

Package ‘pulseTD’

May 17, 2019

Type Package

Title Identification of Transcriptional Dynamics using Pulse Models
via 4su-Seq Data and RNA-Seq Data

Version 0.1.0

Date 2019-5-5

Author XinWang

Maintainer XinWang <xindd_2014@163.com>

Description A tool for analyzing the transcription, processing and degradation rates of genes by 4sU-seq (the Metabolic Label 4-thiouridine) data and RNA-seq (RNA sequencing) data. It can not only recognize the transcriptional dynamic rates at the measurement time points, but also obtain continuous changes in transcriptional dynamics. More importantly, it is able to predict the trend of mRNA (Mature RNA) transcription and expression changes in the future.

License GPL-2

Encoding UTF-8

Depends R (>= 3.4.0)

biocViews

Imports AnnotationDbi, SummarizedExperiment, Rsamtools, Biobase,
S4Vectors, methods, parallel, GenomicFeatures, ggplot2, grid,
GenomicAlignments

RoxygenNote 6.1.1

Suggests knitr, rmarkdown, TxDb.Hsapiens.UCSC.hg19.knownGene

URL https://github.com/bioWzz/pulseTD_0.1.0

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2019-05-17 09:20:03 UTC

R topics documented:

correctionParams	2
estimateExpression	3
estimateParams	4
getParams	5
getParams,pulseTDmodel-method	6
getRates	7
getRates,pulseTDmodel-method	8
integrateA	8
integrateAC	9
plotRates	9
plotRates,pulseTDmodel-method	10
predictExpression	11
predictExpression,pulseTDmodel-method	11
preExp	12
pulseModel	12
pulseRates	13
pulseTDmodel-class	13
rpkmres	14
rpkmSim	14
splitA6Params	15
splitB4Params	15
splitC5Params	16
Index	17

correctionParams	<i>Correct the parameters of the pulse model</i>
------------------	--

Description

It is used to correct pulse model parameters. During the estimation of parameters in [estimateParams](#), some NA will occur, which may be caused by random initial values. Therefore, some NA values can be estimated from the new ones.

Usage

```
correctionParams(object)
```

Arguments

object a 'pulseTDmodel' that has been calculated with [estimateParams](#)

Value

a 'pulseTDmodel' that has been modified

Examples

```
data('pulseRates', package='pulseTD')
pulseRates_correct = correctionParams(pulseRates)
pulseRates_correct@fitfailure
```

estimateExpression	<i>Estimated Expression</i>
--------------------	-----------------------------

Description

This function is used to calculate the expression value of the gene. It includes total exon expression values, total intron expression values, labeled exon expression values, and labeled intron expression values.

Usage

```
estimateExpression(txdb, filelist, by = "gene", mode = "Union",
  singleEnd = TRUE, ignore.strand = TRUE, fragments = FALSE, ...)
```

Arguments

txdb	Genomic annotation file
filelist	A vector of sam file paths containing 4sU labeled sam files and unlabeled files
by	In terms of genes or transcripts, the default is the gene
mode	mode can be one of the pre-defined count methods such as "Union", "IntersectionStrict", or "IntersectionNotEmpty" or it a user supplied count function.
singleEnd	(Default TRUE) A logical indicating if reads are single or paired-end.
ignore.strand	A logical indicating if strand should be considered when matching
fragments	(Default FALSE) A logical; applied to paired-end data only. fragments control which function is used to read the data which subsequently affects which records are included in counting.
...	The parameters of the summarizeOverlaps method

Value

A list of expression

See Also

Other parameters [summarizeOverlaps](#)

Examples

```
library(TxDb.Hsapiens.UCSC.hg19.knownGene)
txdb <- TxDb.Hsapiens.UCSC.hg19.knownGene
test_path <- file.path(system.file(package="pulseTD"), 'extdata/test1.sorted.bam')
test_path2 <- file.path(system.file(package="pulseTD"), 'extdata/test2.sorted.bam')
rpkmres <- estimateExpression(txdb, c(test_path, test_path2), by='gene')
data('rpkmres', package='pulseTD')
head(rpkmres$total_exp)
head(rpkmres$pre_exp)
```

estimateParams	<i>Estimated pulse model parameters!</i>
----------------	--

Description

It is used to estimate pulse model parameters. Transcription rates, processing rates, and degradation correspond to different pulse model parameters, respectively. The transcription rate has 6 parameters, the processing rate has 5 parameters, and the degradation rate has 4 parameters.

Usage

```
estimateParams(labexon, totexon, totintr, TimeGrid, tL,
  clusterNumber = NULL, loopnumber = 50, message = TRUE)
```

Arguments

labexon	A matrix containing expression levels of 4su exons. For the calculation of the expression value, see estimateExpression
totexon	A matrix containing expression levels of total exons. For the calculation of the expression value, see estimateExpression
totintr	A matrix containing expression levels of total introns. For the calculation of the expression value, see estimateExpression
TimeGrid	A vector of points in time at which experimental data is collected, not allowed to be repeated
tL	4sU labeled time during the experiment
clusterNumber	Given the number of cluster cores, the default is the maximum number of available cores
loopnumber	The number of iterations of the gradient descent when solving the parameter. The default is 50.
message	Whether to print the log, the default is TRUE

Value

A 'pulseTDmodel' containing the expression values of the filtered genes, a list of solved parameters, and some basic parameter information

Examples

```

data('rpkmSim', package='pulseTD')
rpkm_TL <- rpkmSim$labexon[1:2,]
rpkm_PT <- rpkmSim$totintr[1:2,]
rpkm_TT <- rpkmSim$totexon[1:2,]
TimeGrid <- c(0, 15, 30, 45, 60, 75, 90, 105, 120, 135, 150, 165, 180)
tL <- 10

pulseRates<- estimateParams(rpkm_TL,rpkm_TT,rpkm_PT,TimeGrid,tL,clusterNumber=1,loopnumber=10)

```

getParams

Get pulseModel params

Description

It is used to obtain the parameters of the pulse function J Comput Biol (2009 Feb) <doi:10.1089/cmb.2008.13TT>, including transcription parameters, degradation parameters and processing parameters. You can use these parameters to analyze transcriptional characteristics, including steady-state analysis. At the same time, you can use [pulseModel](#) to view the curve corresponding to the parameter.

Usage

```
getParams(object, stage, genename = NULL)
```

Arguments

object	a 'pulseTDmodel' that has been calculated with estimateParams
stage	A character is one of three stages of transcriptional dynamics, transcription, processing and degradation.
genename	a vector, default is NULL <ul style="list-style-type: none"> • If it is NULL: Calculate the transcriptional dynamic rate of all genes • If it is a gene vector: only calculate the transcriptional dynamic rate of a given gene vector • If it is a numerical vector: only calculate the transcriptional dynamic rate of the gene corresponding to a given value.

Value

A matrix or vector containing six parameters: h0,h1,h2,t1 t2,beta

Examples

```

data('pulseRates', package='pulseTD')
TimeGrid <- c(0, 15, 30, 45, 60, 75, 90, 105, 120, 135, 150, 165, 180)
pulseRates_correct <- correctionParams(pulseRates)
transcription_params = getParams(pulseRates,'transcription')
degradation_params = getParams(pulseRates, 'degradation')
processing_params = getParams(pulseRates, 'processing')
head(transcription_params)
head(degradation_params)
head(processing_params)
transcription_params = getParams(pulseRates, 'transcription', genename=c(1,2,3))
head(transcription_params)
###
transcription_pulse = pulseModel(as.matrix(transcription_params[1,]), TimeGrid)
degradation_pulse = pulseModel(as.matrix(degradation_params[1,]), TimeGrid)
processing_pulse = pulseModel(as.matrix(processing_params[1,]), TimeGrid)

```

getParams,pulseTDmodel-method

getParams-method

Description

It is used to obtain transcription, processing and degradation rates.

Usage

```

## S4 method for signature 'pulseTDmodel'
getParams(object, stage, genename = NULL)

```

Arguments

object	An object of class 'pulseTDmodel'
stage	A character such as transcription, processing or degradation for rates.
genename	A vector of gene names, default is NULL.

Value

A numeric matrix containing the rates

getRates	<i>Get rate values</i>
----------	------------------------

Description

It is used to calculate the transcriptional dynamic rate of a gene. You can get the discrete or continuous rate values of the measurement time points. At the same time, it has a predictive function that provides rate values for any future time node or any range of time.

Usage

```
getRates(object, stage, timevector = NULL, genename = NULL)
```

Arguments

object	a 'pulseTDmodel' that has been calculated with estimateParams
stage	A character is one of three stages of transcriptional dynamics, transcription, processing and degradation.
timevector	A vector of time, which can be any time, defaults to NULL <ul style="list-style-type: none"> • If it is NULL: only calculate the transcriptional dynamics of the measurement time node, the time is obtained by t_time in pulseTDmodel • If the time vector is positive: calculate the rate value for a given time, or the predicted rate if the time exceeds the measurement time node • If the time vector is negative: Calculate the rate value at a given time, which represents the calculation of the transcriptional dynamic rate before the prediction of the experimental measurement.
genename	a vector, default is NULL <ul style="list-style-type: none"> • If it is NULL: Calculate the transcriptional dynamic rate of all genes • If it is a gene vector: only calculate the transcriptional dynamic rate of a given gene vector • If it is a numerical vector: only calculate the transcriptional dynamic rate of the gene corresponding to a given value.

Value

A matrix, each row representing a gene, each column with a table to calculate the time node, and the value representing the rate of transcriptional dynamics.

Examples

```
load(file.path(system.file(package="pulseTD"), "data", "pulseRates.RData"))
pulseRates_correct <- correctionParams(pulseRates)
transcription = getRates(pulseRates_correct, 'transcription')
degradation = getRates(pulseRates_correct, 'degradation')
processing = getRates(pulseRates_correct, 'processing')
head(transcription)
```

```

head(degradation)
head(processing)
trans = getRates(pulseRates_correct, 'transcription', timevector=c(0,1,2,3))
head(transcription)
trans=getRates(pulseRates_correct, 'transcription', timevector=c(2,3),genename=c(2,3))
head(transcription)

```

getRates,pulseTDmodel-method
getRates-method

Description

It is used to obtain transcription, processing and degradation rates.

Usage

```

## S4 method for signature 'pulseTDmodel'
getRates(object, stage, timevector = NULL,
         genename = NULL)

```

Arguments

object	An object of class 'pulseTDmodel'
stage	A character such as transcription, processing or degradation for rates.
timevector	A vector of times, default is NULL.
genename	A vector of gene names, default is NULL.

Value

A numeric matrix containing the rates

integrateA *integral function*

Description

Calculate the exon expression value of the label time.

Usage

```
integrateA(xita, timegrid, tl)
```


Arguments

xita	Six parameters of the pulse function :h0, h1, h2, t1, t2, beta
timegrid	a vector or point of time
t1	label time

Value

a vector or point of exon expression value

integrateAC	<i>integral function</i>
-------------	--------------------------

Description

Calculate the expression values at different time points.

Usage

```
integrateAC(xita, timegrid)
```

Arguments

xita	Six parameters of the pulse function :h0, h1, h2, t1, t2, beta
timegrid	a vector or point of time

Value

a vector or point of exon expression value

plotRates	<i>draw Method</i>
-----------	--------------------

Description

It is used to draw a rate image that contains six images. The expression values of the real genes and the fitted expression values are plotted separately, and the rate of the change is varied with time. Can also plot predicted expression values and rate values

Usage

```
plotRates(object, genename, predict = FALSE)
```

Arguments

object	a pulseTDmodel that has been calculated with estimateParams
genename	The name of the gene to be drawn
predict	The time interval you want to draw,default is FALSE, or a vector of three values,(start, end, step)

Value

There is no return value, six pictures will be drawn separately.

Examples

```
data('pulseRates', package='pulseTD')
plotRates(pulseRates, 'NM_001001181')
plotRates(pulseRates, 'NM_001001181', predict=c(0,180,20))
```

plotRates,pulseTDmodel-method
plotRate-Method

Description

Draw the image of the fitting result.

Usage

```
## S4 method for signature 'pulseTDmodel'
plotRates(object, genename, predict = FALSE)
```

Arguments

object	An object of class 'pulseTDmodel'
genename	The name of the gene you want to draw.
predict	The time interval you want to draw.

Value

No return value

predictExpression *predict Expression*

Description

It is used to predict the expression of all gene at a given time, including the expression of pre-mRNA (Precursor RNA) and the expression of total mRNA (Mature RNA). End time and time interval can be arbitrarily defined

Usage

```
predictExpression(object, tg)
```

Arguments

object	a 'pulseTDmodel' that has been calculated with estimateParams
tg	A vector of points in time at which experimental data is collected, not allowed to be repeated

Value

Returns a list containing predicted values for each gene and a 0.95 confidence interval

Examples

```
data('pulseRates', package='pulseTD')
pulseRates_correct = correctionParams(pulseRates)
TimeGrid = seq(0,180,15)
preExp = predictExpression(pulseRates_correct, tg=TimeGrid)
data('preExp', packages='pulseTD')
df = data.frame(preExp[['NM_001002011']])
head(df)
```

predictExpression, pulseTDmodel-method
predictExpression-method

Description

It is used to predict the expression of all gene at a given time, including the expression of pre-mRNA (Precursor RNA) and the expression of total mRNA (Mature RNA). End time and time interval can be arbitrarily defined

Usage

```
## S4 method for signature 'pulseTDmodel'
predictExpression(object, tg)
```

Arguments

object	a 'pulseTDmodel' that has been calculated with estimateParams
tg	A vector of points in time at which experimental data is collected, not allowed to be repeated

Value

Returns a list containing predicted values for each gene and a 0.95 confidence interval

Examples

```
data('pulseRates', package='pulseTD')
pulseRates_correct = correctionParams(pulseRates)
TimeGrid = seq(0,180,15)
preExp = predictExpression(pulseRates_correct, tg=TimeGrid)
data('preExp', packages='pulseTD')
df = data.frame(preExp[['NM_001002011']])
head(df)
```

preExp	<i>A gene list that stores predicted values of gene expression and a 0.95 confidence interval.</i>
--------	--

Description

A gene list that stores predicted values of gene expression and a 0.95 confidence interval.

Format

A list

pulseModel	<i>pulse function</i>
------------	-----------------------

Description

Pulse function is a pulse function J Comput Biol (2009 Feb) <doi:10.1089/cmb.2008.13TT>. This is a multiplication of two sigmoid functions, the parameter vector Theta = (h0, h1, h2, t1 t2, beta), where h0, h1, h2, represent the initial state rate value, peak value and the steady state rate value is reached again, t1 and t2 are the maximum times of the first and second rise or fall changes, respectively, and beta is the slope of the two changes.

Usage

```
pulseModel(xita, x)
```

Arguments

`xi` Six parameters of the pulse function :h0, h1, h2, t1, t2, beta. Details [getParams](#)
`x` a vector or point of time

Value

a vector or point of pulse function value

Examples

```
load(file.path(system.file(package="pulseTD"), "data", "pulseRates.RData"))
TimeGrid <- c(0, 15, 30, 45, 60, 75, 90, 105, 120, 135, 150, 165, 180)
pulseRates_correct <- correctionParams(pulseRates)
transcription_params = getParams(pulseRates_correct, 'transcription')
degradation_params = getParams(pulseRates_correct, 'degradation')
processing_params = getParams(pulseRates_correct, 'processing')
###
transcription_pulse = pulseModel(as.matrix(transcription_params[1,]), TimeGrid)
degradation_pulse = pulseModel(as.matrix(degradation_params[1,]), TimeGrid)
processing_pulse = pulseModel(as.matrix(processing_params[1,]), TimeGrid)
```

pulseRates	<i>A pulseTDMModel Rdata file containing 100 gene expression data, transcriptional dynamic data already related parameters.</i>
------------	---

Description

A pulseTDMModel Rdata file containing 100 gene expression data, transcriptional dynamic data already related parameters.

Format

A pulseTD model

pulseTDmodel-class	<i>pulseTD-model</i>
--------------------	----------------------

Description

'pulseTD-model' is S4 class for storing calculation results. It includes pulse model parameters corresponding to transcription rate, processing rate and degradation rate, and also stores some basic parameters of the calculation process, such as gene name, sequencing time, labeling time and so on.

Slots

genenames All gene names entered by the user.
 t_time Storage time series.
 tL 4sU labeled duration
 fitfailure Gene vector for parameter optimization failure
 filterExpression.foursu_exons 'AnnotatedDataFrame' that stores 4sU-labeled exons expression values
 filterExpression.total_exons 'AnnotatedDataFrame' that stores total exons expression values
 filterExpression.total_introns 'AnnotatedDataFrame' that stores total introns expression values
 ratesPar.degradation 'AnnotatedDataFrame' that stores degradation pulse parameters
 ratesPar.processing 'AnnotatedDataFrame' that stores processing pulse parameters
 ratesPar.transcription 'AnnotatedDataFrame' that stores transcription pulse parameters
 ratesPar.score 'AnnotatedDataFrame' that stores the fit scores

rpkmres	<i>a list containing the expression values of exons and introns</i>
---------	---

Description

a list containing the expression values of exons and introns

Format

A list

rpkmSim	<i>A simulated gene expressing Rdata file that contains expression data for total RNA, precursor RNA, and 4sU-labeled RNA.</i>
---------	--

Description

A simulated gene expressing Rdata file that contains expression data for total RNA, precursor RNA, and 4sU-labeled RNA.

Format

A list

splitA6Params	<i>estimate transcription parameters</i>
---------------	--

Description

It is used to solve the transcription parameters, including the error function and the optimization function.

Usage

```
splitA6Params(TL, tL, TimeGrid)
```

Arguments

TL	Exon expression value of 4sU label RNA
tL	label time
TimeGrid	a vector of times

Value

a vector of transcription parameters, contains pulse parameters and fitting error values

splitB4Params	<i>estimate degradation parameters</i>
---------------	--

Description

It is used to solve the degradation parameters, including the error function and the optimization function.

Usage

```
splitB4Params(TT, pre_a, TimeGrid)
```

Arguments

TT	exon expression value of total RNA
pre_a	Parameter vector of transcription pulse function
TimeGrid	a vector of times

Value

a vector of degradation parameters, contains pulse parameters and fitting error values

splitC5Params	<i>estimate processing parameters</i>
---------------	---------------------------------------

Description

It is used to solve the processing parameters, including the error function and the optimization function.

Usage

```
splitC5Params(PT, pre_a, TimeGrid)
```

Arguments

PT	Intron expression value of total RNA
pre_a	Parameter vector of transcription pulse function
TimeGrid	a vector of times

Value

a vector of processing parameters, contains pulse parameters and fitting error values

Index

correctionParams, 2

estimateExpression, 3, 4
estimateParams, 2, 4, 5, 7, 10–12

getParams, 5, 13
getParams, pulseTDmodel-method, 6
getRates, 7
getRates, pulseTDmodel-method, 8

integrateA, 8
integrateAC, 9

plotRates, 9
plotRates, pulseTDmodel-method, 10
predictExpression, 11
predictExpression, pulseTDmodel-method,
11

preExp, 12
pulseModel, 5, 12
pulseRates, 13
pulseTDmodel (pulseTDmodel-class), 13
pulseTDmodel-class, 13

rpkmres, 14
rpkmSim, 14

splitA6Params, 15
splitB4Params, 15
splitC5Params, 16
summarizeOverlaps, 3