Package 'spliceR'

April 10, 2015

Version 1.8.0

Date 2014/04/29

Title Classification of alternative splicing and prediction of coding potential from RNA-seq data.

Author Johannes Waage <johannes.waage@gmail.com>, Kristoffer Vitting-Seerup <k.vitting.seerup@gmail.com>

Maintainer Johannes Waage <johannes.waage@gmail.com>, Kristoffer Vitting-Seerup <k.vitting.seerup@gmail.com>

Imports GenomicRanges, IRanges

Depends R (>= 2.15.0), methods, cummeRbund, rtracklayer, VennDiagram, RColorBrewer, plyr

Suggests BSgenome.Hsapiens.UCSC.hg19, BSgenome

Description An R package for classification of alternative splicing and prediction of coding potential from RNA-seq data.

License GPL (>=2)

biocViews GeneExpression, Transcription, AlternativeSplicing, DifferentialExpression, DifferentialSplicing, Sequencing, Visualization

R topics documented:

---

annotatePTC   *PTC and NMD-sensitivity detection from assembled RNA-seq data*

Description

PTC and NMD-sensitivity detection from assembled RNA-seq data.

Usage

```
annotatePTC(transcriptData, cds, genomeObject, PTCDistance=50)
```

Arguments

transcriptData	A SpliceRList object, containing transcript and exon information.
cds	A CDSSet object, containing CDS information.
genomeObject	A BSgenome object, containing sequence for the relevant genome. Contained in BSGenome objects, downloadable from BioConductor.
PTCDistance	A numeric giving the premature termination codon-distance: The minimum distance from a STOP to the final exon-exon junction, for a transcript to be marked as NMD-sensitive.

Details

annotatePTC retrieves sequence data for all exons given in transcriptData, uses the CDS-information in cds to scan for the most upstream reading frame, and translates the mRNA, storing information about the first codon in relation to distance from TTS, distance to the final exon-exon junction, etc. If the STOP distance to the final exon-exon junction is larger than the threshold given in PTCDistance (and the STOP does not fall in the last exon), the STOP is considered premature and the transcript is marked as NMD (nonsense mediated decay) sensitive. For a review of the PTC and NMD mechanism, see Weischenfeldt et al. 2012.

Value

A SpliceRList, with the transcript_features object containing additional columns:

spliceR.cdsPosGenomic, the genomic position of the used START codon.

spliceR.stopPosGenomic, the genomic position of the identified STOP codon,

spliceR.ExonWithStart, the exon which the used START codon falls within,

spliceR.ExonWithStop, the exon which the STOP codon falls within.

spliceR.cdsPosTranscipt, the position relative to transcript start where the used START codon is (measured in nucleotides).

spliceR.stopPosTranscipt, the position relative to transcript start where the found STOP codon is (measured in nucleotides).

spliceR.stopDistance, the distance from the found STOP codon to the last exon-exon junction, realative to transcript start.

spliceR.junctionIndex, the exon number in which the found STOP codon falls when compared to the last exon-exon junction, where 0 is the last exon of the transcript, -1 is the second-last, etc NA, if annotatePTC was not able to find a ORF.

spliceR.PTC, a boolean, indicating whether the transcript is (theoretically) susceptible to nonsense mediated decay. annotatePTC sets this value to TRUE if the stop codon falls if any exon other than the last, and the distance to the final downstream exon-exon junction is larger than PTCDistance (default 50 nt).

Author(s)

Kristoffer Vitting-Seerup, Johannes Waage

References

Vitting-Seerup K , Porse BT, Sandelin A, Waage J. (2014) spliceR: an R package for classification of alternative splicing and prediction of coding potential from RNA-seq data. BMC Bioinformatics 15:81.

Examples

```
## Not run:

#Rebuild cummeRbunds internal dataset
cuffDB <- readCufflinks(dir=system.file("extdata", package="cummeRbund"), gtf=system.file("extdata/chr1_snippet.

#Generate SpliceRList from cufflinks data
cuffDB_spliceR <- prepareCuff(cuffDB)

# Require BSgenome object, containing genomic sequence
require("BSgenome.Hsapiens.UCSC.hg19",character.only = TRUE)

#Get CDS from UCSC
ucscCDS <- getCDS(selectedGenome="hg19", repoName="UCSC")

#Annotate with PTCs
cuffDB_spliceR_PTC <- annotatePTC(cuffDB_spliceR, cds=ucscCDS, Hsapiens, PTCDistance=50)


## End(Not run)
```

CDSSet   *Container for coding sequence (CDS) annotation information*

Description

A container for coding sequence annotation information.

Usage

```
CDSSet(cds)
```

Arguments

cds   A data.frame object contaning CDS annotation. See details for required columns.

Details

This object can be generated automatically from [getCDS](#), or can be be generated manually by creating a new CDSSet from a data.frame with the following columns:

chrom, the chromosome name (NB: chromosome names must match when running [annotatePTC](#)). strand, the strand, cdsStart, the genomic start of the coding sequence (beware of 0/1-frame issuses), and cdsEnd, the genomic end of the coding sequence (beware of 0/1-frame issuses).

The CDSset object is required by [annotatePTC](#) for translation of transcripts and premature termination codon annotation.

For an example, see [annotatePTC](#).

Value

A CDSSet object.

Author(s)

Kristoffer Vitting-Seerup, Johannes Waage

References

Vitting-Seerup K , Porse BT, Sandelin A, Waage J. (2014) spliceR: an R package for classification of alternative splicing and prediction of coding potential from RNA-seq data. BMC Bioinformatics 15:81.

---

conditions	*Returns sample condictions of an SpliceRList or an CuffSet object*

Description

Returns samples/condictions of an SpliceRList or an CuffSet object.

Usage

```
conditions(object)
```

Arguments

object   a `SpliceRList` object or a `CuffSet` object.

Details

This helper function returns the "conditions" slot of a `SpliceRList`, or the "sample" slot of a `CuffSet`.

Value

A `character` vector, giving the samples/conditions.

Author(s)

Kristoffer Vitting-Seerup, Johannes Waage

References

Vitting-Seerup K , Porse BT, Sandelin A, Waage J. (2014) spliceR: an R package for classification of alternative splicing and prediction of coding potential from RNA-seq data. BMC Bioinformatics 15:81.

Examples

```
#Load cufflinks example data
cuffDB <- prepareCuffExample()

conditions(cuffDB)

#Generate SpliceRList from cufflinks data
cuffDB_spliceR <- prepareCuff(cuffDB)

conditions(cuffDB_spliceR)
```

---

dim   *Retrieve the Dimensions of a SpliceRList*

Description

Retrieve the number of (transcripts) contained in `SpliceRList`.

Usage

```
## S3 method for class SpliceRList
dim(x)
## S3 method for class SpliceRList
length(x)
```

Arguments

x   an object of class `SpliceRList`.

Details

As documented in `SpliceRList`, a SpliceRList contains two objects; a 'transcript_features' GRanges
object, containing transcript information, and a 'exon_features' object, containing exon informa-
tion. Dim and length currently only gives information about the number of transcripts in a SpliceRList
object, i.e. the length() of the 'transcript_features' GRanges object.

Value

Numeric vector of length 1, indicating the number of trancript comparisons in the SpliceRList.

See Also

[dim](#) in the base package.

---

generateGTF   *Generate GTF files for transcript visualization in genome browsers*

Description

Generate GTF files for transcript visualization in genome browsers.

Usage

```
generateGTF(transcriptData, filters=NULL, expressionCutoff=0,scoreMethod="local", filePrefix="splice
```

Arguments

transcriptData   A SpliceRList object, created manually from transcript and exon information, or procuced by prepareCuff from CuffLinks data, and optionally processed by spliceR and/or annotatePTC.

filters   Vector, giving the filters that should be applied - any combinations of 'geneOK', 'expressedGenes', 'sigGenes', 'isoOK', 'expressedIso', 'isoClass' and/or 'sigIso'. Works only for data from cufflinks.

expressionCutoff
    Numeric, giving the expression threshold (often in FPKM) used for the 'expressedGenes' and 'expressedIso' filter. Default value is 0.

scoreMethod   Character, either of 'local" of 'global', indicating whether to score isoform expression values for GTF color coding based on expression of the isoform in relation to the sample (global) or the gene (local).

filePrefix   Output file name prefix, including path.

shortDescription
    A short description for the GTF track.

longDescription
    A long description for the GTF track.

useProgressBar   Boolean, indicating whether to use progressbars. For compatibility. Default = TRUE.

Details

generateGTF generates GTF files, one for each sample/condition type, and writes these to disk in the current working directory. If the data was generated using cufflinks and the "source_id" slot of the transcriptData is set to "cufflinks", a number of filters can be applied (see spliceR for a full description of filters). Transcripts will be colored on a grayscale according to the scoreMethod parameter; for "local", the isoform most expressed for a given gene symbol will be darkest; for "global", the color coding will be relative to each transcripts expression across the sample.

Author(s)

Kristoffer Vitting-Seerup, Johannes Waage

References

Vitting-Seerup K, Porse BT, Sandelin A, Waage JE. (2013) spliceR: An R package for classification of alternative splicing and prediction of coding potential from RNA-seq data. *PeerJ PrePrints* 1:e80v1

Examples

```
#Load cufflinks example data
cuffDB <- prepareCuffExample()

#Generate SpliceRList from cufflinks data
cuffDB_spliceR <- prepareCuff(cuffDB)
```

```
#Reduce dataset size for fast example runtime
cuffDB_spliceR[[1]] <- cuffDB_spliceR[[1]][1:500]

#Run spliceR
mySpliceRList <- spliceR(cuffDB_spliceR, compareTo=preTranscript, filters=c(expressedGenes,geneOK, isoOK, expres

#Export to GTF
generateGTF(mySpliceRList, filters=c("geneOK", "isoOK", "expressedGenes", "expressedIso"), scoreMethod="local")
```

getCDS   *Retrieve CDS information from UCSC*

Description

Retrieve CDS information from a selected repository from UCSC genome browser repositories.

Usage

```
getCDS(selectedGenome, repoName)
```

Arguments

selectedGenome   A character, giving the genome. Currently supported are "hg19" and "mm9".

repoName   A character, giving the gene model repository. Currently supported are "ensemble", "UCSC" (knownGene), and "refseq".

Details

For other genomes and/or gene model repositories, please construct a CDSSet directly (see [CDSSet]).

For a full example of how to use getCDS in a workflow, please see [annotatePTC].

Value

A CDSSet containing the annotated CDSs. For a description of the dataframe, see [CDSSet].

Author(s)

Kristoffer Vitting-Seerup, Johannes Waage

References

Vitting-Seerup K , Porse BT, Sandelin A, Waage J. (2014) spliceR: an R package for classification of alternative splicing and prediction of coding potential from RNA-seq data. BMC Bioinformatics 15:81.

Examples

```
## Not run:
mm9UCSC <- getCDS("mm9", "UCSC")

## End(Not run)
```

---

prepareCuff   *Prepare assembled RNA-seq data from Cufflinks for spliceR*

Description

Prepare assembled RNA-seq data from Cufflinks for spliceR.

Usage

```
prepareCuff(cuffDB, fixCufflinksAnnotationProblem=TRUE,removeNonChanonicalChr=TRUE)
```

Arguments

cuffDB   a cuffDB object, produced by cummeRbund. This object must have been gener-
                    ated with cummeRbund, using the gtf parameter (see example), for spliceR to
                    extract transcript model and exon information.

fixCufflinksAnnotationProblem
                    Fixes problems with Cufflinks gene symbol annotation. Please see the vignette
                    for additional information.

removeNonChanonicalChr
                    Removes non-canonical chromosome names.

Details

NB: prepareCuff is optimized to work with the cummeRbund vs v2.7.2 or later. Please check
your version, and update if appropriate. Use prepareCuff to prepare a cummeRbund/Cufflinks
DB object for use by spliceR (see example). Often, it's appropriate to prefilter cufflinks data after
running prepareCuff with preSpliceRFilter to reduce overhead on downstream analyses.

Value

A SpliceRList containing a transcript_features GRanges object with the following additional
metacolumns extracted from the cufflinks DB:

spliceR.isoform_id
                    Cufflinks unique isoform id

spliceR.sample_1
                    Sample 1 identifier

spliceR.sample_2
                    Sample 2 identifier

`spliceR.gene_id`
> Cufflinks unique gene id

`spliceR.CDS_id`   Cufflinks unique CDS id

`spliceR.gene_short_name`
> Cufflinks unique short gene name

`spliceR.TSS_group_id`
> Cufflinks unique TSS id

`spliceR.class_code`
> Cufflinks class code (see cufflinks documentation)

`spliceR.nearest_ref_id`
> Nearest reference id

`spliceR.length`   Length of the transcript

`spliceR.gene_status`
> Cufflinks gene quantification status

`spliceR.gene_value_1`
> Gene FPKM value for sample 1

`spliceR.gene_value_2`
> Gene FPKM value for sample 2

`spliceR.gene_log2_fold_change`
> Log2 fold change of gene expression (sample2 / sample1)

`spliceR.gene_p_value`
> P-value for differential testing of difference of gene expression between samples

`spliceR.gene_q_value`
> Adjusted p-value for differential testing of difference of gene expression between samples

`spliceR.gene_significant`
> Yes/no; yes if difference of gene expression is significant

`spliceR.iso_status`
> Cufflinks isoform quantification status

`spliceR.iso_value_1`
> Isoform FPKM value for sample 1

`spliceR.iso_value_2`
> Isoform FPKM value for sample 2

`spliceR.iso_log2_fold_change`
> Log2 fold change of isoform expression (sample2 / sample1)

`spliceR.iso_p_value`
> P-value for differential testing of difference of isoform expression between samples

`spliceR.iso_q_value`
> P-value for differential testing of difference of isoform expression between samples

`spliceR.iso_significant`
> Yes/no; yes if difference of isoform expression is significant

and a `exon_features` GRanges object containing exon model information.

Author(s)

Kristoffer Vitting-Seerup, Johannes Waage

References

Vitting-Seerup K , Porse BT, Sandelin A, Waage J. (2014) spliceR: an R package for classification of alternative splicing and prediction of coding potential from RNA-seq data. BMC Bioinformatics 15:81.

Examples

```
#Load cufflinks example data
cuffDB <- prepareCuffExample()

#Generate SpliceRList from cufflinks data
cuffDB_spliceR <- prepareCuff(cuffDB)
```

---

prepareCuffExample   *Prepare the Cufflinks example data*

Description

Prepare the Cufflinks example data set.

Usage

```
prepareCuffExample()
```

Details

This helper function prepares the Cufflinks example dataset, including the example GTF-file.

Value

A CuffSet object.

Author(s)

Kristoffer Vitting-Seerup, Johannes Waage

References

Vitting-Seerup K , Porse BT, Sandelin A, Waage J. (2014) spliceR: an R package for classification of alternative splicing and prediction of coding potential from RNA-seq data. BMC Bioinformatics 15:81.

Examples

```
#Load cufflinks example data
cuffDB <- prepareCuffExample()
```

---

preSpliceRFilter *Filters on spliceR-lists for reduction of data sets*

Description

Applies a number of filters on a spliceR object to reduce data set size before running downstream analyses.

Usage

```
preSpliceRFilter(spliceRobject, filters, expressionCutoff=0)
```

Arguments

spliceRobject   a SpliceRList object, either created manually from transcript and exon information (see [SpliceRList]), or created by [prepareCuff] from CuffLinks data.

filters   vector, giving the filters that should be applied - any combinations of 'geneOK', 'expressedGenes', 'sigGenes', 'isoOK', 'expressedIso', 'isoClass' and/or 'sigIso'. Works only for data from cufflinks, as a manually generated SpliceRList does not include these metacolumns.

expressionCutoff
   Numeric, giving the expression threshold (often in FPKM) used for the 'expressedGenes' and 'expressedIso' filter. Default value is 0.

Details

Often, many genes and isoforms are flagged as not "OK" or "LOWDATA" by Cufflinks, indicating low confidence in these. This function is handy for reducing the data size of a Cufflinks data set to reduce running times for downstream analyses.

Note, that preSpliceRFilter removes trancsripts from the dataset permanently, reducing size, while the filter options of [spliceR] and [annotatePTC] only selects transcripts for analysis, but does not remove any data.

Value

A SpliceRList with transcripts after filtering.

Author(s)

Kristoffer Vitting-Seerup, Johannes Waage

References

Vitting-Seerup K , Porse BT, Sandelin A, Waage J. (2014) spliceR: an R package for classification of alternative splicing and prediction of coding potential from RNA-seq data. BMC Bioinformatics 15:81.

Examples

```
#Load cufflinks example data
cuffDB <- prepareCuffExample()

#Generate SpliceRList from cufflinks data
cuffDB_spliceR <- prepareCuff(cuffDB)

#Filter
cuffDB_spliceR_filtered <- preSpliceRFilter(cuffDB_spliceR, filters=c("expressedIso", "isoOK", "expressedGenes",
```

---

spliceR   *Splice class detection from assembled RNA-seq data*

Description

Splice class detection from assembled RNA-seq data.

Usage

```
spliceR(transcriptData, compareTo, filters, expressionCutoff=0, useProgressBar=T)
```

Arguments

transcriptData	a SpliceRList object, either created manually from transcript and exon information (see [SpliceRList](#)), or created by [prepareCuff](#) from CuffLinks data.
compareTo	a character, either 'preTranscript', for comparison to the hypothetical pre-splicing transcript for each gene, or a character, indicating the reference sample against which to classify splicing events.
filters	vector, giving the filters that should be applied - any combinations of 'geneOK', 'expressedGenes', 'sigGenes', 'isoOK', 'expressedIso', 'isoClass' and/or 'sigIso'. Works only for data from Cufflinks, as a manually generated SpliceRList does not include these metacolumns.
expressionCutoff	
	Numeric, giving the expression threshold (often in FPKM) used for the 'expressedGenes' and 'expressedIso' filter. Default value is 0.
useProgressBar	Boolean, indicating whether to use progressbars. For compatibility. Default = TRUE.

Details

The following filters are allowed for filters: geneOK requires Cufflinks to have reported the quantification of the gene as OK. Only works on transcript data from Cufflinks. expressedGenes requires the parent gene to be expressed. sigGenes requires the parent gene to be expressed in at least one sample. isoOK requires cufflinks to have reported the quantification of the isoform as OK. Only works on transcript data from Cufflinks. expressedIso requires the isoform to be expressed in at least one sample. isoClass removed transcripts marked by cufflinks to be either 'possible

pre-mRNA fragment', 'Possible polymerase run-on fragment', or 'Repeat'. Only works on transcript data from Cufflinks. `sigIso` requires cufflinks to have reported the isoform as significant deregulated between samples. Only works on transcript data from Cufflinks.

Value

A `SpliceRList`, identical to input SpliceRList transcriptData, with the `transcript_features` slot containing the following additional columns:

spliceR.major	yes/no, indicating if this isoform is the major isoform expressed of the relevant gene for the reference sample.
spliceR.IF1	Isoform Fraction of total gene expression for sample 1
spliceR.IF2	Isoform Fraction of total gene expression for sample 2
spliceR.dIF	Delta IF (sample 2-sample 1)
spliceR.ESI	Number of exon skipping/inclusion events for this isoform
spliceR.MEE	Number of mutually exclusive exon events for this isoform
spliceR.MESI	Number of mutliple exon skipping/inclusion events for this isoform
spliceR.ISI	Number of intron skipping/retention events for this isoform
spliceR.A5	Number of alternative 5' splice site events for this isoform
spliceR.A3	Number of alternative 3' splice site events for this isoform
spliceR.ATSS	0/1, 1 indicating that this isoform uses an alternative transcription start site
spliceR.ATTS	0/1, 1 indicating that this isoform uses an alternative transcription terminating site
spliceR.analyzed	
	Yes/no, indicating if this isoform was analyzed(yes), or removed in filtering(no)
spliceR.ESI.start	
	Genomic start location(s) of ESI elements spliced in/out
spliceR.ESI.end	
	Genomic end location(s) of ESI elements spliced in/out
spliceR.MEE.start	
	Genomic start location(s) of MEE elements spliced in/out
spliceR.MEE.end	
	Genomic end location(s) of MEE elements spliced in/out
spliceR.MESI.end	
	Genomic end location(s) of MESI elements spliced in/out
spliceR.MESI.start	
	Genomic start location(s) of MESI elements spliced in/out
spliceR.ISI.start	
	Genomic start location(s) of ISI elements spliced in/out
spliceR.ISI.end	
	Genomic end location(s) of ISI elements spliced in/out
spliceR.A5.start	
	Genomic start location(s) of A5 elements spliced in/out

`spliceR.A5.end` Genomic end location(s) of A5 elements spliced in/out

`spliceR.A3.start`

Genomic start location(s) of A3 elements spliced in/out

`spliceR.A3.end` Genomic end location(s) of A3 elements spliced in/out

`spliceR.ATSS.start`

Genomic start location(s) of ATSS elements spliced in/out

`spliceR.ATSS.end`

Genomic end location(s) of ATSS elements spliced in/out

`spliceR.ATTS.start`

Genomic start location(s) of ATTS elements spliced in/out

`spliceR.ATTS.end`

Genomic end location(s) of ATTS elements spliced in/out

Author(s)

Kristoffer Vitting-Seerup, Johannes Waage

References

Vitting-Seerup K , Porse BT, Sandelin A, Waage J. (2014) spliceR: an R package for classification of alternative splicing and prediction of coding potential from RNA-seq data. BMC Bioinformatics 15:81.

Examples

```
#Load cufflinks example data
cuffDB <- prepareCuffExample()

#Generate SpliceRList from cufflinks data
cuffDB_spliceR <- prepareCuff(cuffDB)

#Reduce dataset size for fast example runtime
cuffDB_spliceR[[1]] <- cuffDB_spliceR[[1]][1:500]

#Run spliceR
mySpliceRList <- spliceR(cuffDB_spliceR, compareTo=preTranscript, filters=c(expressedGenes,geneOK, isoOK, expres
```

SpliceRList *Transcript data and annotation object for spliceR*

Description

Creates a `SpliceRList` object from two `GRanges` objects, an assembly id, and a source id. The first `GRanges`, transcript_features, containing a list of transcripts, and including the columns `gene_id` for gene id, `tx_id` for transcript id, `sample_1` and `sample_2` for sample identifiers, `expression_1` and `expression_2` for expression values for sample 1 and sample 2, respectively (typically FPKM values or some other normalized count values), and additional optional columns (see `prepareCuff`).

The second, exon_features, containing a list of exons, and including the columns gene_id for gene id and tx_id for transcript id. Assembly id, denoting genome assembly ('hg19', 'hg18', 'mm9', etc.) Source id, denoting source of transcript assembly (currently 'cufflinks' or 'other') Note, that the cromosome identifiers should match the assembly. For experiments

Usage

```
SpliceRList(transcript_features, exon_features, assembly_id, source_id, conditions, transcripts_plot=
```

Arguments

transcript_features

GRanges object containing transcript features.

exon_features   GRanges object containing transcript features.

assembly_id   character, giving genome assemlby.

source_id   A character, either "cufflinks" or "granges", stating source of transcript assembly.

conditions   A character vector, giving the samples or conditions for the RNA-seq experiment.

transcripts_plot

A dataframe, reserved for plotting functions

filter_params   A character vector, reserved for plotting functions.

Details

For cufflinks data, call prepareCuff to prepare a SpliceRList. For other RNA-seq assemblies, use this constructor to create a SpliceRList.

See the spliceR vignette for an example of creating a spliceRList from another source than Cufflinks.

Value

A SpliceRList object.

Author(s)

Kristoffer Vitting-Seerup, Johannes Waage

References

Vitting-Seerup K , Porse BT, Sandelin A, Waage J. (2014) spliceR: an R package for classification of alternative splicing and prediction of coding potential from RNA-seq data. BMC Bioinformatics 15:81.

---

spliceRPlot   *Plot venn diagrams of alternative splicing events*

Description

Plot venn diagrams of alternative splicing events vs. samples.

Usage

```
spliceRPlot(spliceRobject, evaluate="nr_transcript", asType="All",colors=NULL, alpha=NULL, reset=FALS
```

Arguments

spliceRobject   A SpliceRList object, processed and returned by [spliceR](#).

evaluate   A character, giving the evaulation criteria (see details).

asType   The alternative splicing type to visualize, either 'ESI','MEE','MESI','ISI','A5', 'A3','ATSS','ATTS' or 'All'. See [spliceR](#) for a full description of alternative splicing types.

colors   Character, giving plot colors for each condition. Must be same length as number of conditions. If NULL, colors from the ColorBrewer "Dark2" pallette is used.

alpha   A numeric between 0 and 1, giving the transparency of the plot. If NULL, the alpha will be set optimally depending on number of samples.

reset   A boolean, indicating whether to reinitialize the SpliceRList object for faster replotting.

filters   vector, giving the filters that should be applied - any combinations of 'geneOK', 'expressedGenes', 'sigGenes', 'isoOK', 'expressedIso', 'isoClass' and/or 'sigIso'. Works only for data from cufflinks, as a manually generated SpliceRList does not include these metacolumns.

expressionCutoff

Numeric, giving the expression threshold (often in FPTKM) used for the 'expressedGenes' and 'expressedIso' filter. Default value is 0.

Details

Upon inital usage of spliceRPlot, the SpliceRList is initiated with internal data, allowing for faster replotting. If the SpliceRList changes because of filtering or other manipulation, rerun spliceR-Plot with reset=T. For the evaulate parameter, the following are valid: 'nr_transcript','nr_genes', 'nr_transcript_pr_gene', 'nr_AS', 'mean_AS_gene', 'mean_AS_transcript', 'mean_transcript_exp', 'mean_gene_exp'. 'nr_transcript' outputs number of transcripts, 'nr_AS' outputs number of alternative splicing events, 'mean_as' outputs the average number of AS events per gene, 'mean_transcript_exp' outputs the mean transcript expression and 'mean_gene_exp' output the mean gene expression. For a detailed description of filters, see [spliceR](#).

Value

A SpliceRList, contianing additional temporary data for fast subsequent re-plotting.

Author(s)

Kristoffer Vitting-Seerup, Johannes Waage

References

Vitting-Seerup K , Porse BT, Sandelin A, Waage J. (2014) spliceR: an R package for classification of alternative splicing and prediction of coding potential from RNA-seq data. BMC Bioinformatics 15:81.

Examples

```
#Load cufflinks example data
cuffDB <- prepareCuffExample()

#Generate SpliceRList from cufflinks data
cuffDB_spliceR <- prepareCuff(cuffDB)

#Reduce dataset size for fast example runtime
cuffDB_spliceR[[1]] <- cuffDB_spliceR[[1]][1:500]

#Run spliceR
mySpliceRList <- spliceR(cuffDB_spliceR, compareTo=preTranscript, filters=c(expressedGenes,geneOK, isoOK, expres

#Plot number of exon skipping/inclusion events
mySpliceRList <- spliceRPlot(mySpliceRList, evaluate="nr_AS", asType="ESI")
```

---

topIsoShift   *Returns top transcripts in terms of isoform switching*

Description

Returns top transcripts in terms of isoform switching.

Usage

```
topIsoShift(spliceRObject, n=10)
```

Arguments

spliceRObject	a SpliceRList object, that has been successfully analyzed and annotated by spliceR.
n	An integer, giving the number of transcripts to return.

Details

This helper function returns the transcripts with the highest delta-isoform fraction (dIF) between samples. If the data is based on cufflinks (source_id=="cufflinks"), only isoforms flagged significantly changing between samples will be returned.

Value

A dataframe, containing a cast of the GRanges rows of the highest scoring transcripts by dIF.

Author(s)

Kristoffer Vitting-Seerup, Johannes Waage

References

Vitting-Seerup K , Porse BT, Sandelin A, Waage J. (2014) spliceR: an R package for classification of alternative splicing and prediction of coding potential from RNA-seq data. BMC Bioinformatics 15:81.

Examples

```
#Load cufflinks example data
cuffDB <- prepareCuffExample()

#Generate SpliceRList from cufflinks data
cuffDB_spliceR <- prepareCuff(cuffDB)

#Reduce dataset size for fast example runtime
cuffDB_spliceR[[1]] <- cuffDB_spliceR[[1]][1:500]

#Run spliceR
mySpliceRList <- spliceR(cuffDB_spliceR, compareTo=preTranscript, filters=c(expressedGenes,geneOK, isoOK, expres

#Get top dIF transcripts
topIsoShift(mySpliceRList, n=20)
```

---

totalNumberOfAS *Returns total number of alternative splicing events*

Description

Returns total number of alternative splicing events an SpliceRList.

Usage

```
totalNumberOfAS(spliceRObject)
```

Arguments

spliceRObject   a SpliceRList object returned by spliceRPlot.

Details

This helper function returns number of total number of alternative splicing events. Object must be analyzed by spliceRPlot first.

Value

A vector, giving the total number of splicing events for each splice class.

Author(s)

Kristoffer Vitting-Seerup, Johannes Waage

References

Vitting-Seerup K , Porse BT, Sandelin A, Waage J. (2014) spliceR: an R package for classification of alternative splicing and prediction of coding potential from RNA-seq data. BMC Bioinformatics 15:81.

Examples

```
#Load cufflinks example data
cuffDB <- prepareCuffExample()

#Generate SpliceRList from cufflinks data
cuffDB_spliceR <- prepareCuff(cuffDB)

#Reduce dataset size for fast example runtime
cuffDB_spliceR[[1]] <- cuffDB_spliceR[[1]][1:500]

#Run spliceR
mySpliceRList <- spliceR(cuffDB_spliceR, compareTo=preTranscript, filters=c(expressedGenes,geneOK, isoOK, expres

#Plot number of exon skipping/inclusion events
mySpliceRList <- spliceRPlot(mySpliceRList, evaluate="nr_AS", asType="ESI")

totalNumberOfAS(mySpliceRList)
```

---

transcripts   *Returns the transcript or exon GRanges from a SpliceRList object*

Description

Returns the transcript or exon GRanges object from a SpliceRList object.

Usage

```
transcripts(transcriptData)
exons(transcriptData)
```

Arguments

transcriptData a SpliceRList object.

Details

These helper functions returns either the "transcript_features" or "exon_features" object of a SpliceRList object.

Value

A `GRanges` object. See `SpliceRList` for a full description of the contents of a SpliceRList.

Author(s)

Kristoffer Vitting-Seerup, Johannes Waage

References

Vitting-Seerup K , Porse BT, Sandelin A, Waage J. (2014) spliceR: an R package for classification of alternative splicing and prediction of coding potential from RNA-seq data. BMC Bioinformatics 15:81.

Examples

```
#Load cufflinks example data
cuffDB <- prepareCuffExample()

#Generate SpliceRList from cufflinks data
cuffDB_spliceR <- prepareCuff(cuffDB)

myTranscripts <- transcripts(cuffDB_spliceR)
myExons <- exons(cuffDB_spliceR)
```

Index