcT-method (tpts), 45

- viewModelRates, 14, 24, 25, 45
- viewModelRates, INSPEcT-method
(viewModelRates), 45