Package 'APAlyzer'

April 15, 2020

Type Package

Title A toolkit for APA analysis using RNA-seq data

Version 1.0.0

Description Perform 3'UTR APA, Intronic APA and gene expression analysis using RNA-seq data.

biocViews Sequencing, RNASeq, DifferentialExpression, GeneExpression, GeneRegulation, Annotation, DataImport, Software

Imports GenomicRanges, GenomicFeatures, GenomicAlignments, DESeq, SummarizedExperiment, Rsubread, stats, methods

Suggests knitr, rmarkdown, BiocStyle, org.Mm.eg.db, AnnotationDbi, TBX20BamSubset, Rsamtools, ggplot2, testthat

URL https://github.com/RJWANGbioinfo/APAlyzer/

BugReports https://github.com/RJWANGbioinfo/APAlyzer/issues

VignetteBuilder knitr

License LGPL-3

Encoding UTF-8

git_url https://git.bioconductor.org/packages/APAlyzer

git_branch RELEASE_3_10

git_last_commit 2874b52

git_last_commit_date 2019-10-29

Date/Publication 2020-04-14

Author Ruijia Wang [cre, aut] (<https://orcid.org/0000-0002-4211-5207>), Bin Tian [aut]

Maintainer Ruijia Wang <rjwang.bioinfo@gmail.com>

R topics documented:

APAdiff *APAdiff, calculate delta relative expression (RED) and statistics significance between two sample groups*

Description

Calculate delta relative expression (RED) and statistics significance between two sample groups.

Usage

```
APAdiff(sampleTable,mutiraw, conKET='NT',
trtKEY='KD',PAS='3UTR',CUTreads=0)
```

Arguments

sampleTable	a dataframe of sample table containing 8 colmuns for Intronic PASs: 'samplename','condition'
mutiraw	a dataframe output obtained using either PASEXP_3UTR or PASEXP_IPA
conKET	the name of control in the sampletable, default is 'NT'
trtKEY	the name of control in the sampletable, default is 'KD'
PAS	type of PAS analyzed, either '3UTR' or 'IPA', default is '3UTR'
CUTreads	reads cutoff used for the analysis, default is 0

Value

The function APAdiff return a dataframe containning RED, pvalue and regulation pattern (UP, DN or NC) for either each gene (3'UTR APA) or each PAS (IPA).

Author(s)

Ruijia Wang

Examples

```
library("TBX20BamSubset")
library("Rsamtools")
flsall = getBamFileList()
extpath = system.file("extdata",
"mm9_TBX20.APAout.RData", package="APAlyzer")
load(extpath)
sampleTable1 = data.frame(samplename = c(names(flsall)),
    condition = c(rep("NT",3),rep("KD",3)))
sampleTable2 = data.frame(samplename = c("SRR316184","SRR316187"),
    condition = c("NT","KD"))
## Analysis 3'UTR APA between KD and NT group using muti-replicates
test_3UTRmuti=APAdiff(sampleTable1,DFUTRraw,
conKET='NT',trtKEY='KD',PAS='3UTR',CUTreads=0)

## Analysis 3'UTR APA between KD and NT group without replicates
test_3UTRsing=APAdiff(sampleTable2,DFUTRraw,
conKET='NT',trtKEY='KD',PAS='3UTR',CUTreads=0)
```

```
## Analysis IPA between KD and NT group
test_IPAmuti=APAdiff(sampleTable1,IPA_OUTraw,
conKET='NT',trtKEY='KD',PAS='IPA',CUTreads=0)

## Analysis IPA between KD and NT group without replicates
test_IPAsing=APAdiff(sampleTable2,IPA_OUTraw,
conKET='NT',trtKEY='KD',PAS='IPA',CUTreads=0)
```

GENEXP_CDS *GENEXP_CDS, count reads mapped to CDS regions and calculate TPM for coding gene*

Description

Map reads to CDS regions and calculate TPM for each gene.

Usage

```
GENEXP_CDS(CDSbygene, flS, Strandtype="NONE")
```

Arguments

CDSbygene	a genomic ranges of CDS regions for each coding gene
flS	bamfile lists containing the file and path of bam files
Strandtype	strand type of the bam file; "forward" is forwand sequencing, "invert" is reverse sequencing and "NONE" is non-strand specific, Default is "NONE".

Value

The function GENEXP_CDS() return a dataframe containing reads count, TPM for each gene

Author(s)

Ruijia Wang

Examples

```
## count reads mapped to CDS regions and calculate TPM for each gene
## using forward sequencing
    library("TBX20BamSubset")
    library("Rsamtools")
    library("GenomicAlignments")
    library("GenomicFeatures")
    library("org.Mm.eg.db")
    flsall = getBamFileList()
    extpath = system.file("extdata", "mm9.chr19.refGene.R.DB", package="APAlyzer")
    txdb = loadDb(extpath, packageName='GenomicFeatures')
    IDDB = org.Mm.eg.db
    CDSdbraw = REFCDS(txdb,IDDB)
    DFGENEraw = GENEXP_CDS(CDSdbraw, flsall, Strandtype="forward")
```

PASEXP_3UTR *PASEXP_3UTR, calculate relative expression of aUTR and cUTR regions*

Description

Map reads to 3'UTR APA regions and calculate relative expression of aUTR and cUTR regions.

Usage

```
PASEXP_3UTR(UTRdb, flS, Strandtype="NONE")
```

Arguments

UTRdb	a genomic ranges of aUTR(pPAS to dPAS) and cUTR(cdsend to pPAS) regions for each gene
flS	bamfile lists containing the file and path of bam files
Strandtype	strand type of the bam file; "forward" is forwand sequencing, "invert" is reverse sequencing and "NONE" is non-strand specific, Default is "NONE".

Value

The function PASEXP_3UTR() return a dataframe containning reads count, RPKM and relative expression of aUTR and cUTR for each gene

Author(s)

Ruijia Wang

Examples

```
## count reads mapped to 3'UTR APA regions and
## calculate relative expression of aUTR and cUTR regions
## using forward sequencing
    library("TBX20BamSubset")
    library("Rsamtools")
    library("GenomicAlignments")
    flsall = getBamFileList()
    extpath = system.file("extdata", "mm9_REF.RData", package="APAlyzer")
    load(extpath)
    refUTRraw = refUTRraw[which(refUTRraw$Chrom=='chr19'),]
    UTRdbraw = REF3UTR(refUTRraw)
    DFUTRraw = PASEXP_3UTR(UTRdbraw, flsall, Strandtype="forward")
```

PASEXP_IPA *PASEXP_IPA, calculate relative expression of IPA regions*

Description

Map reads to IPA regions and calculte relative expression of aUTR and cUTR regions.

Usage

```
PASEXP_IPA(dfIPAraw, dfLEraw, flS, Strandtype="NONE", nts=1)
```

Arguments

dfIPAraw	a dataframe containing 8 colmuns for Intronic PASs: 'PASid', 'gene_symbol', 'Chrom', 'Strand', 'Pos', 'upstreamSS', 'downstreamSS'. 'upstreamSS' means closest 5'/3' splice site to IPA, 'downstreamSS' means closest 3' splice site
dfLEraw	a dataframe containing 5 colmuns for 3' least exon: 'gene_symbol', 'Chrom', 'Strand', 'LEstart', 'TES'. 'LEstart' means the start position of last 3' exon.
flS	bamfile lists containing the file and path of bam files
Strandtype	strand type of the bam file; "forward" is forwand sequencing, "invert" is reverse sequencing and "NONE" is non-strand specific, Default is "NONE".
nts	number of threads used for computing, parameter used by featureCounts, nthread option, Default is 1

Value

The function PASEXP_IPA() return a dataframe containning reads count, RPKM and relative expression of aUTR and cUTR for each gene

Author(s)

Ruijia Wang

Examples

```
## count reads mapped to IPA regions and
## calculte relative expression of aUTR and cUTR regions
## using forward sequencing
    library("TBX20BamSubset")
    library("Rsamtools")
    library("GenomicAlignments")
    flsall = getBamFileList()
    extpath = system.file("extdata", "mm9_REF.RData", package="APAlyzer")
    load(extpath)
    IPA_OUTraw=PASEXP_IPA(dfIPA, dfLE, flsall, Strandtype="forward", nts=1)
```

REF3UTR *REF3UTR, build reference regions for 3'UTR PASs*

Description

Build 3'UTR PAS Reference for distal and proximal PAS.

Usage

```
REF3UTR(refUTR)
```

Arguments

refUTR	a dataframe containing 6 colmuns for 3'UTR PASs: 'gene_symbol', 'Chrom', 'Strand', 'Proximal', 'Distal', 'cdsend'

Value

The function REF3UTR() returns a genomic ranges of aUTR(pPAS to dPAS) and cUTR(cdsend to pPAS) regions for each gene

Author(s)

Ruijia Wang

Examples

```
## build Reference ranges for 3'UTR PASs in human
    extpath = system.file("extdata", "mm9_REF.RData", package="APAlyzer")
    load(extpath)
    refUTRraw=refUTRraw[which(refUTRraw$Chrom=='chr19'),]
    UTRdbraw=REF3UTR(refUTRraw)
```

REFCDS *REFCDS, build reference regions for CDS of protein coding genes*

Description

Build CDS reference for protein coding genes.

Usage

```
REFCDS(txdb,IDDB)
```

Arguments

txdb	a TranscriptDb generate using GenomicFeatures
IDDB	Genome annotation of the corresponding species, e.g., "org.Hs.eg.db"

Value

The function `REFCDS()` returns a genomic ranges of CDS regions for each coding gene

Author(s)

Ruijia Wang

Examples

```
## build Reference ranges for CDS in human coding genes
    library("GenomicFeatures")
    library("org.Mm.eg.db")
    extpath = system.file("extdata", "mm9.chr19.refGene.R.DB", package="APAlyzer")
    txdb = loadDb(extpath, packageName='GenomicFeatures')
    IDDB = org.Mm.eg.db
    CDSdbraw = REFCDS(txdb,IDDB)
```

Index