Package 'REIDS'

July 30, 2018

Type Package

Title Random Effects for the Identification of Differential Splicing

Version 0.1.0

Date 2018-08-19

Author Marijke Van Moerbeke

Maintainer Marijke Van Moerbeke <marijke.vanmoerbeke@uhasselt.be>

Description Contains the REIDS
model presented in Van Moerbeke et al
(2017) <doi:10.1186/s12859-017-1687-8> for the detection of alternative splicing. The method is extended by incorporating junction information for the
assessment of alternative splicing. The vignette introduces the model and shows an example work flow.

License GPL-3

LazyData true

Depends R (>= 2.10), aroma.affymetrix, aroma.core, GenomeGraphs,
biomaRt

Imports data.table,RColorBrewer,MCMCpack,lmtest,methods

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2018-07-30 10:10:03 UTC

R topics documented:

`AnnotateGenes` *"AnnotateGenes"*

Description

The AnnotateGenes function annotates the TC ID to a HGNC Gene symbol.

Usage

```
AnnotateGenes(transcript.IDs, trinfo)
```

Arguments

`transcript.IDs`	The transcript IDs to annotate.
`trinfo`	The transcript data frame from which to retrieve the annotation symbol.

AnnotateGeneSymbol *"AnnotateGeneSymbol"*

Description

The AnnotateGeneSymbol function annotates HGNC Gene symbol to an ensemble region.

Usage

```
AnnotateGeneSymbol(symbol.to.annotate)
```

Arguments

`symbol.to.annotate`
: Gene symbol to annotate to an ensembl region.

arcplot *Arc Diagram Plot*

Description

Give me an edgelist and I'll help you plot a pretty damn arc diagram

Usage

```
arcplot(edgelist, vertices, sorted = FALSE, decreasing = FALSE,
  ordering = NULL, labels = NULL, horizontal = TRUE, above = NULL,
  col.arcs = "#5998ff77", lwd.arcs = 1.8, lty.arcs = 1, lend = 1,
  ljoin = 2, lmitre = 1, show.nodes = TRUE, pch.nodes = 19,
  cex.nodes = 1, col.nodes = "gray80", bg.nodes = "gray80",
  lwd.nodes = 1, show.labels = TRUE, col.labels = "gray55",
  cex.labels = 0.9, las = 2, font = 1, line = 0, outer = FALSE,
  adj = NA, padj = NA, axes = FALSE, ...)
```

Arguments

`edgelist`	basically a two-column matrix with edges
`vertices`	optional vector of vertex names corresponding with those in the edgelist
`sorted`	logical to indicate if nodes should be sorted (default `FALSE`)
`decreasing`	logical to indicate type of sorting (used only when `sorted=TRUE`)
`ordering`	optional numeric or string vector providing the ordering of nodes. When provided, this parameter overrides `sorted=TRUE`). See the details section for more information.
`labels`	optional string vector with labels for the nodes

horizontal	logical indicating whether to plot in horizontal orientation
above	optional vector indicating which arcs should be displayed above (or to the right) and below (or to the left) of the axis
col.arcs	color for the arcs (default "gray50")
lwd.arcs	line width for the arcs (default 1)
lty.arcs	line type for the arcs (see par)
lend	the line end style for the arcs (see par)
ljoin	the line join style for the arcs (see par)
lmitre	the line mitre limit for the arcs (see par)
show.nodes	logical indicating whether to show node symbols
pch.nodes	plotting 'character', i.e. symbol to use when plotting nodes (pch.nodes=0:25)
cex.nodes	expansion of the node symbols (default 1)
col.nodes	color of the node symbols (default "gray50")
bg.nodes	background (fill) color for the node symbols given by pch.nodes=21:25
lwd.nodes	line width for drawing node symbols (see points)
show.labels	logical indicating whether to show node labels
col.labels	color of the node labels (default "gray50")
cex.labels	expansion of node labels (default "gray50")
las	numeric in 0,1,2,3; the style of axis labels (see par)
font	font used for node labels (see par)
line	on which margin line the node labels are displayed, starting at 0 counting outwards (see mtext)
outer	use outer margins, if available, to plot node labels (see mtext)
adj	adjustment for each string in reading direction (see mtext)
padj	adjustment for each string perpendicular to the reading direction (see mtext)
axes	logical indicating whether to plot the axes (default FALSE)
...	further graphical parameters (see par), including family, xpd, main, asp, etc.

Details

The arcs are scaled such that they fit in a plot region with its x-axis ranging from zero to one. Node symbols and labels can be optionally displayed. Node symbols are displayed through the function points. In turn, node labels are displayed through the function mtext.

When ordering is provided in numeric format and node labels are strings, the labels are alphabetically ordered first, and then nodes are sorted according to the provided ordering.

If ordering is provided in string format, the node labels must be strings as well. The nodes will be sorted according to ordering.

Author(s)

Gaston Sanchez

ASExons *"ASExons"*

Description

The ASExons functions alternatively spliced exons from the exon scores and array scores. It filters probesets on their exon scores, adjusts p-values for multiplicity and only keeps the significant probesets.

Usage

```
ASExons(ExonScores, ArrayScores, Exonthreshold = 0.5, Groups = list(group1 =
  NULL, group2 = NULL), paired = FALSE, significancelevel = 0.05,
  Location = NULL, Name = "REIDSAS")
```

Arguments

ExonScores	The path to the file with the exon scores of the probe sets.
ArrayScores	The path to the file with the array scores of the probe sets.
Exonthreshold	The exon score threshold to be maintained. If not NULL, probe sets with an exon score lower than this value are not considered further and the p-values will be adjusted for multiplicity after testing. If NULL, all probesets are considered and a multiplicity correction is not performed.
Groups	A list with elements specifying the columns of the data in each group.
paired	Logical. Are the groups paired? If TRUE the mean paired differences are calculated and tested whether these are significantly different from zero or not.
significancelevel	The significance level to be maintained on the p-values. The filtering on the significance is conducted only if an Exonthreshold is specified and the p-value are adjusted for multiplicity.
Location	A character string indication the place where the outputs are saved.
Name	A character string with the name of the ouput file. Defaults to "REIDSAS".

Value

A data frame with one line per exon. The columns contain the gene ID, the exon ID, the exon score the test statistic, a p-value and an adjusted p-value. If the groups are paired also the mean paired difference is given. Only the probesets with high enough exon scores and a significant test are kept in the data frame.

Examples

```
## Not run:
data(TC1500264)

PivotTransformData(Data=TC1500264,GeneID=NULL,ExonID=NULL,
```

```
REMAPSplitFile="TC1500264_Gene_SplitFile.txt",Location="Output/",Name="TC1500264_Pivot")

REIDSFunction(ASPSR=c(), Indices="Output/TC1500264_LineIndex.csv",
DataFile="Output/TC1500264_Pivot.csv",nsim=50,informativeCalls=FALSE,Summarize=
c("WeightedAll","EqualAll"),rho=0.5,Low_AllSamples=c(),Groups=list(c(1:3),c(4:6)),
Location="Output",Name="TC1500264")

TC1500264_1vs2=ASExons(ExonScores="Output/TC1500264_ExonScores.txt",ArrayScores=
"Output/TC1500264_ArrayScores.txt",Exonthreshold=0.5,Groups=list(c(1:3),c(4:6)),
paired=FALSE,significancelevel=0.05)

## End(Not run)
```

`CreateOutput` *"CreateOutput"*

Description

The CreateOutput functions writes the .RData files returned by the REIDS_HPCVersion to .txt files: "Name_INICalls.txt", Name_ExonScores.txt", "Name_ArrayScores.txt" and the summarized values distributed across "Name_WeightedAll.txt", "Name_EqualAll.txt", "Name_WeightedConst.txt" and "Name_EqualConst.txt". The function can also be used for the returned files of REIDSIsoformAssesment_HPCVersion for which it will create the files: "Name_ASInfo.txt" "Name_Compositions.txt","Name_GroupTra and "Name_NovelConnections". The function is advised to be used with the CreateOutput.R and CreateOutput.pbs file in the documentation folder.

Usage

```
CreateOutput(ID, Groups, Location = "", Name)
```

Arguments

`ID`	A data frame with a "geneID" column.
`Groups`	A list with elements specifying the columns of the data in each group.
`Location`	The location where the file should be saved.
`Name`	A name for the returned list.

Value

.txt files with the information of the REIDSFunction and REIDSJunctionAssesment.

DataProcessing *DataProcessing*

Description

The DataProcessing function processes raw .CEL files to probe intensities values with the help of functions of the aroma.affymetrix package. It returns a data frame and saves it as an .RData file.

Usage

```
DataProcessing(chipType = "HuEx-1_0-st-v2", tags = "coreR3,A20071112,EP",
  ExonSummarization = TRUE, GeneSummarization = TRUE, FIRMA = TRUE,
  Location = NULL, Name = "", verbose = TRUE)
```

Arguments

chipType	The name of the chip type of the array data.
tags	Tags that is added to the chipType.
ExonSummarization	Logical. Should the data be summarized at the exon level?
GeneSummarization	Logical. Should the data be summarized at the gene level?
FIRMA	Logical. Should the FIRMA model be performed on the data?
Location	The location where the .rda file is to be stored. If NULL, a list containing the requested objects is returned to the user.
Name	A string indicating the prefix for the names of the outputs to be saved at the Location. Defaults to "".
verbose	Logical. If TRUE, messages are printed during the data processing.

Details

The DataProcessing function is a wrapper of several functions of the aroma.affymetrix package. To obtain the data to perform the REIDS model on the raw .CEL files are background corrected with the rma background correction and normalization is performed with the quantile normalization. In order for the function to run properly, a chipType and its possible tags need to be specified. It is also important to have the same folder structure as required by the aroma.affymetrix package. This implies the following: a rawData folder with therein a folder with the "Name" parameter. This "Name" folder should contain a folder with the chipType name and herein the .CEL files should be placed. Also a folder annotationData should be present. Herein a folder chipTypes should be make which contains folders for type of chips with the respective names. In the folder of each chiptype the corresponding .cdf file should be saved. If specified, the processed data will be saved at a specific location as a data frame with the first colum the gene IDs and the second column the exon IDs. All other columns contain the sample values. Further the object also contains a vector of the unique gene ID and a vector of the unique exon IDs. If requested, exon and gene level summarization are performed and saved as data frames at the specified location. Further,the option is provided to perform the FIRMA model on the data as well.

Value

An .rda file that is saved at the specified location.

Examples

```
## Not run:
DataProcessing(chipType="HTA-2_0",tags="*,r",
ExonSummarization=TRUE,GeneSummarization=TRUE,FIRMA=TRUE,
Location="HTAData",Name="HTAData",verbose=TRUE)

## End(Not run)
```

ExampleFirmaOutput *Example data set for FIRMA*

Description

An example data set for the FIRMA functions

Usage

```
data(ExampleFirmaOutput)
```

Format

An object of class "data.frame".

Examples

```
data(ExampleFirmaOutput)
```

ExonTesting *"ExonTesting"*

Description

The ExonTesting function performs a t-test (2 groups) or F-test (more than 2 groups) between the array scores of predefined groups. If specified, probe sets are filtered out on exon scores and significance level. The function is the internal function of ASExons.

Usage

```
ExonTesting(ExonScores, ArrayScores, Exonthreshold = NULL, Groups = list(),
  paired = FALSE, significancelevel = NULL)
```

Arguments

ExonScores	The path to the file with the exon scores of the probe sets.
ArrayScores	The path to the file with the array scores of the probe sets.
Exonthreshold	The exon score threshold to be maintained. If not NULL, probe sets with an exon score lower than this value are not considered further and the p-values will be adjusted for multiplicity after testing. If NULL, all probesets are considered and a multiplicity correction is not performed.
Groups	A list with elements specifying the columns of the data in each group.
paired	Logical. Are the groups paired? If TRUE the mean paired differences are calculated and tested whether these are significantly different from zero or not.
significancelevel	The significance level to be maintained on the p-values. The filtering on the significance is conducted only if an Exonthreshold is specified and the p-value are adjusted for multiplicity.

Value

A data frame with one line per exon. The columns contain the gene ID, the exon ID, the exon score, the test statistic, a p-value and an adjusted p-value. If the groups are paired also the mean paired difference is given. The p-values are adjusted for multiplicity and filtered on significance if significancelevel is not NULL.

ExpressionLevelPlot *"ExpressionLevelPlot"*

Description

The ExpressionLevelPlot produces a plot of the expression levels of a specific exon and its corresponding gene.

Usage

```
ExpressionLevelPlot(GeneID = NULL, ExonID = NULL, Data,
  GeneLevelData = NULL, ExonLevelData = NULL, Groups, ylabel = NULL,
  title = "")
```

Arguments

GeneID	The gene ID of the gene of interest.
ExonID	The exon ID of the exon of interest.
Data	The processed data as returned by DataProcessing. This is were the observed probe intensities will be retrieved.
GeneLevelData	The gene level summarized data to retrieve the gene level values.
ExonLevelData	The exon level summarized data to retrieve the exon level values.

Groups	The groups of interest in the data.
ylabel	The label for the y-axis.
title	A title for the plot.

Examples

```
## Not run:
data(TC12000010)
data(TC12000010_ExonLevel)
data(TC12000010_GeneLevel)
ExpressionLevelPlot(GeneID="TC12000010",ExonID="PSR12000150",
Data=TC12000010,GeneLevelData=TC12000010_GeneLevel,ExonLevelData
=TC12000010_ExonLevel,Groups=list(c(1:9),c(10:18)),ylabel="",
title="PSR12000150")

## End(Not run)
```

FIRMAScores *"FIRMAScores"*

Description

The FIRMAScores function performs an analyais on the FIRMA scores of the sanples. If the groups are not paired, the Firma all sample score will be the minimum value of group 1. A test statistic is perforrmed on the grouping of interest to see if there is a significant difference between them. If the data is paired, the mean paired differences are obtained and tested versus zero.

Usage

```
FIRMAScores(Data, InformativeExons = NULL, groups = list(group1 =
  list(group1a = NULL, group1b = NULL), group2 = NULL), paired = FALSE,
  significancelevel = NULL)
```

Arguments

Data	The output of the FIRMA model
InformativeExons	A character vector of exon IDs. As for the REIDS model probesets are filtered out by I/NI calls model and later on exon score, the remaining exons can be specified here. Only these shall be considered in the FIRMA analysis to make the results between REIDS and FIRMA more comparable
groups	A list with two elements speficifing the columns of the data of group 1 in group1 and those of group 2 in group2. Here is the possibility to specifify multiple subgroups in group 1 for the calculation of the all FIRMA sample score.
paired	Logical. Are the groups paired? If TRUE the mean paired differences are calculated and tested whether these are significantly different from zero or not.
significancelevel	If specified, filtering is conducted on the p-values.

Details

The input to this function is the output of the FIRMA model. On the FIRMA scores, a t-test is performed. This is either between groups if the data is not paired and on the mean paired differences if the data is paired. If no pairing is present, an all sample FIRMA score is computed as the maximum of the minimum values of the of the subgroups in group 1. The returned p-value is adjusted for multiplicity. If a vector is given in InformativeExons, only these exon IDS are considered in the calculations.

Value

A data frame with one line per exon. The columns contain the gene ID, the exon ID, the test statistic, a p-value and an adjusted p-value. If the groups are paired also the mean paired difference is given. Only the probesets with high enough exon scores and a significant test are kept in the data frame.

Examples

```
data(ExampleFirmaOutput)
FIRMATest=FIRMAScores(Data=ExampleFirmaOutput,InformativeExons=NULL,groups=list(group1=
list(group1a=c(1,2,3),group1b=NULL),group2=c(4,5,6)),paired=FALSE,significancelevel=NULL)
```

GetIntensities *"GetIntensities"*

Description

The GetIntensities function retrieves the exon expression level in the data.

Usage

```
GetIntensities(trans, Data, Groups)
```

Arguments

trans	The transcript ID.
Data	The data frame with the exon level expression of the transcript.
Groups	A list with the groups (columns) of interest

iniREIDS *"iniREIDS"*

Description

The function is part of the larger REIDS function and performs the I/NI calls filtering model on the data for a single gene to select only the informative probesets. The method is performed if Informative=TRUE in the REIDS function before before applying the REIDS model itself. The function is written to fit in the flow of the REIDS model

Usage

```
iniREIDS(SubgeneData, nsim = 1000)
```

Arguments

SubgeneData	A subset of the data. Particularly, a subset of the data corresponding to one gene.
nsim	The number of iterations to perform.

Value

A list with an item per gene. Per gene it is mentioned wich probesets are informative and which are not.

JunInfo *JunInfo*

Description

JunInfo functions asses the junction information for a single gene

Usage

```
JunInfo(file_name, file_pos, line_length, ASPSR = c(), Juninfo = "User",
  JAnnotI, JAnnot = NULL, EandTrAnnotI = NULL, EandTrAnnot = NULL,
  PartiallyAnnotated = FALSE, positionData = NULL, transcriptData = NULL,
  Groups = list(), Low_AllSamples = c(), Low_GSamples = c(),
  Plot = FALSE, Location = NULL, Name = "")
```

Arguments

file_name	The name of the pivot transformed .csv file.
file_pos	The position in the file where to start reading.
line_length	The length of the line to read.
ASPSR	The AS probe sets as identified by ASExons.
Juninfo	A parameter specifying wether the annotations are user of Ensembl defined. If JunInfo is "User" (default) the annotations provided in EandTrAnnot are used. If JunInfo is "Ensembl" the annotations in EandTrAnnot are used to set up tje junction associations but the gene name and position in transcriptData and positionData are used to connect with the Ensembl data base and retrieve corresponding information.
JAnnotI	The file name with line indices for the junction associations.
JAnnot	The file name with the junction associations.
EandTrAnnotI	The file name with line indices for the exon and isoform annotations.
EandTrAnnot	The file name with the exon and isoform annotations.
PartiallyAnnotated	Logical. Should the exon annotations with partially annotated probe sets still be included? If FALSE, these are excluded. If TRUE, these are included. Default is FALSE.
positionData	The file with the chromosome start and ends for the probe sets. Only needed in JunInfo=Ensembl.
transcriptData	The file with gene name of the transcripts. Only needed in JunInfo=Ensembl.
Groups	A list with elements speficifing the columns of the data in each group.
Low_AllSamples	A character vector containing the probe sets which are not DABG in all samples.
Low_GSamples	A list with a character vector per group containing the probe sets which are not DABG in that group.
Plot	Should a plot of the gene model be made?
Location	A character string indication the place where the outputs are saved.
Name	A character string with the name of the ouput file.

Details

The plot is produced by the arcplot function of the arcdiagram package (https://github.com/gastonstat/arcdiagram)

Value

The function returns four files. The first file has name "Name_ASInfo.txt" and contains a line per probe set. It shows the reached decision regarding the probe set (Const/AS/not DABG),its linking and exclusion junctions, the fold change, the AS type and its annotated exons. The second file, "Name_Compositions.txt", is a list of all found transcripts for a particular TC ID. The third file,"Name_GroupTranscripts.txt" indicates whether a specific transcript is present or absent in a group. The fourth file "Name_NovelConnections.txt" contains junctions which are showing an undocumented connection between probe sets.

node_coords *Node Coordinates*

Description

Computes axis locations of each node. This function can be helpful when you want to separately plot the node labels using the function mtext.

Usage

```
node_coords(edgelist, sorted = FALSE, decreasing = FALSE, ordering = NULL,
  labels = NULL)
```

Arguments

edgelist	basically a two-column matrix with edges
sorted	logical to indicate if nodes should be sorted
decreasing	logical to indicate type of sorting
ordering	optional numeric vector providing the ordering of nodes
labels	character vector with labels for the nodes

Value

a vector with the location of nodes in the x-axis

Author(s)

Gaston Sanchez

PivotTransformData *"PivotTransformation"*

Description

The PivotTransformation function converts a data frame with multiple rows per gene into a .csv file with one row per gene. This is the first step in data transformation to apply the REIDS function on a HPC Cluster.

Usage

```
PivotTransformData(Data, GeneID = NULL, ExonID = NULL,
  REMAPSplitFile = NULL, NotAnnotated = FALSE, Location = NULL,
  Name = "Pivot")
```

Arguments

Data	The data frame to be transformed.
GeneID	A character vector of the the gene IDs that correspond to the rows of the data frame. Necessary if no GeneID column is present in the data frame
ExonID	A character vector of the the gene IDs that correspond to the rows of the data frame. Necessary if no ExonID column is present in the data frame
REMAPSplitFile	The name of the file with the REMAP information regarding the split of the probe sets if the TC ID is annotated to mutiple genes.
NotAnnotated	Logical. Should the probe sets which are not annotated to a gene still be included? If FALSE, these are excluded. If TRUE, these are included. Default is FALSE.
Location	The location where the file should be saved. If NULL, the object is returned to the user. Otherwise, a file with the specified name is created.
Name	The name of the output file. Defaults to "Pivot".

Details

All information concerning one gene is gathered. The first column of the returned data frame is the gene ID, the second column contains the exon IDs of all exons of that gene. The third colum indicates the number of probes per exon, the fourth contains the values of thos probes per sample and the last column contains the sample names.This way a .csv file is created for processing on a HPC cluster.

Value

A data frame with one row per gene. This row contains the values for each exon per sample and is convenient for processing on a HPC cluster. Futher also a data frame with a column of the gene ID's is returned.

Examples

```
data(TC12000010)

PivotTest=PivotTransformData(Data=TC12000010,GeneID=NULL,ExonID=NULL,
Location=NULL)

## Not run:
data(TC1500264)

PivotTransformData(Data=TC1500264,GeneID=NULL,ExonID=NULL,
REMAPSplitFile="TC1500264_Gene_SplitFile.txt",Location=
"Output",Name="TC1500264_Pivot")

## End(Not run)
```

REIDS *Random Effects for the Identification of Differential Splicing.*

Description

The REIDS package contains the REIDS mixed model for the identification of differentially spliced exons and ranking by junction support.

REIDSFunction *"REIDSFunction"*

Description

The REIDSFunction performs the REIDS model on the pivot transformed data by calling on the line indexed file. The REIDS model is performed gene by gene and the returned outputs are knitted together.

Usage

```
REIDSFunction(ASPSR = c(), Indices, DataFile, nsim = 1000,
  informativeCalls = TRUE, Summarize = FALSE, rho = 0.5,
  Low_AllSamples = c(), Groups, Location = NULL, Name = "REIDS")
```

Arguments

ASPSR	A vector with alternatively spliced probe sets which are taken out of the analysis and summarization. This is useful if Summarize is "WeightedConst" and/or "EqualConst".
Indices	The .csv file created by Line_Indexer.py which contains indices for every gene in geneIDs.
DataFile	The .csv file created by PivotTransformation.
nsim	The number of iterations to perform. Defaults to 1000.
informativeCalls	Logical. Should the I/NI calls method be perform before applying the REIDS model?
Summarize	A character vector specifying wich summarization method is to be performed. The choices are "EqualAll", "WeightedAll", "EqualConst" and "WeightedConst". The former two use all probe sets while the latter use only the constituitive probe sets. Summarization on the constistuitive probe sets will only be performed if ASPSR is specified.
rho	The threshold for filtering in the I/NI calls method. Probesets with scores higher than rho are kept.
Low_AllSamples	A character vector containing the probe sets which are not DABG in all samples.

Groups	A list with elements specifying the columns of the data in each group.
Location	A character string indication the place where the outputs are saved.
Name	A name for the output to be saved at Location. Defaults to "REIDS".

Value

The functions writes the obtained information to .txt files: "Name_INICalls.txt", Name_ExonScores.txt", "Name_ArrayScores.txt" and the summarized values distributed across "Name_WeightedAll.txt", "Name_EqualAll.txt", "Name_WeightedConst.txt" and "Name_EqualConst.txt".

Examples

```
## Not run:
data(TC1500264)
PivotTransformData(Data=TC1500264,GeneID=NULL,ExonID=NULL,
REMAPSplitFile="TC1500264_Gene_SplitFile.txt",Location="Output/",Name="TC1500264_Pivot")

REIDSFunction(ASPSR=c(), Indices="Output/TC1500264_LineIndex.csv",
DataFile="Output/TC1500264_Pivot.csv",nsim=50,informativeCalls=FALSE,
Summarize=c("WeightedAll","EqualAll"),
rho=0.5,Low_AllSamples=c(),Groups=list(c(1:3),c(4:6)),Location="Output",Name="TC1500264")

## End(Not run)
```

---

reidsfunction_genebygene

"reidsfunction_genebygene"

Description

The reidsfunction_genebygene performs the REIDS model and is an internal function of the REIDSFunction. The function calls on the pivot transformed .csv file and transforms the read lines into a data frame on which the REIDS model is performed.

Usage

```
reidsfunction_genebygene(file_name, file_pos, line_length, ASPSR = c(),
  nsim = 1000, informativeCalls = TRUE, Summarize = FALSE, rho = 0.5,
  Low_AllSamples = c(), Location = NULL, Name = "REIDS")
```

Arguments

file_name	The name of the pivot transformed .csv file.
file_pos	The position in the file where to start reading.
line_length	The length of the line to read.
ASPSR	A vector with alternatively spliced probe sets which are taken out of the analysis and summarization. This is useful if Summarize is "WeightedConst" and/or "EqualConst".

nsim	The number of iterations to perform. Defaults to 1000.
informativeCalls	Logical. Should the I/NI calls method be perform before applying the REIDS model?
Summarize	A character vector specifying wich summarization method is to be performed. The choices are "EqualAll", "WeightedAll", "EqualConst" and "WeightedConst". The former two use all probe sets while the latter use only the constituitive probe sets. Summarization on the constistuitive probe sets will only be performed if ASPSR is specified.
rho	The threshold for filtering in the I/NI calls method. Probesets with scores higher than rho are kept.
Low_AllSamples	A character vector containing the probe sets which are not DABG in all samples.
Location	A character string indication the place where the output should be saved.
Name	A name for the output to be saved at Location. Defaults to "REIDS".

Value

The functions writes the obtained information to .txt files: "Name_INICalls.txt", Name_ExonScores.txt", "Name_ArrayScores.txt" and the summarized values distributed across "Name_WeightedAll.txt", "Name_EqualAll.txt", "Name_WeightedConst.txt" and "Name_EqualConst.txt".

REIDSFunction_HPCVersion *"REIDS_HPCVersion"*

Description

The REIDS_ClusterVersion performs the REIDS model and was adapted for use on a HPC cluster. This function should be used with the REIDS_HPCVersion.R file and REIDS_HPCVersion.pbs script in the documentation folder of the package. After running this function on the cluster, the output files should be binded together with the CreateOutput function.

Usage

```
REIDSFunction_HPCVersion(geneID, geneData, ASPSR = c(), nsim = 1000,
  informativeCalls = TRUE, Summarize = FALSE, rho = 0.5,
  Low_AllSamples = c())
```

Arguments

geneID	The gene ID
geneData	The data with as rows the probesets and as columns the samples. Note that the first column should contain the gene IDs and the second column the exon IDs
ASPSR	A vector with alternatively spliced probe sets which are taken out of the analysis and summarization. This is useful if Summarize is "WeightedConst" and/or "EqualConst".

nsim	The number of iterations to perform. Defaults to 1000.
informativeCalls	Logical. Should the I/NI calls method be perform before applying the REIDS model?
Summarize	A character vector specifying wich summarization method is to be performed. The choices are "EqualAll", "WeightedAll", "EqualConst" and "WeightedConst". The former two use all probe sets while the latter use only the constituitive probe sets. Summarization on the constistuitive probe sets will only be performed if ASPSR is specified.
rho	The threshold for filtering in the I/NI calls method. Probesets with scores higher than rho are kept.
Low_AllSamples	A character vector containing the probe sets which are not DABG in all samples.

Value

A .RData file will be saved for each gene with the elements returned by the iniREIDS and REIDS functions. The outputs can be bound together by CreateOutput.

`REIDSJunctionAssessment_HPCVersion`
REIDSJunctionAssessment_HPCVersion

Description

REIDSJunctionAssessment_HPCVersion is the HPC version of REIDSJunctionAssessment. This function should be used with the REIDSJunctionAssessment_HPCVersion.R file and REIDSJunctionAssessment_HPCVersion.pbs script in the documentation folder of the package. After running this function on the cluster, the output files should be binded together with the CreateOutput function.

Usage

```
REIDSJunctionAssessment_HPCVersion(geneID, DataS = DataS, ASPSR = ASPSR,
  Juninfo = "User", JAnnotI, JAnnot = NULL, EandTrAnnotI = NULL,
  EandTrAnnot = NULL, PartiallyAnnotated, positionData = NULL,
  transcriptData = NULL, Groups = list(), Low_AllSamples = c(),
  Low_GSamples = c(), Plot = FALSE)
```

Arguments

geneID	The gene ID.
DataS	The data with as rows the probesets and as columns the samples. Note that the first column should contain the gene IDs and the second column the exon IDs
ASPSR	The AS probe sets as identified by ASExons.

Juninfo	A parameter specifying wether the annotations are user of Ensembl defined. If JunInfo is "User" (default) the annotations provided in EandTrAnnot are used. If JunInfo is "Ensembl" the annotations in EandTrAnnot are used to set up tje junction associations but the gene name and position in transcriptData and positionData are used to connect with the Ensembl data base and retrieve corresponding information.
JAnnotI	The file name with line indices for the junction associations.
JAnnot	The file name with the junction associations.
EandTrAnnotI	The file name with line indices for the exon and isoform annotations.
EandTrAnnot	The file name with the exon and isoform annotations.
PartiallyAnnotated	Logical. Should the exon annotations with partially annotated probe sets still be included? If FALSE, these are excluded. If TRUE, these are included. Default is FALSE.
positionData	The file with the chromosome start and ends for the probe sets. Only needed in JunInfo=Ensembl.
transcriptData	The file with gene name of the transcripts. Only needed in JunInfo=Ensembl.
Groups	A list with elements speficifing the columns of the data in each group.
Low_AllSamples	A character vector containing the probe sets which are not DABG in all samples.
Low_GSamples	A list with a character vector per group containing the probe sets which are not DABG in that group.
Plot	Should a plot of the gene model be made?

Details

The plot is produced by the arcplot function of the arcdiagram package (https://github.com/gastonstat/arcdiagram)

Value

A .RData file will be saved for each gene with the four elements returned REIDSJunctionAssessment function. The outputs can be bound together by CreateOutput.

REIDSmodel_intern *"REIDSmodel_intern"*

Description

The function is part of the larger REIDS function and performs the REIDS model on the data for a single gene.

Usage

```
REIDSmodel_intern(SubgeneData, nsim = 1000)
```

Arguments

SubgeneData	A subset of the data. Particularly, a subset of the data corresponding to one gene.
nsim	The number of iterations to perform. Defaults to 1000.

Value

A list with 2 items per gene. The first item is the exon scores of the corresponding probesets and the second contains a data frame with the array scores of the exons across the samples. If the iniREIDS model was performed. The items will be added to the previously made list.

REIDS_Analysis *REIDS_Analysis*

Description

The REIDS_Analysis is a wrapper function for the REIDSFunction, the ASExon function, the REIDS_JunctionAssesment function and the REIDS_IsoformAssesment function.

Usage

```
REIDS_Analysis(geneIDs, Indices, DataFile, nsim = 5000,
  informativeCalls = FALSE, Summarize = TRUE, rho = 0.5,
  Exonthreshold = 0.5, significancelevel = 0.05, Groups, paired = FALSE,
  Low_AllSamples = c(), Low_GSamples = c(), Juninfo = "User",
  JAnnotI = NULL, JAnnot = NULL, EandTrAnnotI = NULL,
  EandTrAnnot = NULL, PartiallyAnnotated = FALSE, positionData = NULL,
  transcriptData = NULL, Location = "Output", Name = "REIDSAnalysis")
```

Arguments

geneIDs	A vector with the geneIDs to analyze.
Indices	The .csv file created by Line_Indexer.py which contains indices for every gene.
DataFile	The .csv file created by PivotTransformation.
nsim	The number of iterations to perform. Defaults to 1000.
informativeCalls	Logical. Should the I/NI calls method be perform before applying the REIDS model?
Summarize	A character vector specifying the wich summarization method to be performed. The choices are using "EqualAll", "WeightedAll", "EqualConst", "WeightedConst". The former two use all probe sets while the latter to use only the consituitive probe sets. Summarization on the consistuitive probe sets will only be performed if ASPSR is specified.
rho	The threshold for filtering in the I/NI calls method. Probesets with scores higher than rho are kept.

`Exonthreshold` The exon score threshold to be maintained. If not NULL, probesets with an exon score lower than this value are not considered further and the p-values will be adjusted for multiplicity after testing. If NULL, all probesets are considered and a multiplicity correction is not performed.

`significancelevel` The significance level to be maintained on the p-values. The filtering on the significance is conducted only if an Exonthreshold is specified and the p-value are adjusted for multiplicity.

`Groups` A list with elements specifying the columns of the data in each group.

`paired` Logical. Are the groups paired? If TRUE the mean paired differences are calculated and tested whether these are significantly different from zero or not.

`Low_AllSamples` A character vector containing the probe sets which are not DABG in all samples.

`Low_GSamples` A list with a character vector per group containing the probe sets which are not DABG in that group.

`Juninfo` A parameter specifying wether the annotations are user of Ensembl defined. If JunInfo is "User" (default) the annotations provided in EandTrAnnot are used. If JunInfo is "Ensembl" the annotations in EandTrAnnot are used to set up tje junction associations but the gene name and position in transcriptData and positionData are used to connect with the Ensembl data base and retrieve corresponding information.

`JAnnotI` The file name with line indices for the junction associations.

`JAnnot` The file name with the junction associations.

`EandTrAnnotI` The file name with line indices for the exon and isoform annotations.

`EandTrAnnot` The file name with the exon and isoform annotations.

`PartiallyAnnotated` Logical. Should the exon annotations with partially annotated probe sets still be included? If FALSE, these are excluded. If TRUE, these are included. Default is FALSE.

`positionData` The file with the chromosome start and ends for the probe sets. Only needed in JunInfo=Ensembl.

`transcriptData` The file with gene name of the transcripts. Only needed in JunInfo=Ensembl.

`Location` A character string indication the place where the outputs are saved. Defaults to Output.

`Name` A character string with the name of the ouput file. Defaults to "REIDSAnalysis".

Value

The output will be written to each of the corresponding .txt files of the called upon functions.

Examples

```
## Not run:
data(TC1500264)
PivotTransformData(Data=TC1500264,GeneID=NULL,ExonID=NULL,
REMAPSplitFile="TC1500264_Gene_SplitFile.txt",Location="Output/",Name="TC1500264_Pivot")
```

```
REIDS_Analysis(Indices="Output/TC1500264_LineIndex.csv",DataFile="Output/TC1500264_Pivot.csv",
nsim=100,informativeCalls=FALSE,Summarize=c("WeightedAll","EqualAll","WeightedConst","EqualConst"),
rho=0.5,Exonthreshold=0.5,significancelevel=0.05,Groups=Groups,paired=FALSE,Low_AllSamples=c()
,Low_GSamples=c(),Juninfo="User",JAnnotI=NULL,JAnnot=NULL,EandTrAnnotI="Output/REMAP_Indices.txt",
EandTrAnnot="Output/HJAY_REMAP.txt",positionData=NULL,transcriptData=NULL,
Location="OutputREIDSAnalysis",Name="TC1500264")

## End(Not run)
```

`REIDS_IsoformAssesment` *REIDS_IsoformAssesment*

Description

The REIDS_IsoformAssesment is an experimental function to analyze the isoform information based on the exon level values and the isoform composition.

Usage

```
REIDS_IsoformAssesment(geneIDs, IsoformInfo, ExonLevel, Groups, paired,
  Location = NULL, Name = "REIDSIsoforms")
```

Arguments

`geneIDs`	A vector with the geneIDs to analyze.
`IsoformInfo`	The path to the Composition file created by REIDS_JunctionAssesment or CreateOutput.
`ExonLevel`	The path to the ExonLevel.txt file
`Groups`	A list with elements speficifing the columns of the data in each group.
`paired`	Logical. Are the groups paired samples?
`Location`	A character string indication the place where the outputs are saved.
`Name`	A character string with the name of the ouput file. Defaults to "REIDSIsoforms".

Value

The function returns three files. The first file has name "Name_IsoformIndication.txt" and contains an assesment of the relative expression levels of the isoforms. A second file,"Name_ExonTesting.txt", shows information regarding the differential expression of exon. A final file,"Name_PossibelDEIsoforms" lists isoforms which might be differentially expressed between the groups.

REIDS_JunctionAssesment

REIDS_JunctionAssessment

Description

The REIDS_JunctionAssessment functions assess identified AS exons based on their 5'end and 3'end and exclusion junction support.

Usage

```
REIDS_JunctionAssesment(Indices, DataFile, ASProbeSets = c(),
  Juninfo = "User", JAnnotI = NULL, JAnnot = NULL, EandTrAnnotI = NULL,
  EandTrAnnot = NULL, PartiallyAnnotated = FALSE, positionData = NULL,
  transcriptData = NULL, Groups = list(c(3, 4, 5), c(6, 7, 8)),
  Low_AllSamples = c(), Low_GSamples = c(), Location = NULL,
  Name = "REIDS_Jun")
```

Arguments

`Indices`	The .csv file created by Line_Indexer.py which contains indices for every gene in geneIDs.
`DataFile`	The .csv file created by PivotTransformation.
`ASProbeSets`	The AS probe sets as identified by ASExons.
`Juninfo`	A parameter specifying wether the annotations are user of Ensembl defined. If JunInfo is "User" the annotations provided in EandTrAnnot are used. If JunInfo is "Ensembl" the annotations in EandTrAnnot are used to set up tje junction associations but the gene name and position in transcriptData and positionData are used to connect with the Ensembl data base and retrieve corresponding information.
`JAnnotI`	The file name with line indices for the junction associations.
`JAnnot`	The file name with the junction associations.
`EandTrAnnotI`	The file name with line indices for the exon and isoform annotations.
`EandTrAnnot`	The file name with the exon and isoform annotations.
`PartiallyAnnotated`	Logical. Should the exon annotations with partially annotated probe sets still be included? If FALSE, these are excluded. If TRUE, these are included. Default is FALSE.
`positionData`	The file with the chromosome start and ends for the probe sets. Only needed in JunInfo=Ensembl.
`transcriptData`	The file with gene name of the transcripts. Only needed in JunInfo=Ensembl.
`Groups`	A list with elements speficifing the columns of the data in each group.
`Low_AllSamples`	A character vector containing the probe sets which are not DABG in all samples.

Low_GSamples	A list with a character vector per group containing the probe sets which are not DABG in that group.
Location	A character string indication the place where the outputs are saved.
Name	A character string with the name of the ouput file. Defaults to "REIDS_Jun".

Value

The function returns four files. The first file has name "Name_ASInfo.txt" and contains a line per probe set. It shows the reached decisionregarding the probe set (Const/AS/not DABG),its linking and exclusion junctions, the fold change, the AS type and its annotated exons. The secondfile, "Name_Compositions.txt", is a list of all found transcripts for a particular TC ID. The third file,"Name_GroupTranscripts.txt" indicates whether a specific transcript is present or absent in a group. The fourth file "Name_NovelConnections.txt" contains junctions which are showing an undocumented connection between probe sets.

Examples

```
## Not run:
data(TC1500264)
PivotTransformData(Data=TC1500264,GeneID=NULL,ExonID=NULL,
REMAPSplitFile="TC1500264_Gene_SplitFile.txt",Location="Output/",Name="TC1500264_Pivot")
REIDSFunction(ASPSR=c(), Indices="Output/TC1500264_LineIndex.csv",
DataFile="Output/TC1500264_Pivot.csv",nsim=50,informativeCalls=FALSE,Summarize=
c("WeightedAll","EqualAll"),rho=0.5,Low_AllSamples=c(),Groups=list(c(1:3),c(4:6)),
Location="Output",Name="TC1500264")

TC1500264_1vs2=ASExons(ExonScores="Output/TC1500264_ExonScores.txt",ArrayScores=
"Output/TC1500264_ArrayScores.txt",Exonthreshold=0.5,Groups=list(c(1:3),c(4:6)),paired=FALSE,
significancelevel=0.05)

ASPSR_PSR=TC1500264_1vs2[which(round(TC1500264_1vs2$ExonScore,2)>=0.50&
TC1500264_1vs2$adj.p.value<0.05),2]

REIDS_JunctionAssesment(Indices="Output/TC1500264_LineIndex.csv",DataFile=
"Output/TC1500264_Pivot.csv",ASProbeSets=ASPSR_PSR,Juninfo="User",JAnnotI=NULL,
JAnnot=NULL,EandTrAnnotI="Output/REMAP_Indices.txt",EandTrAnnot=
"Output/HJAY_REMAP.txt",positionData=NULL,transcriptData=NULL,Groups=
list(c(1:3),c(4:6)),Low_AllSamples=c(),Low_GSamples=c(),Location="Output",
Name="TC1500264")

## End(Not run)
```

REMAP_SplitProbeSets *"REMAP_SplitProbeSets"*

Description

The REMAP_SplitProbeSets function converts the original data frame to a data frame based on the retrieved REMAP annotation.

Usage

```
REMAP_SplitProbeSets(Data, REMAPSplitFile, NotAnnotated = FALSE,
  Location = NULL, Name = "REMAP")
```

Arguments

Data	The data frame to be transformed.
REMAPSplitFile	The name of the file with the REMAP information regarding the split of the probe sets if the TC ID is annotated to mutiple genes.
NotAnnotated	Logical. Should the probe sets which are not annotated to a gene still be included? If FALSE, these are excluded. If TRUE, these are included. Default is FALSE.
Location	The location where the file should be saved. If NULL, the object is returned to the user. Otherwise, a file with the specified name is created.
Name	The name of the output file. Defaults to "REMAP".

Value

A data frame with similar information as the data frame specified in Data with an altered gene ID (first) column in order to match the REMAP annotation.

Examples

```
## Not run:
data(TC1500264)

REMAPTest=REMAP_SplitProbeSets(Data=TC1500264,REMAPSplitFile=
"TC1500264_Gene_Split.txt")

## End(Not run)
```

SpliceIndex *"SpliceIndex"*

Description

The SpliceIndex function computes the ratio of the splice indices of defined groups. Further, it performs the SI algorithm if the length of the groups is two and the MiDAS algorithm is more groups are specified. Both algorithms are implemented as defined by Affymetrix.

Usage

```
SpliceIndex(GeneData, ExonData, InformativeExons = NULL,
  Groups = list(group1 = NULL, group2 = NULL), paired = FALSE,
  significancelevel = NULL)
```

Arguments

GeneData	The microarray data summarized at gene level.
ExonData	The microarray data summarized at exon level.
InformativeExons	A character vector of exon IDs. As for the REIDS model probesets are filtered out by I/NI calls model and later on exon score, the remaining exons can be specified here. Only these shall be considered in the FIRMA analysis to make the results between REIDS and FIRMA more comparable
Groups	A list with the groups (columns) of interest in the data.
paired	Logical. Are the groups paired? only used if two groups are present.
significancelevel	If specified, filtering is conducted on the p-values.

Details

Given the gene level and exon level summarized data, the splice index method for the detection of alternative splicing is performed. The first step is to normalize the exon data by taking the ratio with the gene level data. These values are referred to as the splice indices. If only two groups are specified, the ratio of their splice indices is taken as a measure for alternative splicing. The more the ratio deviates from zero, the more there is an indication of alternative splicing. A t-test is conducted on the splice indices of the two groups to test their difference. If more than two groups are specified, an ANOVA model is fitted on the splice indices to discover with an F-test whether there is a difference between the groups somewhere. If a vector of informative exons is given to the function, only these are considered for the analysis. Finally, the p-values are adjusted for multiplicity and if a significance level is specified only the significant p-valuesare kept in the data frame.

Value

A data frame wiith one line per exon. The columns conatin the gene ID, the exon ID, the ratio of the splice indices if two groups are present, a t- or F-statitic, a p-value and an adjusted p-value.

Examples

```
data(TC12000010_ExonLevel)
data(TC12000010_GeneLevel)
SI_Test=SpliceIndex(GeneData=TC12000010_GeneLevel,ExonData=TC12000010_ExonLevel
,InformativeExons=NULL,Groups=list(group1=c(1:9),group2=c(10:18)),
paired=FALSE,significancelevel=NULL)
```

TC12000010 *Example data set*

Description

An example data set containing the expression values of the probe sets of transcript cluster TC12000010.

Usage

```
data(TC12000010)
```

Format

An object of class "data.frame".

Examples

```
data(TC12000010)
```

TC12000010_ExonLevel *Exon level data*

Description

An example data set containing the summarized expression values on the exon level of the probe sets of TC12000010.

Usage

```
data(TC12000010_ExonLevel)
```

Format

An object of class "data.frame".

Examples

```
data(TC12000010_ExonLevel)
```

TC12000010_GeneLevel *Gene level data*

Description

An example data set containing the summarized expression value of TC12000010.

Usage

```
data(TC12000010_GeneLevel)
```

Format

An object of class "data.frame".

Examples

```
data(TC12000010_GeneLevel)
```

TC12000010_positions *Positions*

Description

Genome location of the probe sets of TC12000010

Usage

```
data(TC12000010_positions)
```

Format

An object of class "data.frame".

Examples

```
data(TC12000010_positions)
```

TC12000010_REIDS_Output

REIDS Output

Description

An example data set containing the obtained REIDS values of TC12000010.

Usage

```
data(TC12000010_REIDS_Output)
```

Format

An object of class "data.frame".

Examples

```
data(TC12000010_REIDS_Output)
```

TC12000010_transcript.clusters

Transcript Clusters

Description

Transcript cluster compositions of TC12000010

Usage

```
data(TC12000010_transcript.clusters)
```

Format

An object of class "data.frame".

Examples

```
data(TC12000010_transcript.clusters)
```

TC1500264 *Example data set*

Description

An example data set containing the expression values of the probe sets of transcript cluster TC1500264.

Usage

```
data(TC1500264)
```

Format

An object of class `"data.frame"`.

Examples

```
data(TC1500264)
```

TranscriptsPlot *"TranscriptsPlot "*

Description

The TranscriptsPlot function plots the known Ensemble transcript isoform composition plots.

Usage

```
TranscriptsPlot(trans, positions, transcriptinfo, display.probesets = TRUE,
  Data, Groups = list(), Start = NULL, Stop = NULL, Highlight = NULL)
```

Arguments

`trans`	The TC ID of the transcript to be plotted.
`positions`	A table with the start and stop positions of the probe sets.
`transcriptinfo`	A table with the transcript information of the TC ID.
`display.probesets`	Logical. Should the probe sets be shown?
`Data`	The exon level summarized data.
`Groups`	A list with the groups (columns) of interest in the data.
`Start`	Specify a specific start point on in the genome.
`Stop`	Specify a specific stop point on in the genome.
`Highlight`	A character string specifying a probe set to be highlighted in the transcript composition.

Examples

```
## Not run:
data(positions_36)
data(transcript.clusters.NetAffx.36)
data(TC12000010_ExonLevel)
TranscriptsPlot(trans="TC12000010", display.probesets = TRUE,Data=TC12000010_ExonLevel,
Groups=list(c(10:18),c(19:27)),Start=NULL,Stop=NULL,Highlight="PSR12000150")

## End(Not run)
```

---

trim *"trim"*

Description

The trim function trims a long character string on gene annotation to a short single gene annotation.

Usage

```
trim(s)
```

Arguments

s	Character vector to trim.

---

xynodes *X or Y coordinates of node locations*

Description

Gives axis locations of each node

Usage

```
xynodes(num_nodes, aux_ord, labels)
```

Arguments

num_nodes	number of nodes
aux_ord	vector with the index number for ordering the nodes
labels	optional string vector with labels for the nodes

Index