Package ‘InPAS’

April 26, 2017

Type Package
Title Identification of Novel alternative PolyAdenylation Sites (PAS)
Version 1.8.0
Date 2017-04-10
Author Jianhong Ou, Sung Mi Park, Michael R. Green and Lihua Julie Zhu
Maintainer Jianhong Ou <jianhong.ou@umassmed.edu>, Lihua Julie Zhu <Julie.Zhu@umassmed.edu>
Description Alternative polyadenylation (APA) is one of the important post-transcriptional regulation mechanisms which occurs in most human genes. InPAS facilitates the discovery of novel APA sites from RNAseq data. It leverages cleanUpdTSeq to fine tune identified APA sites.
biocViews RNASeq, Sequencing, AlternativeSplicing, Coverage, DifferentialSplicing, GeneRegulation, Transcription
License GPL (>= 2)
Lazyload yes
Imports AnnotationDbi, BSgenome, cleanUpdTSeq, Gviz, seqinr, preprocessCore, IRanges, GenomeInfoDb, depmixS4, limma, BiocParallel
Depends R (>= 3.1), methods, Biobase, GenomicRanges, GenomicFeatures, S4Vectors
Suggests RUnit, BiocGenerics, BiocStyle, BSgenome.Hsapiens.UCSC.hg19, BSgenome.Mmusculus.UCSC.mm10, org.Hs.eg.db, org.Mm.eg.db, TxDb.Hsapiens.UCSC.hg19.knownGene, TxDb.Mmusculus.UCSC.mm10.knownGene, rtracklayer, knitr
VignetteBuilder knitr
NeedsCompilation no

R topics documented:

InPAS-package ................................................................. 2
coverageFromBedGraph ....................................................... 3
coverageRate ................................................................. 4
covThreshold ............................................................... 5
CPsites .............................................................. 6
InPAS-package

alternative polyadenylation and cleavage estimations

Description

predict and estimate the alternative polyadenylation and cleavage site for mRNA-seq data

Details
coverageFromBedGraph

Package: InPAS
Type: Package
Version: 1.0
Date: 2014-09-12
License: GPL (>= 2)

Author(s)
Jianhong Ou, Sung Mi Park, Michael R. Green and Lihua Julie Zhu
Maintainer: Jianhong Ou <jianhong.ou@umassmed.edu>

References

coverageFromBedGraph  read coverage from bedGraph files

Description
read coverage from bedGraph files and save as a list.

Usage
coverageFromBedGraph(bedgraphs, tags, genome,
    hugeData=FALSE, BPPARAM=NULL, ...)

Arguments

bedgraphs        The file names of bedgraphs generated by bedtools. eg: bedtools genomecov -bg -split -ibam $bam -g mm10.size.txt > $bedgraph

tags             the names for each input bedgraphs

genome           an object of BSgenome

hugeData         is this dataset consume too much memory? if it is TRUE, the coverage will be saved into tempfiles.

BPPARAM   An optional BiocParallelParam instance determining the parallel back-end to be used during evaluation, or a list of BiocParallelParam instances, to be applied in sequence for nested calls to bplapply.

... parameters can be passed into tempfile. This is useful when you submit huge dataset to cluster.

Value
return a list of coverage for each bedgraph files. For each item in the list, it is a list of coverage for each chromosome. And the chromosome must start from "chr".
coverageRate

Author(s)
Jianhong Ou

Examples

if(interactive()){
  library(BSgenome.Mmusculus.UCSC.mm10)
  path <- file.path(find.package("InPAS"), "extdata")
  bedgraphs <- file.path(path, "Baf3.extract.bedgraph")
  data(utr3.mm10)
  tags <- "Baf3"
  genome <- BSgenome.Mmusculus.UCSC.mm10
  coverage <-
    coverageFromBedGraph(bedgraphs, tags, genome, hugeData=FALSE)
}

coverageRate

coverage rate of genes and 3UTRs

Description

calculate coverage rate of gene and 3UTRs. This function is used for quality control of the coverage. The coverage rate can show the complexity of RNA-seq library.

Usage

coverageRate(coverage, txdb, genome,
            cutoff_readsNum=1,
            cutoff_expdGene_cvgRate=0.1,
            cutoff_expdGene_sampleRate=0.5,
            which=NULL, ...)

Arguments

coverage coverage for each sample, output of coverageFromBedGraph
txdb an object of TxDb
genome an object of BSgenome
cutoff_readsNum cutoff reads number. If the coverage in the location is greater than cutoff_readsNum, the location will be treated as covered by signal.
cutoff_expdGene_cvgRate, cutoff_expdGene_sampleRate
cutoff_expdGene_cvgRate and cutoff_expdGene_sampleRate are the parameters used to calculate which gene is expressed in all input dataset. cutoff_expdGene_cvgRate set the cutoff value for the coverage rate of each gene; cutoff_expdGene_sampleRate set the cutoff value for ratio of numbers of expressed and all samples for each gene. For example, by default, cutoff_expdGene_cvgRate=0.1 and cutoff_expdGene_sampleRate=0.5. Suppose there are 4 samples, for one gene, if the coverage rates by base are: 0.05, 0.12, 0.2, 0.17, this gene will be counted as expressed gene because mean(c(0.05, 0.12, 0.2, 0.17) > cutoff_expdGene_cvgRate) > cutoff_expdGene_sampleRate if the coverage rates by base are: 0.05, 0.12, 0.07, 0.17, this gene will be counted as un-expressed gene because mean(c(0.05, 0.12, 0.07, 0.17) > cutoff_expdGene_cvgRate) <= cutoff_expdGene_sampleRate
covThreshold

which an object of GRanges or NULL. If it is not NULL, only the exons overlapping the given ranges are used.

Value

return a data from with col names: gene.coverage.rate: coverage per base for all genes, expressed.gene.coverage.rate: coverage per base for expressed genes, UTR3.coverage.rate: coverage per base for all 3' UTRs, UTR3.expressed.gene.subset.coverage.rate: coverage per base for 3' UTRs of expressed genes. and row names: the names of coverage.

Author(s)

Jianhong Ou

Examples

if(interactive()){
  library(BSgenome.Mmusculus.UCSC.mm10)
  library(TxDb.Mmusculus.UCSC.mm10.knownGene)
  path <- file.path(find.package("InPAS"), "extdata")
  bedgraphs <- c(file.path(path, "Baf3.extract.bedgraph"),
                  file.path(path, "UM15.extract.bedgraph"))
  hugeData <- FALSE
  coverage <- coverageFromBedGraph(bedgraphs,
                                 tags=c("Baf3", "UM15"),
                                 genome=BSgenome.Mmusculus.UCSC.mm10,
                                 hugeData=hugeData)
  coverageRate(coverage,
               txdb=TxDb.Mmusculus.UCSC.mm10.knownGene,
               genome=BSgenome.Mmusculus.UCSC.mm10,
               which = GRanges("chr6", ranges=IRanges(98013000, 140678000)))
}

covThreshold calculate the cutoff threshold of coverage

description

calculate the cutoff threshold of coverage for long form and short form

Usage

covThreshold(coverage, genome, txdb, utr3,
             chr="chr1", hugeData, groupList)

Arguments

covThreshold coverage for each sample, output of coverageFromBedGraph
   genome an object of BSgenome
tx db an object of TxDb
predict the alternative cleavage and polyadenylation (CP or APA) site.

Usage

CPsites(coverage, groupList=NULL, genome, utr3, window_size=100, search_point_START=50, search_point_END=NA, cutStart=window_size, cutEnd=0, adjust_distal_polyA_end=TRUE, coverage_threshold=5, long_coverage_threshold=2, background=c("same_as_long_coverage_threshold", "1K", "5K", "10K", "50K"), txdb=NA, PolyA_PWM=NA, classifier=NA, classifier_cutoff=.8, step=1, two_way=FALSE, shift_range=window_size, BPPARAM=NULL, tmpfolder=NULL, silence=TRUE)

Arguments

coverage coverage for each sample, output of coverageFromBedGraph
groupList group list of tag names
genome an object of BSgenome
utr3 output of utr3Annotation
window_size window size for noval distal position searching and adjusted polyA searching, default: 100
search_point_START start point for searching
search_point
end point for searching
cutStart
how many nucleotides should be removed from the start before search, 0.1 means 10 percent, 25 means cut first 25.
cutEnd
how many nucleotides should be removed from the end before search, 0.1 means 10 percent.

adjust_distal_polyA_end
If true, adjust distal polyA end by cleanUpdTSeq
coverage_threshold
cutoff coverage threshold for first 100 nucleotides. If the coverage of first 100 nucleotides is lower than coverage_threshold, that transcript will be dropped.

long_coverage_threshold
cutoff threshold for coverage in the region of long form. If the coverage in the region of long form is less than long_coverage_threshold, that transcript will be dropped.

background
the range for calculating cutoff threshold of local background
txdb
an object of TxDb
PolyA_PWM
Position Weight Matrix of polyA
classifier
An object of class "PASclassifier"
classifier_cutoff
This is the cutoff used to assign whether a putative pA is true or false. This can be any floating point number between 0 and 1. For example, classifier_cutoff = 0.5 will assign an putative pA site with prob.1 > 0.5 to the True class (1), and any putative pA site with prob.1 <= 0.5 as False (0).

step
adjust step, default 1, means adjust by each base by cleanUpdTSeq.
two_way
Search the proximal site from both direction or not.
shift_range
the shift range for polyA site searching

BPPARAM
An optional BiocParallelParam instance determining the parallel back-end to be used during evaluation, or a list of BiocParallelParam instances, to be applied in sequence for nested calls to bplapply.

tmpfolder
temp folder could save and reload the analysis data for resume analysis.
silence
report progress or not. default not report.

Value
return an object of GRanges contain the estimated CP sites.

Author(s)
Jianhong Ou

References
mappability could be calculated by [GEM](http://algorithms.cnag.cat/wiki/Man:gem-mappability)
Examples

```r
if(interactive()){
  library(BSgenome.Mmuscule.UCSC.mm10)
  path <- file.path(find.package("InPAS"), "extdata")
  bedgraphs <- file.path(path, "Baf3.extract.bedgraph")
  data(utr3.mm10)
  tags <- "Baf3"
  genome <- BSgenome.Mmuscule.UCSC.mm10
  coverage <-
    coverageFromBedGraph(bedgraphs, tags, genome, hugeData=FALSE)
  CP <- CPsites(coverage=coverage, gp1=tags, gp2=NULL, genome=genome,
                utr3=utr3.mm10, coverage_threshold=5, long_coverage_threshold=5)
}
```

### CPsite_estimation

*estimate the cpsites*

**Description**

estimate the cpsites for a giving chromosome

**Usage**

```r
CPsite_estimation(chr.cov, utr3, MINSIZE, window_size, search_point_START,
                  search_point_END, cutStart, cutEnd, adjust_distal_polyA_end,
                  background, z2s, coverage_threshold, long_coverage_threshold,
                  PolyA_PWM, classifier, classifier_cutoff, shift_range,
                  depth.weight, genome, step=1, two_way=FALSE,
                  tmpfolder=NULL, silence=TRUE)
```

**Arguments**

- `chr.cov`: coverage list for one chromosome
- `utr3`: output of utr3Annotation
- `MINSIZE`: min size of short form
- `window_size`: window size
- `search_point_START`: search start point
- `search_point_END`: search end point
- `cutStart`: cut from start
- `cutEnd`: cut from end
- `adjust_distal_polyA_end`: adjust distal site or not
- `background`: how to get the local background
- `z2s`: output of `zScoreThreshold`
- `coverage_threshold`: cutoff value for coverage
depthWeight

long_coverage_threshold
cutoff value for long form
PolyA_PWM	polyA PWM
classifier
classifier
classifier_cutoff
classifier cutoff
shift_range
shift range
depth.weight
output of depthWeight
genome
a BSgenome object
step
adjust step, default 1, means adjust by each base by cleanUpdTSeq.
two_way
Search the proximal site from both direction or not.
tmpfolder
temp folder could save and reload the analysis data for resume analysis.
silence
report progress or not. default not report.

Value

a data.frame

Author(s)

Jianhong Ou

See Also

CPsites, searchProximalCPs, proximalAdj, proximalAdjByPWM, proximalAdjByCleanUpdTSeq, PAscore, PAscore2

depthWeight calculate the depth weight for each example

Description

calculate the depth weight for each example

Usage

depthWeight(coverage, hugeData, groupList=NULL)

Arguments

coverage
a list. output of coverageFromBedGraph
hugeData
is it a huge dataset?
groupList
group list for huge dataset

Value

a numeric vector with depth weight

Author(s)

Jianhong Ou
distalAdj | adjust distal CP sites by cleanUpdTSeq

**Description**

adjust distal CP sites by cleanUpdTSeq

**Usage**

```r
distalAdj(distalCPs, classifier, classifier_cutoff, shift_range, genome, step=1)
```

**Arguments**

- `distalCPs`: the output of `searchDistalCPs`
- `classifier`: cleanUpdTSeq classifier
- `classifier_cutoff`: cutoff value of the classifier
- `shift_range`: the searching range for the better CP sites
- `genome`: a `BSgenome` object
- `step`: adjust step, default 1, means adjust by each base by cleanUpdTSeq.

**Value**

a list could be input of `searchProximalCPs`

**Author(s)**

Jianhong Ou

**See Also**

- `searchDistalCPs`, `PAscore2`

---

filterRes | filter results

**Description**

filter results of `testUsage`

**Usage**

```r
filterRes(res, gp1, gp2, background_coverage_threshold=2, P.Value_cutoff=0.05, adj.P.Val_cutoff=0.05, dPDUI_cutoff=0.3, PDUI_logFC_cutoff)
```
filterRes

Arguments
res output of testUsage
gp1 tag names involved in group 1
gp2 tag names involved in group 2
background_coverage_threshold
background coverage cut off value. for each group, more than half of the long
form should greater than background_coverage_threshold. for both group, at
least in one group, more than half of the short form should greater than back-
ground_coverage_threshold.
P.Value_cutoff cutoff of P value
adj.P.Val_cutoff cutoff of adjust P value
dPDUI_cutoff cutoff of dPDUI
PDUI_logFC_cutoff cutoff of PDUI log2 transformed fold change

Value
a data.frame

Author(s)
Jianhong Ou

See Also
testUsage

Examples
path <- file.path(find.package("InPAS"), "extdata")
load(file.path(path, "CPs.MAQC.rda"))
load(file.path(path, "coverage.MAQC.rda"))
library(BSgenome.Hsapiens.UCSC.hg19)
data(utr3.hg19)
res <- testUsage(CPsites=CPs,
 coverage=coverage,
 genome=BSgenome.Hsapiens.UCSC.hg19,
 utr3=utr3.hg19,
 method="fisher.exact",
 gp1=c("Brain.auto", "Brain.phiX"),
 gp2=c("UHR.auto", "UHR.phiX"))
filterRes(res,
 gp1=c("Brain.auto", "Brain.phiX"),
 gp2=c("UHR.auto", "UHR.phiX"),
 background_coverage_threshold=2,
 P.Value_cutoff=0.05,
 adj.P.Val_cutoff=0.05,
 dPDUI_cutoff=0.3,
 PDUI_logFC_cutoff=-.59)
get.regions.coverage

fisher.exact.test  
do fisher exact test for two group datasets

**Description**
do fisher exact test for two group datasets

**Usage**

```r
cisher.exact.test(UTR3eset, gp1, gp2)
```

**Arguments**

- `UTR3eset`: output of `getUTR3eSet`
- `gp1`: tag names of group 1
- `gp2`: tag names of group 2

**Value**
a matrix of test results

**Author(s)**
Jianhong Ou

**See Also**

- `singleSampleAnalyze`
- `singleGroupAnalyze`
- `limmaAnalyze`

**Examples**

```r
path <- file.path(find.package("InPAS"), "extdata")
load(file.path(path, "eset.MAQC.rda"))
tags <- colnames(eset$PDUI.log2)
res <- fisher.exact.test(eset, gp1=tags[1:2], gp2=tags[3:4])
```

get.regions.coverage  
calculate coverage for giving region

**Description**
calculate coverage for giving region

**Usage**

```r
get.regions.coverage(chr, utr3.regions.chr,
                      hugeData, coverage, phmm=FALSE)
```
getCov

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>chr</td>
<td>chromosome</td>
</tr>
<tr>
<td>utr3.regions.chr</td>
<td>the GRanges of region to be extracted</td>
</tr>
<tr>
<td>hugeData</td>
<td>is it a huge dataset?</td>
</tr>
<tr>
<td>coverage</td>
<td>output of coverageFromBedGraph</td>
</tr>
<tr>
<td>phmm</td>
<td>prepare data for singleSample analysis?</td>
</tr>
</tbody>
</table>

Value

GRanges with coverage data

Author(s)

Jianhong Ou

Description

extract coverage from bedgraph file

Usage

getcov(bedgraph, genome, seqLen)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bedgraph</td>
<td>bedGraph file names</td>
</tr>
<tr>
<td>genome</td>
<td>an object BSgenome</td>
</tr>
<tr>
<td>seqLen</td>
<td>lengths of each chromosome</td>
</tr>
</tbody>
</table>

Value

a Rle object for a sample coverage

Author(s)

Jianhong Ou

See Also

coverageFromBedGraph
getUTR3eSet

prepare dataset for test

Description

Generate a UTR3eSet object with PDUI information for statistic test

Usage

getUTR3eSet(CPsites, coverage, genome, utr3,
      normalize=c("none", "quantiles", "quantiles.robust",
                  "mean", "median"),
      ..., BPPARAM=NULL, singleSample=FALSE)

Arguments

CPsites     outputs of CPsites
coverage    coverage for each sample, outputs of coverageFromBedGraph
genome      an object of BSgenome
utr3        output of utr3Annotation
normalize   normalization method
...          parameter can be passed into normalize.quantiles.robust
BPPARAM     An optional BiocParallelParam instance determining the parallel back-end to
            be used during evaluation, or a list of BiocParallelParam instances, to be applied
            in sequence for nested calls to bplapply.
singleSample prepare data for singleSample analysis? default is FALSE

Value

An object of UTR3eSet which contains following elements:
usage: an GRanges object with CP sites info.
PDUI: a matrix of PDUI
PDUI.log2: log2 transformed PDUI matrix
short: a matrix of usage of short form
long: a matrix of usage of long form
if singleSample is TRUE, one more element, signals, will be included.

Author(s)

Jianhong Ou
**getUTR3region**

*Examples*

```r
path <- file.path(find.package("InPAS"), "extdata")
load(file.path(path, "CPs.MAQC.rda"))
load(file.path(path, "coverage.MAQC.rda"))
library(BSgenome.Hsapiens.UCSC.hg19)
data(utr3.hg19)
getUTR3eSet(CPsites=CPs,
            coverage=coverage,
            genome=BSgenome.Hsapiens.UCSC.hg19,
            utr3=utr3.hg19)
```

**Description**

extract long and short 3UTR region

**Usage**

```r
getUTR3region(.grs)
```

**Arguments**

- `.grs` output of CPsites

**Value**

GRanges with short form and long form

**Author(s)**

Jianhong Ou

---

**inPAS**

*do estimation of alternative polyadenylation and cleavage site in one step*

**Description**

do estimation of alternative polyadenylation and cleavage site in one step
Usage

```r
inPAS(bedgraphs, genome, utr3, txdb=NA,
    tags, hugeData=FALSE, ...,
    gp1, gp2,
    window_size=100,
    search_point_START=50, search_point_END=NA,
    cutStart=window_size, cutEnd=0,
    coverage_threshold=5, long_coverage_threshold=2,
    background=c("same_as_long_coverage_threshold",
                  "1K", "5K", "10K", "50K"),
    adjust_distal_polyA_end=TRUE,
    PolyA_PWM=NA, classifier=NA, classifier_cutoff=.8,
    shift_range=window_size,
    method=c("limma", "fisher.exact",
              "singleSample", "singleGroup"),
    normalize=c("none", "quantiles", "quantiles.robust",
                "mean", "median"),
    design, contrast.matrix, coef=1,
    P.Value_cutoff=0.05,
    adj.P.Val_cutoff=0.05,
    dPDUI_cutoff=0.3,
    PDUI_logFC_cutoff=0.59,
    BPPARAM=NULL)
```

Arguments

- `bedgraphs`: The file names of bedgraphs generated by bedtools. eg: bedtools genomecov -bg -split -ibam $bam -g mm10.size.txt > $bedgraph
- `genome`: an object of BSgenome
- `utr3`: output of utr3Annotation
- `txdb`: an object of TxDb
- `tags`: the names for each input bedgraphs
- `hugeData`: is this dataset consume too much memory? if it is TRUE, the coverage will be saved into tempfiles.
- `...`: parameters can be passed into tempfile. This is useful when you submit huge dataset to cluster.
- `gp1`: tag names involved in group 1
- `gp2`: tag names involved in group 2
- `window_size`: window size for noval distal position searching and adjusted polyA searching, default: 100
- `search_point_START`: start point for searching
- `search_point_END`: end point for searching
inPAS

**cutStart**  how many nucleotides should be removed from the start before search. 0.1 means 10 percent.

**cutEnd**  how many nucleotides should be removed from the end before search, 0.1 means 10 percent.

**coverage_threshold**  cutoff threshold for coverage in the region of short form

**long_coverage_threshold**  cutoff threshold for coverage in the region of long form

**background**  the range for calculating cutoff threshold of local background

**adjust_distal_polyA_end**  If true, adjust distal polyA end by `cleanUpdTSeq`

**PolyA_PWM**  Position Weight Matrix of polyA

**classifier**  An object of class "PASclassifier"

**classifier_cutoff**  This is the cutoff used to assign whether a putative pA is true or false. This can be any floating point number between 0 and 1. For example, classifier_cutoff = 0.5 will assign an putative pA site with prob.1 > 0.5 to the True class (1), and any putative pA site with prob.1 <= 0.5 as False (0).

**shift_range**  the shift range for polyA site searching

**method**  test method. see `singleSampleAnalyze`, `singleGroupAnalyze`, `fisher.exact.test`, `limmaAnalyze`

**normalize**  normalization method

**design**  the design matrix of the experiment, with rows corresponding to arrays and columns to coefficients to be estimated. Defaults to the unit vector meaning that the arrays are treated as replicates. see `model.matrix`

**contrast.matrix**  numeric matrix with rows corresponding to coefficients in fit and columns containing contrasts. May be a vector if there is only one contrast. see `makeContrasts`

**coef**  column number or column name specifying which coefficient or contrast of the linear model is of interest. see more `topTable`. default value: 1

**P.Value_cutoff**  cutoff of P value

**adj.P.Val_cutoff**  cutoff value for adjusted p.value

**dPDUI_cutoff**  cutoff value for differential PAS(polyadenylation signal) usage index

**PDUI_logFC_cutoff**  cutoff value for log2 fold change of PAS(polyadenylation signal) usage index

**BPPARAM**  An optional `BiocParallelParam` instance determining the parallel back-end to be used during evaluation, or a list of BiocParallelParam instances, to be applied in sequence for nested calls to bplapply.

**Value**

return an object of GRanges

**Author(s)**

Jianhong Ou
Examples

if(interactive()){
  library(BSgenome.Mus musculus.UCSC.mm10)
  library(TxDb.Mus musculus.UCSC.mm10.knownGene)

  path <- file.path(find.package("InPAS"), "extdata")
  bedgraphs <- file.path(path, "Baf3.extract.bedgraph")
  data(utr3.mm10)
  res <- inPAS(bedgraphs=bedgraphs, tags=c("Baf3"),
               genome=BSgenome.Mus musculus.UCSC.mm10,
               utr3=utr3.mm10, gp1="Baf3", gp2=NULL,
               txdb=TxDb.Mus musculus.UCSC.mm10.knownGene,
               search_point_START=200,
               short_coverage_threshold=15,
               long_coverage_threshold=3,
               cutStart=0, cutEnd=.2,
               hugeData=FALSE)
  res
}

lastCDSusage

extract coverage of last CDS exon region

Description

extract coverage of last CDS exon region

Usage

lastCDSusage(CDS, coverage, hugeData, BPPARAM=NULL, phmm=FALSE)

Arguments

  CDS          GRanges object of CDS
  coverage     output of coverageFromBedGraph
  hugeData     is it a huge dataset?
  BPPARAM      An optional BiocParallelParam instance determining the parallel back-end to
                be used during evaluation, or a list of BiocParallelParam instances, to be applied
                in sequence for nested calls to bplapply.
  phmm         prepare data for singleSample analysis?

Value

the average coverage of last CDS for each transcript

Author(s)

Jianhong Ou
limmaAnalyze

Description

use limma to analyze the PDUI

Usage

limmaAnalyze(UTR3eset, design, contrast.matrix, coef=1, robust=FALSE, ...)

Arguments

- **UTR3eset**: an UTR3eSet object
- **design**: the design matrix of the experiment, with rows corresponding to arrays and columns to coefficients to be estimated. Defaults to the unit vector meaning that the arrays are treated as replicates. see model.matrix
- **contrast.matrix**: numeric matrix with rows corresponding to coefficients in fit and columns containing contrasts. May be a vector if there is only one contrast. see makeContrasts
- **coef**: column number or column name specifying which coefficient or contrast of the linear model is of interest. see more topTable. default value: 1
- **robust**: logical, should the estimation of the empirical Bayes prior parameters be robustified against outlier sample variances?
- **...**: other arguments are passed to lmFit.

Value

fit results of eBayes by limma. It is an object of class MArrayLM containing everything found in fit. see eBayes

Author(s)

Jianhong Ou

See Also

singleSampleAnalyze, singleGroupAnalyze, fisher.exact.test

Examples

```r
library(limma)
path <- file.path(find.package("InPAS"), "extdata")
load(file.path(path, "eset.MAQC.rda"))
tags <- colnames(eset$PDUI.log2)
g <- factor(gsub("\..*$", ",", tags))
design <- model.matrix(~-1+g)
colnames(design) <- c("Brain", "UHR")
contrast.matrix <- makeContrasts(contrasts="Brain-UHR", levels=design)
res <- limmaAnalyze(eset, design, contrast.matrix)
head(res)
```
optimalSegmentation calculate SSE

**Description**

calculate SSE values

**Usage**

```
optimalSegmentation(.ele, search_point_START, search_point_END, n = 1, savedID = NA)
```

**Arguments**

- `.ele` 3UTR coverage
- `search_point_START` start position to calculate
- `search_point_END` end position to calculate
- `n` the length of output
- `savedID` the proximal CPsites for noval distal events

**Value**

a list of SSE and idx

**Author(s)**

Jianhong Ou

---

PAscore calculate the CP score

**Description**

calculate the CP score by PWM

**Usage**

```
PAscore(seqname, pos, str, idx, PWM, genome, ups = 50, dws = 50)
```

**Arguments**

- `seqname` sequence names
- `pos` genomic positions
- `str` strands
- `idx` offset position
- `PWM` polyA position weight matrix
- `genome` an object of BSgenome
- `ups` upstream base
- `dws` downstream base
**Value**

idx list after filter

**Author(s)**

Jianhong Ou

**See Also**

PAscore2

---

**PAscore2**

*calculate the CP score*

**Description**

calculate CP score by cleanUpdTSeq

**Usage**

PAscore2(seqname, pos, str, idx, idx.gp, genome, classifier, classifier_cutoff)

**Arguments**

- `seqname`: sequence names
- `pos`: genomic positions
- `str`: strands
- `idx`: offset postion
- `idx.gp`: group number of the offset postion
- `genome`: an object of BSgenome
- `classifier`: a cleanUpdTSeq classifier
- `classifier_cutoff`: classifier cutoff value

**Value**

a data.frame

**Author(s)**

Jianhong Ou

**See Also**

PAscore
polishCPs  

**polish the searching results of CP sites**

**Description**
remove the multiple positions of CP sites for same 3UTRs and only keep the best CP sites for proximal and distal.

**Usage**
polishCPs(CPs)

**Arguments**

CPs  
output of searchProximalCPs or proximalAdj

**Value**

a matrix with columns: "fit_value", "Predicted_Proximal_APA", "Predicted_Distal_APA", "utr3start", "utr3end", "type"

**Author(s)**

Jianhong Ou

**See Also**

CPsite_estimation, searchProximalCPs, proximalAdj, proximalAdjByPWM, proximalAdjByCleanUpdTSeq, PAscore, PAscore2

---

prepare4GSEA  

**prepare the files for GSEA analysis**

**Description**
output the log2 transformed delta PDUI txt file and chip file for GSEA analysis

**Usage**

prepare4GSEA(eset, groupList, Preranked=TRUE, folder=".", rnkFilename="InPAS.rnk", chipFilename="InPAS.chip", dataFilename="dPDUI.txt", PhenFilename="group.cls")
proximalAdj

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>eset</td>
<td>a UTR3eSet object</td>
</tr>
<tr>
<td>groupList</td>
<td>group list of tag names</td>
</tr>
<tr>
<td>Preranked</td>
<td>logical value, out preranked or not</td>
</tr>
<tr>
<td>folder</td>
<td>output folder</td>
</tr>
<tr>
<td>rnkFilename</td>
<td>filename of preranked file</td>
</tr>
<tr>
<td>chipFilename</td>
<td>filename of chip</td>
</tr>
<tr>
<td>dataFilename</td>
<td>filename of dataset</td>
</tr>
<tr>
<td>PhenFilename</td>
<td>filename of Phenotype labels</td>
</tr>
</tbody>
</table>

Value

None

Author(s)

Jianhong Ou

Examples

```r
file <- system.file("extdata", "eset.MAQC.rda", package="InPAS")
load(file)
gp1=c("Brain.auto", "Brain.phiX")
gp2=c("UHR.auto", "UHR.phiX")
groupList <- list(Brain=gp1, UHR=gp2)
prepare4GSEA(eset, groupList=groupList, Preranked=FALSE)
```

proximalAdj     adjust the proximal CP sites

Description

adjust the proximal CP sites by PolyA PWM and cleanUpdTSeq

Usage

```r
proximalAdj(CPs, MINSIZE, PolyA_PWM, genome, classifier, classifier_cutoff,
            shift_range, search_point_START, step=1)
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPs</td>
<td>the outputs of searchProximalCPs</td>
</tr>
<tr>
<td>MINSIZE</td>
<td>min size for short from</td>
</tr>
<tr>
<td>PolyA_PWM</td>
<td>PolyA position weight metrix</td>
</tr>
<tr>
<td>genome</td>
<td>a BSgenome object</td>
</tr>
<tr>
<td>classifier</td>
<td>cleanUpdTSeq classifier</td>
</tr>
<tr>
<td>classifier_cutoff</td>
<td>cutoff value of the classifier</td>
</tr>
</tbody>
</table>
proximalAdjByCleanUpdTSeq

shift_range the searching range for the better CP sites
search_point_START just in case there is no better CP sites
step adjust step, default 1, means adjust by each base by cleanUpdTSeq.

Value

keep same as searchProximalCPs, which can be handled by polishCPs.

Author(s)

Jianhong Ou

See Also

searchProximalCPs, polishCPs, proximalAdjByPWM, proximalAdjByCleanUpdTSeq, PAscore, PAscore2

Description

adjust the proximal CP sites by cleanUpdTseq

Usage

proximalAdjByCleanUpdTSeq(idx.list, cov_diff.list, seqnames, starts, strands, genome, classifier, classifier_cutoff, shift_range, search_point_START, step=1)

Arguments

idx.list the offset of positions of CP sites
cov_diff.list the SSE values
seqnames sequence names
starts starts
strands strands
genoome a BSgenome object
classifier cleanUpdTSeq classifier
classifier_cutoff cutoff value of the classifier
shift_range the searching range for the better CP sites
search_point_START just in case there is no better CP sites
step adjust step, default 1, means adjust by each base by cleanUpdTSeq.
proximalAdjByPWM

Details
the step for calculating is 10, can not do every base base it is really very slow.

Value
the offset of positions of CP sites after filter

Author(s)
Jianhong Ou

See Also
proximalAdjByPWM, proximalAdj, PAscore2

---

proximalAdjByPWM adjust the proximal CP sites by PWM

Description
adjust the proximal CP sites by polyA Position Weight Metrix. It only need the PWM get match in upstream or downstream shift_range nr.

Usage
proximalAdjByPWM(idx, PolyA_PWM, seqnames, starts, strands, genome,
shift_range, search_point_START)

Arguments
idx the offset of positions of CP sites
PolyA_PWM polyA PWM
seqnames sequence names
starts start position in the genome
strands strands
genome an BSgenome object
shift_range the shift range of PWM hits
search_point_START Not use

Details
the hits is searched by matchPWM and the cutoff is 70%

Value
the offset of positions of CP sites after filter
searchDistalCPs

Author(s)
Jianhong Ou

See Also
proximalAdjByCleanUpdTSeq, proximalAdj, PAscore

removeUTR3__UTR3
remove the candidates LIKE UTR3___UTR3

Description
some of the results is from connected two UTR3. We want to remove them. However, the algorithm need to be improved.

Usage
removeUTR3__UTR3(x)

Arguments
x the distal 3UTR coverage

Value
the 3UTR coverage after removing the next 3UTR

Author(s)
Jianhong Ou

searchDistalCPs
search distal CP sites

Description
search distal CP sites

Usage
searchDistalCPs(chr.cov.merge, conn_next_utr3, curr_UTR, window_size, depth.weight, long_coverage_threshold, background, z2s)
searchProximalCPs

Arguments

chr.cov.merge coverage of current chromosome
conn_next_utr3 joint to next 3UTR or not (used for removeUTR3__UTR3)
curr_UTR GRanges of current 3UTR
window_size window size
depth.weight output of depthWeight
long_coverage_threshold cutoff value for coverage of long form 3UTR
background local background range
z2s cut off background scores. see zScoreThreshold

Value
a list

Author(s)
Jianhong Ou

See Also
distalAdj, PAscore2

description
search proximal CPsites

Usage

searchProximalCPs(CPs, curr_UTR, window_size,
MINSIZE, cutEnd,
search_point_START,
search_point_END,
two_way=FALSE)

Arguments

CPs output of searchDistalCPs or distalAdj
curr_UTR GRanges of current 3UTR
window_size window size
MINSIZE MINSIZE for short form
cutEnd how many nucleotides should be removed from the end before search, 0.1 means 10 percent.
search_point_START start point for searching
search_point_END end point for searching
two_way Search the proximal site from both direction or not.
seqLen

Value

a list

Author(s)

Jianhong Ou

See Also

proximalAdj, polishCPs, proximalAdjByPWM, proximalAdjByCleanUpdTSeq, PAscore, PAscore2

seqLen

get sequence lengths

Description

get sequence lengths from a BSgenome object

Usage

seqLen(genome)

Arguments

genome an object of BSgenome

Value

a numeric vector

Author(s)

Jianhong Ou

See Also

seqlengths
singleGroupAnalyze  
do analysis for single group samples

Description

do analysis for single group samples by anova test

Usage

singleGroupAnalyze(UTR3eset)

Arguments

UTR3eset    must be the output of getUTR3eSet

Value

a matrix of test results

Author(s)

Jianhong Ou

See Also

UTR3eset, getUTR3eSet

Examples

path <- file.path(find.package("InPAS"), "extdata")
load(file.path(path, "eset.MAQC.rda"))
res <- singleGroupAnalyze(eset)

---

description

do analysis for single sample by a hidden Markov model

Usage

singleSampleAnalyze(UTR3eset)

Arguments

UTR3eset    must be the output of getUTR3eSet

Details

the test will be performed by a two states hidden Markov model.
**Value**

a matrix of test results

**Author(s)**

Jianhong Ou

**See Also**

UTR3eSet, getUTR3eSet, depmix

**Examples**

```r
path <- file.path(find.package("InPAS"), "extdata")
load(file.path(path, "eset.MAQC.rda"))
res <- singleSampleAnalyze(eset)
```

---

**sortGR**

**sort GRanges**

**Description**

sort a GRanges by chromosome and start position

**Usage**

```r
sortGR(.ele)
```

**Arguments**

`.ele` an object of GRanges

**Value**

an sorted object of GRanges

**Author(s)**

Jianhong Ou
Description

do test for dPDU

Usage

testUsage(CPsites, coverage, genome, utr3, BPPARAM=NULL,
method=c("limma", "fisher.exact",
  "singleSample", "singleGroup"),
normalize=c("none", "quantiles", "quantiles.robust",
  "mean", "median"),
design, contrast.matrix, coef=1, robust=FALSE, ..., gp1, gp2)

Arguments

CPsites outputs of CPsites
coverage coverage for each sample, outputs of coverageFromBedGraph
genome an object of BSgenome
utr3 output of utr3Annotation
BPPARAM An optional BiocParallelParam instance determining the parallel back-end to be used during evaluation, or a list of BiocParallelParam instances, to be applied in sequence for nested calls to bplapply.
method test method. see singleSampleAnalyze, singleGroupAnalyze, fisher.exact.test, limmaAnalyze
normalize normalization method
design the design matrix of the experiment, with rows corresponding to arrays and columns to coefficients to be estimated. Defaults to the unit vector meaning that the arrays are treated as replicates. see model.matrix
contrast.matrix numeric matrix with rows corresponding to coefficients in fit and columns containing contrasts. May be a vector if there is only one contrast. see makeContrasts
coef column number or column name specifying which coefficient or contrast of the linear model is of interest. see more topTable. default value: 1
robust logical, should the estimation of the empirical Bayes prior parameters be robustified against outlier sample variances?
... other arguments are passed to lmFit.
gp1 tag names involved in group 1
gp2 tag names involved in group 2

Details

if method is "limma", design matrix and contrast is required. if method is "fisher.exact", gp1 and gp2 is required.
totalCoverage

Value

a list with test results. the output of test results is a matrix.

Author(s)

Jianhong Ou

See Also

singleSampleAnalyze, singleGroupAnalyze, fisher.exact.test, limmaAnalyze

Examples

library(limma)
path <- file.path(find.package("InPAS"), "extdata")
load(file.path(path, "CPs.MAQC.rda"))
load(file.path(path, "coverage.MAQC.rda"))
library(BSgenome.Hsapiens.UCSC.hg19)
data(utr3.hg19)
tags <- names(coverage)
g <- factor(gsub("\..*$", "", tags))
design <- model.matrix(~-1+g)
colnames(design) <- c("Brain", "UHR")
contrast.matrix<-makeContrasts(contrasts="Brain-UHR",levels=design)
res <- testUsage(CPsites=CPs,
 coverage=coverage,
 genome=BSgenome.Hsapiens.UCSC.hg19,
 utr3=utr3.hg19,
 method="limma",
 design=design,
 contrast.matrix=contrast.matrix)
trimSeqnames

Author(s)
Jianhong Ou

Description
only ^chr[0-9XY]+$ is OK.

Usage
trimSeqnames(genome)

Arguments

gene name an BSgenome object

Value
an character vector with trimmed seqnames

Author(s)
Jianhong Ou

usage4plot

Description
prepare coverage data and fitting data for plot

Usage
usage4plot(gr, coverage, proximalSites, genome, groupList)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>gr</td>
<td>an object of GRanges</td>
</tr>
<tr>
<td>coverage</td>
<td>coverage for each sample</td>
</tr>
<tr>
<td>proximalSites</td>
<td>proximal sites</td>
</tr>
<tr>
<td>genome</td>
<td>an object of BSgenome</td>
</tr>
<tr>
<td>groupList</td>
<td>the list of sample names</td>
</tr>
</tbody>
</table>
utr3.hg19

Value

Formal class ‘GRanges’ [package "GenomicRanges"] with metadata:

- dat: matrix, first column is the fit data, the other columns are coverage data for each sample
- offset: offset from the start of 3'UTR

Author(s)

Jianhong Ou

Examples

```r
library(BSgenome.Mmusculus.UCSC.mm10)
path <- file.path(find.package("InPAS"), "extdata")
bedgraphs <- c(file.path(path, "Baf3.extract.bedgraph"),
                file.path(path, "UM15.extract.bedgraph"))
coverage <- coverageFromBedGraph(bedgraphs, tags=c("Baf3", "UM15"),
                                   genome=Mmusculus, hugeData=FALSE)
gr <- GRanges("chr6", IRanges(128846245, 128850081), strand="-")
dat <- usage4plot(gr, coverage, proximalSites=128849148, Mmusculus)
data <- dat$dat[[1]]
op <- par(mfrow=c(3, 1))
plot(data[,1], type="l", xlab="", ylab="The fitted value")
abline(v=dat$offset)
plot(data[,2], type="l", xlab="", ylab="Baf3")
plot(data[,3], type="l", xlab="", ylab="UM15")
par(op)
```

utr3.hg19

3'UTR annotation for hg19 obtained from utr3Annotation

Description

3'UTR annotation obtained from utr3Annotation by TxDb.Hsapiens.UCSC.hg19.knownGene and org.Hs.eG.db

Usage

data(utr3.hg19)

Format

GRanges with slot start holding the start position of the 3'UTR, slot end holding the end position of the 3'UTR, slot names holding transcripts and gene names of 3'UTR, slot seqnames holding the chromosome location where the 3'UTR is located and slot strand for strand of 3'UTR. In addition, the following variables are included.

- feature: should be unknown or proximalCP_XXXXXXXX
- id: should be utr3 or next.exon.gap
- exon: exon id
- transcript: transcript id
- gene: entrez gene id
- symbol: gene symbol
Details

used in the examples Annotation data obtained by: library(TxDb.Hsapiens.UCSC.hg19.knownGene)
library(org.Hs.eg.db)
utr3Annotation(TxDb.Hsapiens.UCSC.hg19.knownGene, org.Hs.egSYMBOL)

Value

an object of GRanges.

Examples

data(utr3.hg19)
head(utr3.hg19)

utr3.mm10

3’UTR annotation for mm10 obtained from utr3Annotation

Description

3’UTR annotation obtained from utr3Annotation by TxDb.Mmusculus.UCSC.mm10.knownGene and org.Mm.eg.db

Usage

data(utr3.mm10)

Format

GRanges with slot start holding the start position of the 3’UTR, slot end holding the end position of the 3’UTR, slot names holding transcripts and gene names of 3’UTR, slot seqnames holding the chromosome location where the 3’UTR is located and slot strand for strand of 3’UTR. In addition, the following variables are included.

feature should be unknown or proximalCP_XXXXXXXX
id should be utr3 or next.exon.gap
exon exon id
transcript transcript id
gene entriz gene id
symbol gene symbol

Details

used in the examples Annotation data obtained by: library(TxDb.Mmusculus.UCSC.mm10.knownGene)
library(org.Mm.eg.db)
utr3Annotation(TxDb.Mmusculus.UCSC.mm10.knownGene, org.Mm.egSYMBOL)

Value

an object of GRanges.
utr3Annotation

**Examples**

data(utr3.mm10)
head(utr3.mm10)

utr3Annotation extract 3' UTR from TxDb object

**Description**

extract 3'UTR from a TxDb object. The 3'UTR is defined as the last 3'UTR fragment for each transcript and it will be cut if there is any overlaps with other exons.

**Usage**

utr3Annotation(txdb, orgDbSYMBOL, MAX_EXONS_GAP = 10000)

**Arguments**

txdb an object of TxDb
orgDbSYMBOL a string indicates org SYMBOL to entrez id map
MAX_EXONS_GAP maximal exon gap for distal CP site

**Value**

return an object of GRanges with 7 metadata columns: feature (utr3, next.exon.gap, CDS), annotatedProximalCP (unknown, proximalCP_<coordinate>), exon (<transcript id>_<index>), transcript, gene (entrez_id), symbol, truncated (logical).

**Author(s)**

Jianhong Ou

**Examples**

if(interactive()){
  library(TxDb.Mmusculus.UCSC.mm10.knownGene)

  library(org.Mm.eg.db)

  utr3Annotation(TxDb.Mmusculus.UCSC.mm10.knownGene, "org.Mm.egSYMBOL")
}

UTR3eSet-class

Class UTR3eSet

Description

An object of class UTR3eSet represents the results of 3UTR usage.

Objects from the Class

Objects can be created by calls of the form new("UTR3eSet", usage, PDUI, PDUI.log2, short, long, signals, testRes).

Slots

usage an GRanges object with CP sites info.
PDUI a matrix of PDUI.
PDUI.log2 log2 transformed PDUI matrix.
short a matrix of usage of short form.
long a matrix of usage of long form.
signals signals used for single sample.
testRes a matrix of test results of testUsage.

Methods

$, $<- Get or set the slot of UTR3eSet.
as("UTR3eSet", "ExpressionSet") Convert a UTR3eSet to an ExpressionSet.
as("UTR3eSet", "GRanges") Convert a UTR3eSet to an GRanges.

Author(s)

Jianhong Ou

UTR3TotalCoverage extract coverage of 3UTR for CP sites prediction

Description

extract 3UTR coverage from totalCov according and GRanges object utr3.

Usage

UTR3TotalCoverage(utr3, totalCov, gcCompensation = NA, mappabilityCompensation = NA, FFT = FALSE, fft.sm.power = 20)
Arguments

- **utr3**: an GRanges object. must be the output of `utr3Annotation`
- **totalCov**: total coverage of each sample. must be the output of `totalCoverage`
- **gcCompensation**: GC compensation vector. Not support yet.
- **mappabilityCompensation**: mappability compensation vector. Not support yet.
- **FFT**: Use FFT smooth or not.
- **fft.sm.power**: the cut-off frequency of FFT smooth.

Value

- a list. level 1: chromosome; level 2: each transcripts; level3: data matrix

Author(s)

Jianhong Ou

---

**UTR3usage**

*calculate the usage of long and short form of UTR3*

Description

calculate the usage of long and short form of UTR3 for the results of `CPsites`

Usage

```
UTR3usage(CPsites, coverage, hugeData, BPPARAM = NULL, phmm = FALSE)
```

Arguments

- **CPsites**: outputs of `CPsites`
- **coverage**: coverage for each sample, outputs of `coverageFromBedGraph`
- **hugeData**: is this dataset consume too much memory? if it is TRUE, the coverage will be saved into tempfiles.
- **BPPARAM**: An optional BiocParallelParam instance determining the parallel back-end to be used during evaluation, or a list of BiocParallelParam instances, to be applied in sequence for nested calls to bplapply.
- **phmm**: prepare data for singleSample analysis? default is FALSE

Value

- GRanges object

Author(s)

Jianhong Ou

See Also

`CPsites`
utr3UsageEstimation

estimation of 3’UTR usage for each region

Description
estimation of 3’UTR usage for short form and long form

Usage

utr3UsageEstimation(CPsites, coverage, genome, utr3,
gp1, gp2=NULL,
short_coverage_threshold = 10,
long_coverage_threshold = 2,
adjusted.P_val_cutoff = 0.05,
dPDUI_cutoff = 0.3,
PDUI_logFC_cutoff=0.59, BPPARAM=NULL)

Arguments

CPsites outputs of CPsites
coverage coverage for each sample, outputs of coverageFromBedGraph
genome an object of BSgenome
utr3 output of utr3Annotation
gp1 tag names involved in group 1
gp2 tag names involved in group 2
short_coverage_threshold cutoff threshold for coverage in thre region of short form
long_coverage_threshold cutoff threshold for coverage in thre region of long form
adjusted.P_val_cutoff cutoff value for adjusted p.value
dPDUI_cutoff cutoff value for differential PAS(polyadenylation signal) usage index
PDUI_logFC_cutoff cutoff value for log2 fold change of PAS(polyadenylation signal) usage index
BPPARAM An optional BiocParallelParam instance determining the parallel back-end to be used during evaluation, or a list of BiocParallelParam instances, to be applied in sequence for nested calls to bplapply.

Value
return an object of GRanges

Author(s)
Jianhong Ou
valley

get the local minimal square standard error (SSE)

Description

For a giving numeric vectors, calculate the top N local minimal square standard error. It will also include the saved ID if it is in the range of (ss, se)

Usage

valley(x, ss, se, n = 1, savedID = NA, filterByPval = TRUE)

Arguments

x       numeric vector
ss      start searching position
se      end searching position
n       the length of output. If n=-1, output all the local minimal SSE positions.
savedID saved positions
filterByPval logical. Filter the positions by p value or not.

Value

a numeric vector, position list.

Author(s)

Jianhong Ou
zScoreThreshold  

**Description**

calculate local background cutoff value based on z-score

**Usage**

zScoreThreshold(background, introns, totalCov, utr3, z = 2)

**Arguments**

- **background**: background range
- **introns**: GRanges of introns
- **totalCov**: total coverage of output of totalCoverage
- **utr3**: output of utr3Annotation
- **z**: z score cut off value

**Value**

- a numeric vector

**Author(s)**

Jianhong Ou
Index

*Topic classes
  UTR3eSet-class, 37

*Topic datasets
  utr3.hg19, 34
  utr3.mm10, 35

*Topic misc
  coverageFromBedGraph, 3
  coverageRate, 4
  covThreshold, 5
  CPsite_estimation, 8
  CPsites, 6
  depthWeight, 9
  distalAdj, 10
  filterRes, 10
  fisher.exact.test, 12
  get.regions.coverage, 12
  getCov, 13
  getUTR3eSet, 14
  getUTR3region, 15
  inPAS, 15
  lastCDSusage, 18
  limmaAnalyze, 19
  optimalSegmentation, 20
  PAscore, 20
  PAscore2, 21
  polishCPs, 22
  prepare4GSEA, 22
  proximalAdj, 23
  proximalAdjjByCleanUpdTSeq, 24
  proximalAdjjByPWM, 25
  removeUTR3__UTR3, 26
  searchDistalCPs, 26
  searchProximalCPs, 27
  seqLen, 28
  singleGroupAnalyze, 29
  singleSampleAnalyze, 29
  sortGR, 30
  testUsage, 31
  totalCoverage, 32
  trimSeqnames, 33
  usage4plot, 33
  utr3Annotation, 36
  UTR3TotalCoverage, 37
  UTR3usage, 38
  utr3UsageEstimation, 39
  valley, 40
  zScoreThrethold, 41

*Topic package
  InPAS-package, 2
  $, UTR3eSet-method (UTR3eSet-class), 37
  $<-, UTR3eSet-method (UTR3eSet-class), 37
  BiocParallelParam, 3, 7, 14, 17, 18, 31, 38, 39
  BSgenome, 5, 6, 9, 10, 13, 14, 16, 20, 21, 23–25, 28, 31–33, 39
  cleanUpdTSeq, 7, 17
  coverageFromBedGraph, 3, 4–6, 9, 13, 14, 31, 32, 38, 39
  coverageRate, 4
  covThreshold, 5
  CPsite_estimation, 6, 8, 22
  CPsites, 6, 9, 14, 31, 38, 39
  depmix, 30
  depthWeight, 9, 9, 27
  distalAdj, 10, 27
  eBayes, 19
  ExpressionSet, 37
  filterRes, 10
  fisher.exact.test, 12, 17, 19, 31, 32
  get.regions.coverage, 12
  getCov, 13
  getUTR3eSet, 12, 14, 29, 30
  getUTR3region, 15
  GRanges, 5, 37
  InPAS (InPAS-package), 2
  inPAS, 15
  InPAS-package, 2
  lastCDSusage, 18
  limmaAnalyze, 12, 17, 19, 31, 32
  makeContrasts, 17, 19, 31
matchPWM, 25
model.matrix, 17, 19, 31
normalize.quantiles.robust, 14
optimalSegmentation, 20
PASclassifier, 7, 17
PAscore, 9, 20, 21, 22, 24, 26, 28
PAscore2, 9, 10, 21, 22, 24, 25, 27, 28
polishCPs, 22, 24, 28
prepare4GSEA, 22
proximalAdj, 9, 22, 23, 25, 26, 28
proximalAdjByCleanUpdSeq, 9, 22, 24, 26, 28
proximalAdjByPWM, 9, 22, 24, 25, 28
removeUTR3__UTR3, 26, 27
searchDistalCPs, 10, 26, 27
searchProximalCPs, 9, 10, 22–24, 27
seqLen, 28
seqlengths, 28
singleGroupAnalyze, 12, 17, 19, 29, 31, 32
singleSampleAnalyze, 12, 17, 19, 29, 31, 32
sortGR, 30
testUsage, 10, 11, 31, 37
topTable, 17, 19, 31
totalCoverage, 32, 38, 41
trimSeqnames, 33
TxDb, 4, 5, 7, 16, 36
usage4plot, 33
utr3.hg19, 34
utr3.mm10, 35
utr3Annotation, 6, 14, 31, 36, 38, 39, 41
UTR3eSet, 14, 19, 23, 29, 30, 37
UTR3eSet (UTR3eSet-class), 37
UTR3eSet-class, 37
UTR3TotalCoverage, 37
UTR3usage, 38
utr3UsageEstimation, 39
valley, 40
zScoreThreshold, 8, 27, 41