

# Package ‘INSPEcT’

April 15, 2020

**Type** Package

**Title** Modeling RNA synthesis, processing and degradation with RNA-seq data

**Version** 1.16.3

**Date** 2019-11-25

**Author** Stefano de Pretis

**Maintainer** Stefano de Pretis <ste.depo@gmail.com>,  
Mattia Furlan <Mattia.Furlan@iit.it>

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

**License** GPL-2

**Depends** R (>= 3.5), methods, Biobase, BiocParallel

**Imports** pROC, deSolve, rootSolve, KernSmooth, gdata, GenomicFeatures, GenomicRanges, IRanges, BiocGenerics, GenomicAlignments, Rsamtools, S4Vectors, GenomeInfoDb, DESeq2, plgem, SummarizedExperiment, TxDb.Mmusculus.UCSC.mm9.knownGene, shiny

**Suggests** BiocStyle, knitr, rmarkdown

**VignetteBuilder** knitr

**biocViews** Sequencing, RNASeq, GeneRegulation, TimeCourse, SystemsBiology

**NeedsCompilation** no

**RoxygenNote** 7.0.2

**git\_url** <https://git.bioconductor.org/packages/INSPEcT>

**git\_branch** RELEASE\_3\_10

**git\_last\_commit** 57a0ad7

**git\_last\_commit\_date** 2020-02-03

**Date/Publication** 2020-04-14

**R topics documented:**

AIC-INSPEcT-method	3
allcounts	4
calculateRatePvals	4
chisqmodel	5
chisqtest	6
combine	7
compareSteady	8
compareSteadyNoNascent	9
computeConfidenceIntervals	10
convergence	11
correlationPlot	11
dim,INSPEcT-method	12
Extract	12
featureNames,INSPEcT-method	13
featureWidths	14
geneClass	14
inHeatmap	15
INSPEcT	16
INSPEcT-class	17
INSPEcT-shinyApps	19
inspectFromBAM	19
inspectFromPCR	20
INSPEcT_diffsteady-class	21
INSPEcT_model-class	23
labeledSF	24
libsizes	25
logLik	25
makeModelRates	26
makeOscillatorySimModel	27
makeSimDataset	28
makeSimModel	29
modelingParams	30
modelRates	31
modelRatesNF	33
modelSelection	34
newINSPEcT	35
nGenes	36
nTpts	37
plotGene	37
plotMA	38
processingDelay	39
quantifyExpressionsFromBAMs	41
quantifyExpressionsFromTrAbundance	43
quantifyExpressionsFromTrCounts	44
ratePvals	45
ratesFirstGuess	46
ratesFirstGuess Var	46
removeModel	47
rocCurve	48
rocThresholds	49

setConfidenceIntervals . . . . .	50
simData3rep_Nascent . . . . .	51
simData3rep_NoNascent . . . . .	51
simData4rep_Nascent . . . . .	51
simData4rep_NoNascent . . . . .	52
split . . . . .	52
tpts . . . . .	53
viewConfidenceIntervals . . . . .	53
viewModelRates . . . . .	54

<b>Index</b>	<b>55</b>
--------------	-----------

---

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

---

calculateRatePvals	<i>Calculate a single p-value for each rate</i>
--------------------	---

---

### Description

This method is used to calculate all the p-values relative to the variability of synthesis, processing and degradation rates. For object modeled with nascent RNA or when non-functional modeling was used, the variability is calculated using the confidence intervals. For objects modeled without nascent RNA, model selection is performed by comparing the likelihood of different (nested) models.

### Usage

```
calculateRatePvals(
  object,
  modelSelection = c("aic", "llr", "hib"),
  preferPValue = TRUE,
  padj = TRUE,
  p_goodness_of_fit = 0.1,
  p_variability = rep(0.05, 3),
  limitModelComplexity = FALSE
)

## S4 method for signature 'INSPECT'
calculateRatePvals(
  object,
  modelSelection = c("aic", "llr", "hib"),
  preferPValue = TRUE,
  padj = TRUE,
  p_goodness_of_fit = 0.1,
  p_variability = rep(0.05, 3),
  limitModelComplexity = FALSE
)
```

**Arguments**

<code>object</code>	An object of class <code>INSPEcT</code> or <code>INSPEcT_model</code>
<code>modelSelection</code>	'aic' compares nested models closest to the one with lowest AIC, 'lir' compares all nested models, 'hib' is a mix between the previous two. (default 'aic')
<code>preferPValue</code>	a logical, if TRUE (default) limit the search for best models among the ones with succeeded the goodness of fit test.
<code>padj</code>	a logical, if TRUE (default) correct the p-values for multiple testing
<code>p_goodness_of_fit</code>	a numeric, the threshold for the goodness-of-fit test (default = .1)
<code>p_variability</code>	a numeric, a vector with the thresholds for the p-value of the variability test (one threshold for each rate, default = rep(.05, 3))
<code>limitModelComplexity</code>	a logical that limits the complexity of the function used to describe dynamics to the length of the time-course (default = FALSE)

**Details**

`ratePvals` retrieve a single p-value for each rate and gene associated to its variability (null hypothesis = the rate is not changing between the conditions)

**Value**

A matrix containing p-values calculated for each rate

**See Also**

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

**Examples**

```
nascentInspObj10 <- readRDS(system.file(package='INSPEcT', 'nascentInspObj10.rds'))
# Set the chi-squared threshold at .2 for nascentInspObj10 object
nascentInspObj10 <- calculateRatePvals(nascentInspObj10, p_goodness_of_fit=.2)
```

---

`chisqmodel`

*Retrieve results of chi-squared test for the selected models*

---

**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, gc = NULL, tpts = NULL, ...)

## S4 method for signature 'INSPEcT'
chisqmodel(object, gc = NULL, tpts = NULL, ...)
```

**Arguments**

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

<code>x</code>	An object of class INSPEcT
<code>y</code>	An object of class INSPEcT
<code>...</code>	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,
  trivialAngle = NULL,
  returnNormScores = FALSE,
  referenceCondition = NULL
)

## S4 method for signature 'INSPECT'
compareSteadyNoNascent(
  inspectIds,
  expressionThreshold = 0.25,
  log2FCThreshold = 2,
  trivialAngle = NULL,
  returnNormScores = FALSE,
  referenceCondition = NULL
)

```

## Arguments

`inspectIds` An object of class `INSPECT`.

`expressionThreshold` A parameter which sets how many log<sub>2</sub> 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.

`trivialAngle` A numeric between 0 and 90 to define the standard behavior, if NULL (default) it is computed internally from the data.

`returnNormScores` A logical, if TRUE returned the deviations from the standard behavior normalized by the sd.

`referenceCondition` The label of the condition to use as reference, if NULL (default) the medians are used.

### 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=.5)
head(regGenes)
table(regGenes)
```

---

computeConfidenceIntervals

*Compute confidence intervals*

---

### Description

This function is used to compute the confidence intervals for a given set of modeled genes in the NoNascent scenario.

### Usage

```
computeConfidenceIntervals(object, BPPARAM = bpparam())

## S4 method for signature 'INSPECT'
computeConfidenceIntervals(object, BPPARAM = bpparam())
```

**Arguments**

object            An object of class INSPECT\_model  
 BPPARAM         Parallelization parameters for bplapply. By default bpparam()

**Value**

An object of class INSPECT.

---

convergence            *Retrieve the convergence for the selected models of each gene*

---

**Description**

This method is used to retrieve the convergence of the models that have been selected to better represent the behavior of each gene. 0 is converged, 1 not converged, 10 degenerated

**Usage**

```
convergence(object)

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

**Arguments**

object            An object of class INSPECT or INSPECT\_model

**Value**

A vector of numeric

**Examples**

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

---

correlationPlot            *Display rate classification performance*

---

**Description**

This function plot the rates of a simulated dataset against the modeled ones and compute their correlations.

**Usage**

```
correlationPlot(object, object2, plot = TRUE)

## S4 method for signature 'INSPECT_model,INSPECT'
correlationPlot(object, object2, plot = TRUE)
```

**Arguments**

object	An object of class <code>INSPEcT_model</code> with simulated rates.
object2	An object of class <code>INSPEcT</code> .
plot	A logical indicating whether to draw or not the plot. (default=TRUE)

**Value**

An list with the correlation values.

---

`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 <code>INSPEcT</code>
---	---

**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: "exWidths" and "intWidths" containing the length of the exons and introns, respectively, relative to the genes in "allcounts"</i>
---------------	--

---

**Description**

Contains two variables: "exWidths" and "intWidths" 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 variability of each rate is indicated with a letter, 's' for synthesis, 'p' for processing and 'd' for degradation. In case more than one rate is variable, the letters associated to each variable rate are merged, for example 'sd' stands for a gene where synthesis and degradation contributed to transcriptional changes. 'no-reg' is associated to genes with no change in transcription. The classification depends on the thresholds of the goodness-of-fit and rate variability tests that can be changed via the method [calculateRatePvals](#).

**Usage**

```
geneClass(object, ...)

## S4 method for signature 'INSPECT'
geneClass(object, ...)

## S4 method for signature 'INSPECT_model'
geneClass(object, ...)

## S4 method for signature 'INSPECT_diffsteady'
geneClass(object, ...)
```

**Arguments**

object            An object of class INSPECT or INSPECT\_model  
 ...               specify the threshold for rate variability 'bTsh' in case of 'INSPECT\_diffsteady'  
                   objects (default = .1)

**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 rate variability
nascentInspObj10 <- calculateRatePvals(nascentInspObj10, p_variability=rep(1,3))
geneClass(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,
  show_rowLabels = TRUE,
  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,
show_rowLabels = TRUE,
clustering = TRUE,
clustIdx = 3:5
)

```

### Arguments

<code>object</code>	An object of class <code>INSPEcT</code>
<code>type</code>	Either "pre-model" or "model" to switch between pre-modeled or modeled features
<code>breaks</code>	A vector of breaks for the heatmap
<code>palette</code>	A color generating function, output of <code>colorRampPalette</code>
<code>plot_matureRNA</code>	A logical. If set to <code>TRUE</code> , mature-RNA is displayed instead of total-RNA (default: <code>FALSE</code> )
<code>absoluteExpression</code>	A logical. If set to <code>FALSE</code> , the plot representing the intensity of expression is omitted. (default= <code>TRUE</code> )
<code>show_rowLabels</code>	A logical defining whether rownames are reported or not. (default= <code>TRUE</code> )
<code>clustering</code>	A logical. If set to <code>FALSE</code> , it displays genes the order they are, with no clustering (default: <code>TRUE</code> )
<code>clustIdx</code>	A numeric. Indicates which of the features are used for the clustering. 0= <code>absoluteExpression</code> ; 1= <code>total-RNA/mature-RNA</code> ; 2= <code>preMRNA</code> ; 3= <code>synthesis</code> ; 4= <code>degradation</code> ; 5= <code>processing</code> (default= <code>3:5</code> , 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 Transcriptomic 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.

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



INSPEcT implements two main classes ([INSPEcT](#) and [INSPEcT\\_diffsteady](#)) 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_diffsteady-class'
```

To obtain the 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](#)

```
[
calculateDelta
calculateRatePvals
calculateTau
chisqmodel
chisqtest
combine
compareSteady
compareSteadyNoNascent
computeConfidenceIntervals
correlationPlot
dim
featureNames
geneClass
inHeatmap
labeledSF
logLik
```

```

makeModelRates
makeOscillatorySimModel
makeSimModel
modelRates
modelRatesNF
modelSelection
modelingParams
nGenes
nTpts
plotGene
processingDelay
ratePvals
ratesFirstGuess
ratesFirstGuessVar
removeModel
rocCurve
rocThresholds
setConfidenceIntervals
show
split
tpts
viewConfidenceIntervals
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)

confidenceIntervals An object of class ExpressionSet that contains the confidence intervals.

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.

ratePvals A matrix containing the p-value relative to the variability of synthesis, processing and degradation for each gene.

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.

NF A logical indicating if the modeling approach is Non-Functional

degDuringPulse A logical indicating if degradation of RNA during the 4sU pulse was considered.  
 version A character indicating the version of INSPEcT that created the object

---

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. - runINSPEcTGUI: 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()

runINSPEcTGUI()
```

---

inspectFromBAM      *Wrapper function from BAM files*

---

### Description

Function to run the whole INSPEcT differential rate analysis procedure with a single line. The function save the output analysis to file that can be later loaded in the R environment or in the INSPEcT-GUI.

### Usage

```
inspectFromBAM(
  txdb,
  annotation_table,
  labeling_time = NULL,
  strandSpecific = 0,
  isPairedEnd = FALSE,
  estimateRatesWith = "der",
  useSigmoidFun = TRUE,
  file = NULL
)
```

### Arguments

txdb      A TranscriptDB object for the selected organism

annotation_table	Paths and experimental design associated to bam files. They could be provided directly as a 'data.frame', or as a path to the file containing the information. Possible file formats are csv (comma-separated-values), 'tsv' (comma-separated-values), or 'xls' (Excel). In case 'annotation_table' has 2 columns named 'condition' and 'total', INSPEcT- analysis is run. In case 'annotation_table' has 3 columns named 'condition', 'total' and 'nascent', INSPEcT+ analysis is run. 'condition' is a column indicating the experimental condition, a character vector (containing, for example, 'WT' or 'KD') in case of steady-state experiments, or numerical values indicating the time from the unperturbed condition in case of time-course analysis. 'total' and 'nascent' contains the path to totalRNA and nascentRNA BAM files, respectively.
labeling_time	A numeric indicating the time of labeling exposure to the modified nucleotide. To be indicated only in case of INSPEcT+ analysis.
strandSpecific	A numeric indicating the strandness of the BAM files, 0 for non strand-specific, 1 for stranded, 2 for reversely-stranded. 0 by default.
isPairedEnd	A logical indicating if paired-end sequencing have been performed. FALSE by default.
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 "der")
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)
file	A character indicating where the output of the analysis will be stored. If not provided the file name will be created automatically and saved on the current folder.

---

inspectFromPCR

*Wrapper function from PCR quantifications*


---

## Description

Function to run the whole INSPEcT differential rate analysis procedure with a single line. The function save the output analysis to file that can be later loaded in the R environment or in the INSPEcT-GUI.

## Usage

```
inspectFromPCR(
  totalRNA_table,
  nascentRNA_table = NULL,
  labeling_time = NULL,
  estimateRatesWith = "der",
  useSigmoidFun = TRUE,
  file = NULL
)
```

**Arguments**

- `totalRNA_table` Exonic quantification, intronic quantification and experimental design associated to totalRNA of a single gene quantified by PCR. They could be provided directly as a 'data.frame', or as a path to the file containing the information. Possible file formats are 'csv' (comma-separated-values), 'tsv' (comma-separated-values), or 'xls' (Excel). 'totalRNA\_table' must have 3 columns named 'condition', 'total\_exonic' and 'total\_intronic'. 'condition' is a column indicating the experimental condition, a character vector (containing, for example, 'WT' or 'KD') in case of steady-state experiments, or numerical values indicating the time from the unperturbed condition in case of time-course analysis. 'total\_exonic' and 'total\_intronic' contains abundance of gene measured in its exonic and intronic regions, respectively, in the total RNA fraction.
- `nascentRNA_table` similar to 'totalRNA\_table' but referred to nascent RNA fraction. In this case, columns names must be 'condition', 'nascent\_exonic' and 'nascent\_intronic'. In case this information is not provided, INSPEcT- analysis is run. If otherwise this information is present, INSPEcT+ analysis is run.
- `labeling_time` A numeric indicating the time of labeling exposure to the modified nucleotide. To be indicated only in case of INSPEcT+ analysis.
- `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 "der")
- `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)
- `file` A character indicating where the output of the analysis will be stored. If not provided the file name will be created automatically and saved on the current folder.

**Examples**

```
if( Sys.info()["sysname"] != "Windows" ) {
totalAnnTabPCR <- system.file(package = 'INSPEcT', 'totalAnnTabPCR.csv')
nascentAnnTabPCR <- system.file(package = 'INSPEcT', 'nascentAnnTabPCR.csv')
inspectFromPCR(totalAnnTabPCR, nascentAnnTabPCR, labeling_time=1/6)
}
```

---

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](#)  
[chisqtest](#)  
[correlationPlot](#)  
[geneClass](#)  
[logLik](#)  
[makeModelRates](#)  
[makeSimDataset](#)  
[modelSelection](#)  
[rocCurve](#)  
[rocThresholds](#)  
[show](#)

**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 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**

```
nascentInspObj10 <- readRDS(system.file(package='INSPEcT', 'nascentInspObj10.rds'))
viewModelRates(nascentInspObj10, 'degradation')
## force every degradation rate to be accepted as variable (makeModelRates is called internally)
nascentInspObj10 <- calculateRatePvals(nascentInspObj10, p_variability = c(.05,.05,1))
viewModelRates(nascentInspObj10, 'degradation')
```

---

`makeOscillatorySimModel`*Build the synthetic rates with oscillatory pattern*

---

## Description

This method allow the creation of synthesis, degradation and processing rates that generate an oscillatory expression with a period of 24 hours. Two modes are available: one where oscillations arise just by oscillations in the synthesis of the genes (`oscillatoryk3=FALSE`, default) and another one where both synthesis and degradation rates oscillates (`oscillatoryk3=TRUE`). In this latter case, the oscillations of the two rates can be coupled by a certain delay (parameter `k3delay`). After the creation of the synthetic rates, a dataset with noise and contamination added can be made by [makeSimDataset](#).

## Usage

```
makeOscillatorySimModel(  
  object,  
  nGenes,  
  oscillatoryk3 = FALSE,  
  k3delay = NULL,  
  na.rm = TRUE,  
  seed = NULL  
)  
  
## S4 method for signature 'INSPEcT'  
makeOscillatorySimModel(  
  object,  
  nGenes,  
  oscillatoryk3 = FALSE,  
  k3delay = NULL,  
  na.rm = TRUE,  
  seed = NULL  
)
```

## Arguments

<code>object</code>	An object of class <code>INSPEcT</code>
<code>nGenes</code>	A numeric with the number of synthtic genes to be created
<code>oscillatoryk3</code>	A logical that enables also degradation rate to oscillate
<code>k3delay</code>	A numeric that set the delay between synthesis and degradation oscillations. When <code>NULL</code> , no coupling between the two oscillations is set.
<code>na.rm</code>	A logical that set whether missing values in the real dataset should be removed
<code>seed</code>	A numeric to obtain reproducible results

## Value

An object of class `INSPEcT_model` with synthetic rates

**See Also**[makeSimModel](#)**Examples**

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

makeSimDataset

*Generate synthetic rates and concentrations***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,
  b = 0.3,
  tL = 1/6,
  noise_sd = 4
)

## S4 method for signature 'INSPEcT_model'
makeSimDataset(
  object,
  tpts,
  nRep,
  NoNascent = FALSE,
  seed = NULL,
  b = 0.3,
  tL = 1/6,
  noise_sd = 4
)
```

**Arguments**

object	An object of class INSPEcT_model, usually the output of <a href="#">makeSimModel</a>
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. (default=FALSE)
seed	A numeric to obtain reproducible results. When NULL (default) no seed is set.
b	A numeric which represents the probability of contamination of the unlabeled sample due to the labeled one
tL	A numeric which represents the labeling time for an ideal nascent RNA profiling, it is required for the contamination analysis. (default=1/6)
noise_sd	A numeric which represents the noise standard deviation. (default=4)

**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 = rbind(synthesis = c(constant = 0.5, sigmoid = 0.3, impulse = 0.2), processing
= c(constant = 0.5, sigmoid = 0.3, impulse = 0.2), degradation = 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 synthic genes to be created
newTpts	A numeric vector with time points of the synthic dataset, if NULL the time points of the real dataset will be used
probs	A numeric matrix wich describes the probability of each rate (rows) to be constant, shaped like a sigmoid or like an impulse model (columns)
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 associated 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

*Get and set number parameters for the modeling*

---

### Description

A method to get the parameters used for modeling rates and concentrations by the method [modelRates](#)

**Usage**

```
modelingParams(object)

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

**Arguments**

object            An object of class INSPEcT

**Value**

List of parameters and their values

- 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.
- useSigmoidFun A logical, whether to choose between sigmoid and impulse function to fit rates and concentrations. In case not, always impulse function is used.
- testOnSmooth A logical, wheter models should be tested on smoothed pre-mRNA, total mRNA and eventually synthesis rates or not.
- nInit number of optimization to find the best functional representation of each rate
- nIter number of max iteration during optimization
- Dmin lower bondary for degradation rates in the NoNascent mode
- Dmax upper bondary for degradation rates in the NoNascent mode
- seed A numeric, indicatindg the seed set for reproducible results.

**See Also**

[modelRates](#)

**Examples**

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

---

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 options used for the modeling can be later accessed by the user via [modelingParams](#). After modeling, model selection is run by the method [calculateRatePvals](#) with default parameters.

**Usage**

```

modelRates(
  object,
  estimateRatesWith = c("der", "int"),
  useSigmoidFun = TRUE,
  nInit = 10,
  nIter = 300,
  Dmin = 1e-06,
  Dmax = 10,
  seed = NULL,
  BPPARAM = bpparam()
)

## S4 method for signature 'INSPECT'
modelRates(
  object,
  estimateRatesWith = c("der", "int"),
  useSigmoidFun = TRUE,
  nInit = 10,
  nIter = 300,
  Dmin = 1e-06,
  Dmax = 10,
  seed = NULL,
  BPPARAM = bpparam()
)

```

**Arguments**

<code>object</code>	An object of class <code>INSPECT</code>
<code>estimateRatesWith</code>	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 "der")
<code>useSigmoidFun</code>	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)
<code>nInit</code>	number of optimization to find the best functional representation of each rate (by default 10)
<code>nIter</code>	number of max iteration during optimization (default is 300)
<code>Dmin</code>	lower boundary for degradation rates in the NoNascent mode (default 1e-06)
<code>Dmax</code>	upper boundary for degradation rates in the NoNascent mode (default 10)
<code>seed</code>	A numeric, indicating the seed to be set for reproducible results. If NULL it is randomly selected (default NULL)
<code>BPPARAM</code>	Parallelization parameters for <code>bplapply</code> . By default <code>bpparam()</code>

**Value**

An object of class `INSPECT` with modeled rates



**See Also**

[viewModelRates](#), [calculateRatePvals](#), [geneClass](#)

**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)
}
```

---

modelRatesNF	<i>Launch the modeling process without imposing sigmoid/impulse functional form</i>
--------------	---

---

**Description**

This method compute confidence intervals for the rates of synthesis, degradation and processing estimated by [newINSPEcT](#) that will be used to estimate the variability of each rate in [ratePvals](#) method.

**Usage**

```
modelRatesNF(object, BPPARAM = SerialParam())

## S4 method for signature 'INSPEcT'
modelRatesNF(object, BPPARAM = SerialParam())
```

**Arguments**

object	An object of class INSPEcT
BPPARAM	Parallelization parameters for bplapply. By default bpparam()

**Value**

An object of class INSPEcT with modeled rates

**Examples**

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

```
viewModelRates(nascentInspObjThreeGenes, 'synthesis')
## view gene classes
geneClass(nascentInspObjThreeGenes)
}
```

---

modelSelection	<i>Visualize criteria used for rate variability</i>
----------------	---

---

### Description

Method to visualize the criteria used to assess variability of rates.

### Usage

```
modelSelection(object)

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

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

### Arguments

`object` An object of class `INSPEcT` or `INSPEcT_model`

### Value

- `modelSelection 'aic'` compares nested models closest to the one with lowest AIC, `'llr'` compares all nested models, `'hib'` is a mix between the previous two. (default `'aic'`)
- `preferPValue` a logical, if `TRUE` (default) limit the search for best models among the ones with succeeded the goodness of fit test.
- `padj` a logical, if `TRUE` (default) correct the p-values for multiple testing
- `goodness_of_fit` a numeric, the threshold for the goodness-of-fit test (default = .1)
- `variability` a numeric, a vector with the thresholds for the variability test (one threshold for each rate, default = `c('s'=05, 'p'=.05, 'd'=.05)`)
- `limitModelComplexity` a logical that limits the complexity of the function used to describe dynamics to the length of the time-course (default = `FALSE`)

### Examples

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

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**

tpts	A vector of time points, one for each sample
labeling_time	A number, length of the Nascent pulse
nascentExpressions	A list which contains exons and introns expression matrices and variances for the nascent RNA
matureExpressions	A list which contains exons and introns expression matrices and variances for the mature RNA
preexisting	A logical, indicating if the mature expression refers to the pre-existing (unlabeled) population. Not implemented yet for the "degDuringPulse" mode.
BPPARAM	Configuration for BiocParallel parallelization. By default is set to bpparam()
labeledSF	A vector storing user defined normalization scale over Nascent RNA exons and introns quantifications
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
Dmin	A numerical, it is the lower bound of the degradation rate domain for the prior optimization

Dmax	A numerical, it is the upper bound of the degradation rate domain for the prior optimization
genesFilter	A logical, if TRUE, 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
  ,libsizes=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

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

*Plot the pre-modeled and modeled profiles for one gene***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, constantModel = FALSE)

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

**Arguments**

object	An object of class INSPEcT
ix	Eiher 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
constantModel	A logical, if true the constant model for the + nascent modeling are shown

**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 calss 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](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,
  BPPARAM = SerialParam()
)

```

**Arguments**

<code>txdb</code>	A TranscriptDB object
<code>BAMfiles</code>	A vector of paths
<code>experimentalDesign</code>	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.
<code>by</code>	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.
<code>countMultiMappingReads</code>	A logical, if multimapping reads should be counted, FALSE by default. Multimap reads are identified using the tag "NH" in the bam/sam file.
<code>allowMultiOverlap</code>	A logical, indicating if a read is allowed to be assigned to more than one feature, FALSE by default
<code>prioritizeExons</code>	A logical, indicating whether reads assigned to exon shold 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 were intronic reads are sampled at a very low rate compared to exonic reads. By default, TRUE.
<code>libsize</code>	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.
<code>strandSpecific</code>	Numeric, 0 if no strand-specific read counting should be performed, 1 stranded, 2 reversely-stranded. 0 by default
<code>isPairedEnd</code>	A logical, if paired-end reads are used, FALSE by default
<code>DESeq2</code>	A logical, if TRUE exons and introns variances are evaluated through the package DESeq2, if FALSE through plgem
<code>varSamplingCondition</code>	A character reporting which experimental condition should be used to sample the variance if DESeq2 = FALSE.
<code>BPPARAM</code>	Parallelization parameters for bplapply. By default SerialParam() 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 intorns 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 relative to the variability of synthesis, processing and degradation rates.

### Usage

```
ratePvals(object)
```

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

### Arguments

object            An object of class INSPEcT

### Examples

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

---

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      *Retrieve pre-modeling rates and concentrations variance*

---

### 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 retireve, "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)
## Not run: nascentInspObj10_new <- modelRates(nascentInspObj10_new, useSigmoidFun = FALSE)

```

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, plot = TRUE, comparative = FALSE)

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

```

**Arguments**

object	An object of class INSPEcT_model, with true rates
object2	An modeled object of class INSPEcT
plot	A logical indicating whether ROC curves should be plotted or not
comparative	A logical indicating whether the cross-prediction should be visualized. When this mode is selected, the p-values assigned to the variability of one rate (e.g. synthesis) are tested against the variability the other rates (e.g. processing and degradation). Cross-prediction ROC curves are plotted with dashed lines.



**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

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

---

**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, xlim = c(1e-05, 1), plot = TRUE)

## S4 method for signature 'INSPEcT_model,INSPEcT'
rocThresholds(object, object2, xlim = c(1e-05, 1), 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
xlim	A numeric representing limits for the x-axis (default is c(1-e-5,1))
plot	A logical that indicates whether to plot or not. (default=TRUE)

**Value**

The thresholds that maximize both sensitivity and specificity

**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)
}
```

---

setConfidenceIntervals

*Set confidence intervals*

---

**Description**

This function is used to set the confidence intervals in the nascent RNA mode.

**Usage**

```
setConfidenceIntervals(object, confidenceIntervals)

## S4 method for signature 'INSPEcT'
setConfidenceIntervals(object, confidenceIntervals)
```

**Arguments**

object            An object of class INSPECT\_model  
 confidenceIntervals    list of confidence intervals.

**Value**

An object of class ExpressionSet containing the confidence intervals.

---

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    *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: 0,1/6,1/3,1/2,1,1.25,1.5,2,3,4,6,8,10,12,16 hours.

**Format**

An INSPECT object

---

`simData4rep_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

---

`split` *Divide an INSPEcT Object into groups*

---

### Description

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

### Usage

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

### Arguments

<code>x</code>	An object of class INSPEcT
<code>f</code>	A vector of length equal to the number of genes in <code>x</code> which defines the groups
<code>drop</code>	A logical belonging to the generic function, useless in this context.
<code>...</code>	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	<i>Accessor to the slot tpts of an INSPEcT object</i>
------	---

---

**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)
```

---

viewConfidenceIntervals	<i>Retrieve the modeled Confidence Intervals</i>
-------------------------	--

---

**Description**

A method to access the modeled confidence intervals computed via the method [computeConfidenceIntervals](#)

**Usage**

```
viewConfidenceIntervals(object, feature)

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

**Arguments**

object            An object of class INSPEcT  
feature           A character indicating the feature to retrieve: "synthesis", "degradation", "processing".

**Value**

A numeric matrix containing the values for the selected feature

**Examples**

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

---

viewModelRates	<i>Retrieve the modeled rates and concentrations</i>
----------------	--

---

**Description**

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**

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

# Index

- [, [17](#), [24](#)
- [, INSPECT, ANY, ANY, ANY-method (Extract), [12](#)
- [, INSPECT\_diffsteady, ANY, ANY, ANY-method (Extract), [12](#)
- [, INSPECT\_model, ANY, ANY, ANY-method (Extract), [12](#)
  
- AIC, [17](#), [24](#)
- AIC, INSPECT-method (AIC-INSPECT-method), [3](#)
- AIC, INSPECT\_model-method (AIC-INSPECT-method), [3](#)
- AIC-INSPECT-method, [3](#)
- allcounts, [4](#)
  
- calculateDelta, [17](#)
- calculateDelta (processingDelay), [39](#)
- calculateDelta, INSPECT-method (processingDelay), [39](#)
- calculateRatePvals, [4](#), [14](#), [17](#), [31](#), [33](#)
- calculateRatePvals, INSPECT-method (calculateRatePvals), [4](#)
- calculateTau, [17](#)
- calculateTau (processingDelay), [39](#)
- calculateTau, INSPECT-method (processingDelay), [39](#)
- chisqmodel, [5](#), [17](#)
- chisqmodel, INSPECT-method (chisqmodel), [5](#)
- chisqtest, [6](#), [17](#), [24](#)
- chisqtest, INSPECT-method (chisqtest), [6](#)
- chisqtest, INSPECT\_model-method (chisqtest), [6](#)
- combine, [7](#), [17](#)
- combine, INSPECT, INSPECT-method (combine), [7](#)
- compareSteady, [8](#), [17](#), [21](#)
- compareSteady, INSPECT-method (compareSteady), [8](#)
- compareSteadyNoNascent, [9](#), [17](#)
- compareSteadyNoNascent, INSPECT-method (compareSteadyNoNascent), [9](#)
- computeConfidenceIntervals, [10](#), [17](#), [53](#)
- computeConfidenceIntervals, INSPECT-method (computeConfidenceIntervals), [10](#)
- convergence, [11](#)
- convergence, INSPECT-method (convergence), [11](#)
- correlationPlot, [11](#), [17](#), [24](#)
- correlationPlot, INSPECT\_model, INSPECT-method (correlationPlot), [11](#)
  
- degradation (INSPECT\_diffsteady-class), [21](#)
- degradation, INSPECT\_diffsteady-method (INSPECT\_diffsteady-class), [21](#)
- dim, [17](#)
- dim, INSPECT-method, [12](#)
  
- Extract, [12](#)
  
- featureNames, [17](#)
- featureNames, INSPECT-method, [13](#)
- featureNames, INSPECT\_diffsteady-method (INSPECT\_diffsteady-class), [21](#)
- featureNames<- , INSPECT-method (featureNames, INSPECT-method), [13](#)
- featureWidths, [14](#)
  
- geneClass, [14](#), [17](#), [24](#), [31](#), [33](#)
- geneClass, INSPECT-method (geneClass), [14](#)
- geneClass, INSPECT\_diffsteady-method (geneClass), [14](#)
- geneClass, INSPECT\_model-method (geneClass), [14](#)
  
- inHeatmap, [15](#), [17](#)
- inHeatmap, INSPECT-method (inHeatmap), [15](#)
- INSPECT, [16](#), [17](#)
- INSPECT-class, [17](#)
- INSPECT-shinyApps, [19](#)
- INSPECT\_diffsteady, [17](#)
- INSPECT\_diffsteady-class, [21](#)
- INSPECT\_model-class, [23](#)
- inspectFromBAM, [19](#)
- inspectFromPCR, [20](#)

- labeledSF, [17, 24](#)
- labeledSF, INSPECT-method (labeledSF), [24](#)
- libsizes, [25](#)
- logLik, [17, 24, 25](#)
- logLik, INSPECT-method (logLik), [25](#)
- logLik, INSPECT\_model-method (logLik), [25](#)
  
- makeModelRates, [18, 24, 26](#)
- makeModelRates, INSPECT-method (makeModelRates), [26](#)
- makeModelRates, INSPECT\_model-method (makeModelRates), [26](#)
- makeOscillatorySimModel, [18, 27](#)
- makeOscillatorySimModel, INSPECT-method (makeOscillatorySimModel), [27](#)
- makeSimDataset, [5, 23, 24, 27, 28, 29, 30, 49, 50](#)
- makeSimDataset, INSPECT\_model-method (makeSimDataset), [28](#)
- makeSimModel, [5, 18, 23, 28, 29, 29, 30, 49, 50](#)
- makeSimModel, INSPECT-method (makeSimModel), [29](#)
- modelingParams, [17, 18, 30, 31](#)
- modelingParams, INSPECT-method (modelingParams), [30](#)
- modelRates, [18, 23, 26, 30, 31, 31, 48, 49, 54](#)
- modelRates, INSPECT-method (modelRates), [31](#)
- modelRatesNF, [18, 33](#)
- modelRatesNF, INSPECT-method (modelRatesNF), [33](#)
- modelSelection, [18, 24, 34](#)
- modelSelection, INSPECT-method (modelSelection), [34](#)
- modelSelection, INSPECT\_model-method (modelSelection), [34](#)
  
- newINSPECT, [17, 31, 33, 35, 46, 47](#)
- nGenes, [12, 18, 36](#)
- nGenes, INSPECT-method (nGenes), [36](#)
- nTpts, [12, 18, 37](#)
- nTpts, INSPECT-method (nTpts), [37](#)
  
- plotGene, [18, 37](#)
- plotGene, INSPECT-method (plotGene), [37](#)
- plotMA, [22, 38](#)
- plotMA, INSPECT\_diffsteady-method (plotMA), [38](#)
- processing (INSPECT\_diffsteady-class), [21](#)
- processing, INSPECT\_diffsteady-method (INSPECT\_diffsteady-class), [21](#)
- processingDelay, [18, 39](#)
- processingDelay, INSPECT-method (processingDelay), [39](#)
  
- quantifyExpressionsFromBAMs, [41](#)
- quantifyExpressionsFromTrAbundance, [43](#)
- quantifyExpressionsFromTrCounts, [44](#)
  
- ratePvals, [15, 18, 33, 45](#)
- ratePvals, INSPECT-method (ratePvals), [45](#)
- ratesFirstGuess, [18, 36, 46, 47](#)
- ratesFirstGuess, INSPECT-method (ratesFirstGuess), [46](#)
- ratesFirstGuessVar, [18, 46, 46](#)
- ratesFirstGuessVar, INSPECT-method (ratesFirstGuessVar), [46](#)
- removeModel, [18, 47](#)
- removeModel, INSPECT-method (removeModel), [47](#)
- rocCurve, [18, 24, 48, 50](#)
- rocCurve, INSPECT\_model, INSPECT-method (rocCurve), [48](#)
- rocThresholds, [18, 24, 49, 49](#)
- rocThresholds, INSPECT\_model, INSPECT-method (rocThresholds), [49](#)
- runINSPECTGUI (INSPECT-shinyApps), [19](#)
- runProcessingRateDelay (INSPECT-shinyApps), [19](#)
  
- setConfidenceIntervals, [18, 50](#)
- setConfidenceIntervals, INSPECT-method (setConfidenceIntervals), [50](#)
- show, [18, 24](#)
- show, INSPECT-method (INSPECT-class), [17](#)
- show, INSPECT\_diffsteady-method (INSPECT\_diffsteady-class), [21](#)
- show, INSPECT\_model-method (INSPECT\_model-class), [23](#)
- simData3rep\_Nascent, [51](#)
- simData3rep\_NoNascent, [51](#)
- simData4rep\_Nascent, [51](#)
- simData4rep\_NoNascent, [52](#)
- split, [18, 52](#)
- split, INSPECT, ANY-method (split), [52](#)
- synthesis (INSPECT\_diffsteady-class), [21](#)
- synthesis, INSPECT\_diffsteady-method (INSPECT\_diffsteady-class), [21](#)
  
- tpts, [18, 53](#)
- tpts, INSPECT-method (tpts), [53](#)
  
- viewConfidenceIntervals, [18, 53](#)
- viewConfidenceIntervals, INSPECT-method (viewConfidenceIntervals), [53](#)



viewModelRates, [18](#), [31](#), [33](#), [54](#)  
viewModelRates, INSPECT-method  
(viewModelRates), [54](#)