**Package ‘MARVEL’**

March 31, 2021

**Title**  Revealing Splicing Dynamics at Single-Cell Resolution

**Version**  1.1.0

**Description**  
Alternative splicing represents an additional and underappreciated layer of complexity underlying gene expression profiles. Nevertheless, there remains hitherto a paucity of softwares to investigate splicing dynamics at single-cell resolution. ‘MARVEL’ quantifies percent spliced-in (PSI) values for the five primary exon-level splicing events, namely skipped-exon (SE), mutually exclusive exons (MXE), retained intron (RI), alternative 5’ and 3’ splice sites (A5SS and A3SS, respectively). Additionally, ‘MARVEL’ performs differential splicing analysis to identify splicing events whose PSI distribution differ between groups of cells. Finally, ‘MARVEL’ models the PSI distribution for each event as a beta distribution and categorises each distribution into modalities (inspired by Song (2017) <doi:10.1016/j.molcel.2017.06.003>.

**Imports**  AnnotationDbi (>= 1.48.0), dplyr (>= 1.0.1), FactoMineR (>= 2.3), factoextra (>= 1.0.7), fitdistrplus (>= 1.1-1), ggplot2 (>= 3.3.2), GO.db (>= 3.10.0), GOstats (>= 2.52.0), methods, plyr (>= 1.8.4), kableExtra (>= 1.3.1), org.Hs.eg.db (>= 3.10.0), parallel, stringr (>= 1.4.0)

**Encoding**  UTF-8

**Author**  Sean Wen

**Maintainer**  Sean Wen <sean.wenwx@gmail.com>

**License**  GPL-3

**RoxygenNote**  6.1.1

**Suggests**  knitr, rmarkdown

**VignetteBuilder**  knitr

**ByteCompile**  true

**NeedsCompilation**  no

**Repository**  CRAN

**Date/Publication**  2021-03-31 10:40:02 UTC
R topics documented:

AssignModality .......................................................... 2
BioPathways ................................................................. 4
CompareValues .............................................................. 5
CompareValues.Exp ....................................................... 6
CompareValues.PSI ...................................................... 7
ComputePSI ................................................................. 8
ComputePSI.A3SS .......................................................... 10
ComputePSI.A5SS .......................................................... 11
ComputePSI.MXE ............................................................ 12
ComputePSI.RI .............................................................. 13
ComputePSI.SE .............................................................. 14
CreateMarvelObject ...................................................... 15
IsoSwitch ................................................................. 17
ModalityChange ........................................................... 18
PlotValues ................................................................. 20
PlotValues.Exp ............................................................. 22
PlotValues.PSI .............................................................. 23
PropModality ............................................................... 24
PropModality.Bar .......................................................... 25
PropModality.Doughnut .................................................. 27
RunPCA ................................................................. 28
RunPCA.Exp ................................................................. 29
RunPCA.PSI ................................................................. 30

Index 32

AssignModality  Modality Assignment

Description

AssignModality assigns modalities to each splicing event for a specified group of cells.

Usage

AssignModality(MarvelObject, cell.type, n.cells, sigma.sq, bimodal.adjust, seed)

Arguments

MarvelObject  S3 object generated from ComputePSI function.
cell.type  Character string. To indicate which group of cells to analyse. Group name should match that in cell.type column of $SplicePheno slot.
n.cells  Numeric value. The minimum no. of cells expressing the splicing event for the event to be included for modality assignment.
### AssignModality

**sigma.sq** Numeric value. The variance threshold below which the included/excluded modality will be defined as primary sub-modality, and above which it will be defined as dispersed sub-modality.

**bimodal.adjust** Logical. When set to TRUE, MARVEL will identify false bimodal modalities and reassign them as included/excluded modality.

**seed** Numeric value. Ensure the `fitdist` function returns the same values for alpha and beta parameters each time this function is executed using the same random number generator.

### Details

This function assigns modalities to each splicing event for a specified group of cells. The five main modalities are included, excluded, bimodal, middle, and multimodal (inspired by Song (2017)). MARVEL further stratifies included and excluded modalities into primary and dispersed sub-modalities depending on the variance (dispersion).

### Value

An object of class S3 containing all the original slots as inputted by the user in addition to one new data frame. This data frame is store in `$Modality$Results` slot. This data frame contains the modality assignment for each splicing event and is saved into the `Modality` slot. `modality` column reflects the five main modalities, i.e. included, excluded, bimodal, middle, and multimodal. `modality.var` column additional stratifies included and excluded modalities into primary and dispersed sub-modalities. `modality.bimodal.adj` column identifies and re-categorizes false bimodals into included or excluded modalities when `bimodal.adjust` is set to TRUE.

### Author(s)

Sean Wen <sean.wenwx@gmail.com>

### Examples

```r
marvel <- readRDS(system.file("extdata/Data", "MarvelObject.rds", package="MARVEL"))

marvel <- AssignModality(MarvelObject=marvel,
                          cell.type=c("iPSC", "Endoderm"),
                          n.cells=25,
                          sigma.sq=0.001,
                          bimodal.adjust=TRUE,
                          seed=1)

marvel$Modality$Results[1:5, ]
```
BioPathways

Gene Ontology Analysis

Description

BioPathways performs gene ontology analysis on genes that are differentially spliced.

Usage

BioPathways(MarvelObject, psi.de.sig, method.adjust, min.genes, p.val.adj.return, plot.top.n)

Arguments

MarvelObject S3 object generated from CompareValues function.
psi.de.sig Numeric value. Adjusted p-value below which the splicing event is considered differentially spliced and included for isoform switching analysis.
method.adjust Character string. Adjust p-values for multiple testing. Options available as per p.adjust function.
min.genes Numeric value. Number of differentially spliced genes required for analysis.
p.val.adj.return Numeric value. Return pathways with adjusted p-value below this value.
plot.top.n Numeric value. Plot these top most significant pathways.

Details

This function performs gene ontology analysis on genes that are differentially spliced to identify significantly regulated biological pathways.

Value

An object of class S3 containing all the original slots as inputted by the user in addition to two new slots. $DE$BioPathways contains the pathways significantly enriched among differentially spliced genes as specified in p.val.adj.return. $DE$BioPathwaysPlot contains the plot of top most significant pathways ranked by adjusted p-values as specified in plot.top.n.

Author(s)

Sean Wen <sean.wenwx@gmail.com>
**Examples**

```r
marvel <- readRDS(system.file("extdata/Data", "MarvelObject.rds", package="MARVEL"))

marvel <- BioPathways(MarvelObject=marvel,
                       psi.de.sig=0.05,
                       method.adjust="fdr",
                       min.genes=50,
                       p.val.adj.return=0.05,
                       plot.top.n=10)

marvel$DE$BioPathways
marvel$DE$BioPathwaysPlot
```

**Description**

`CompareValues` performs differentially splicing and gene expression analysis between 2 groups of cells.

**Usage**

```
CompareValues(MarvelObject, cell.types, n.cells, method, method.adjust, level)
```

**Arguments**

- **MarvelObject**: S3 object generated from `ComputePSI` function.
- **cell.types**: Character string. To indicate which 2 groups of cells that will be used for differential splicing analysis. Group names should match those in `cell.type` column of `$SplicePheno` slot.
- **n.cells**: Numeric value. The minimum no. of cells expressing the splicing event for the event to be included for differential splicing analysis.
- **method**: Character string. Statistical test to compare the 2 groups of cells. "ks", "wilcox", and "t.test" for Kolmogorov-Smirnov, Wilcoxon, and t-test, respectively. We advice "ks" for PSI comparison while "wilcox" or "t.test" for gene expression comparison.
- **method.adjust**: Character string. Adjust p-values for multiple testing. Options available as per `p.adjust` function.
- **level**: Character string. Indicate "splicing" or "gene" for differential splicing or gene expression analysis, respectively

**Details**

This function compares the percent spliced-in (PSI) and gene expression values between 2 groups of cells.
**Value**

An object of class S3 containing all the original slots as inputted by the user in addition to one new slot. When `level` set to "splicing" or "gene", results are returned to `$DE$PSI` or `$DE$Exp` slot, respectively.

**Author(s)**

Sean Wen <sean.wenwx@gmail.com>

**Examples**

```r
marvel <- readRDS(system.file("extdata/Data", "MarvelObject.rds", package="MARVEL"))

marvel <- CompareValues(MarvelObject=marvel,
  cell.types=c("iPSC", "Endoderm"),
  n.cells=25,
  method="t.test",
  method.adjust="fdr",
  level="splicing")

marvel$DE$PSI[1:5, ]
```

---

**CompareValues.Exp  Differential Gene Expression Analysis**

**Description**

CompareValues.Exp performs differentially gene expression analysis between 2 groups of cells.

**Usage**

`CompareValues.Exp(MarvelObject, cell.types, n.cells, method, method.adjust)`

**Arguments**

- **MarvelObject**: S3 object generated from `ComputePSI` function.
- **cell.types**: Character string. To indicate which 2 groups of cells that will be used for differential splicing analysis. Group names should match those in `cell.type` column of `$SplicePheno` slot.
- **n.cells**: Numeric value. The minimum no. of cells expressing the splicing event for the event to be included for differential splicing analysis.
- **method**: Character string. Statistical test to compare the 2 groups of cells. "ks", "wilcox", and "t.test" for Kolmogorov-Smirnov, Wilcox, and t-test, respectively. We advice "ks" for PSI comparison while "wilcox" or "t.test" for gene expression comparison.
- **method.adjust**: Character string. Adjust p-values for multiple testing. Options available as per `p.adjust` function.
Details

This function compares the gene expression values between 2 groups of cells.

Value

An object of class S3 containing all the original slots as inputted by the user in addition to one new slot named $DE$Gene

Author(s)

Sean Wen <sean.wenwx@gmail.com>

Examples

```r
marvel <- readRDS(system.file("extdata/Data", "MarvelObject.rds", package="MARVEL"))
marvel <- CompareValues.Exp(MarvelObject=marvel, 
cell.types=c("iPSC", "Endoderm"), 
n.cells=25, 
method="t.test", 
method.adjust="fdr"
)
marvel$DE$Gene[1:5, ]
```

Description

CompareValues.PSI performs differentially splicing analysis between 2 groups of cells.

Usage

```r
CompareValues.PSI(MarvelObject, cell.types, n.cells, method, method.adjust)
```

Arguments

- **MarvelObject**: S3 object generated from ComputePSI function.
- **cell.types**: Character string. To indicate which 2 groups of cells that will be used for differential splicing analysis. Group names should match those in cell.type column of $SplicePheno$ slot.
- **n.cells**: Numeric value. The minimum no. of cells expressing the splicing event for the event to be included for differential splicing analysis.
- **method**: Character string. Statistical test to compare the 2 groups of cells. "ks", "wilcox", and "t.test" for Kolmogorov-Smirnov, Wilcox, and t-test, respectively. We advice "ks" for PSI comparison while "wilcox" or "t.test" for gene expression comparison.
method.adjust Character string. Adjust p-values for multiple testing. Options available as per p.adjust function.

Details
This function compares the percent spliced-in (PSI) values between 2 groups of cells.

Value
An object of class S3 containing all the original slots as inputted by the user in addition to one new slot named $DEPSI$

Author(s)
Sean Wen <sean.wenwx@gmail.com>

Examples

```r
marvel <- readRDS(system.file("extdata/Data", "MarvelObject.rds", package="MARVEL"))
marvel <- CompareValues.PSI(MarvelObject=marvel,
cell.types=c("iPSC", "Endoderm"),
n.cells=25,
method="t.test",
method.adjust="fdr"
)
marvel$DEPSI[1:5, ]
```

---

**ComputePSI**

**Compute Percent Spliced-in (PSI) Values**

Description
ComputePSI computes percent spliced-in (PSI) values for a specified splicing event type.

Usage

```r
ComputePSI(MarvelObject, CoverageThreshold, EventType,
IntronCountsFile = NULL, thread = NULL)
```

Arguments

- **MarvelObject** S3 object generated from CreateMarvelObject function.
- **CoverageThreshold** Numeric value. Coverage threshold below which the PSI of the splicing event will be censored, i.e. annotated as missing (NA). Coverage defined as the total number of reads supporting both included and excluded isoforms.
ComputePSI

**EventType**  
Character string. Indicate which splicing event type to calculate the PSI values for. Can take value "SE", "MXE", "RI", "A5SS", or "A3SS" which represents skipped-exon (SE), mutually-exclusive exons (MXE), retained-intron (RI), alternative 5' splice site (A5SS), and alternative 3' splice site (A3SS), respectively.

**IntronCountsFile**  
Data frame containing per base coverage of introns. Only applicable when EventType set to "RI". First column should be named coord.intron and indicate the per base intron position in the form of chr:position. Subsequent columns should contain the per base coverage for each sample. These counts can be detected using external softwares such as Bedtools etc..

**thread**  
Numeric value. Set number of threads.

**Details**

This function computes the PSI values for a splicing event as specified in the EventType argument. Splicing events provided in SpliceFeature data frame will first be cross-checked against the splice junctions provided in SpliceJunction data frame. Only events whose junctions are found in SpliceJunction are retained. The formula for computing PSI is the number of junction reads supporting the included isoform divided by the total number of reads supporting both included and excluded isoforms.

**Value**

An object of class S3 containing all the original slots as inputted by the user in addition to two new slots. $SpliceFeatureValidated contains the validated splicing event metadata. $PSI contains the computed PSI values for the validated splicing events.

**Author(s)**

Sean Wen <sean.wenwx@gmail.com>

**Examples**

```R
marvel <- readRDS(system.file("extdata/Data", "MarvelObject.rds", package="MARVEL"))

marvel <- ComputePSI(MarvelObject=marvel,  
CoverageThreshold=10,  
EventType="SE")

marvel$SpliceFeatureValidated$SE
marvel$PSI$SE[,1:5]
```
 Compute Alternative 3’ Splice Site (A3SS) Percent Spliced-in (PSI) Values

**Description**

ComputePSI.A3SS computes percent spliced-in (PSI) Alternative 3’ splice site (A3SS) splicing event.

**Usage**

```r
ComputePSI.A3SS(MarvelObject, CoverageThreshold)
```

**Arguments**

- **MarvelObject**  
  S3 object generated from `CreateMarvelObject` function.
- **CoverageThreshold**  
  Numeric value. Coverage threshold below which the PSI of the splicing event will be censored, i.e. annotated as missing (NA). Coverage defined as the total number of reads supporting both included and excluded isoforms.

**Details**

This function computes the PSI for each A3SS splicing event. Splicing events provided in SpliceFeature data frame will first be cross-checked against the splice junctions provided in SpliceJunction data frame. Only events whose junctions are found in SpliceJunction are retained. The formula for computing PSI is the number of junction reads supporting the included isoform divided by the total number of reads supporting both included and excluded isoforms.

**Value**

An object of class S3 containing all the original slots as inputted by the user in addition to two new slots. `$SpliceFeatureValidated$A3SS` contains the validated splicing event metadata. `$PSI$A3SS` contains the computed PSI values for the validated splicing events.

**Author(s)**

Sean Wen <sean.wenwx@gmail.com>

**Examples**

```r
marvel <- readRDS(system.file("extdata/Data", "MarvelObject.rds", package="MARVEL"))

marvel <- ComputePSI.A3SS(MarvelObject=marvel, CoverageThreshold=10)

marvel$SpliceFeatureValidated$A3SS
marvel$PSI$A3SS[,1:5]
```
### Description

ComputePSI.A5SS computes percent spliced-in (PSI) alternative 5' splice site (A5SS) splicing event.

### Usage

\[
\text{ComputePSI.A5SS}(\text{MarvelObject}, \text{CoverageThreshold})
\]

### Arguments

- **MarvelObject**: S3 object generated from `CreateMarvelObject` function.
- **CoverageThreshold**: Numeric value. Coverage threshold below which the PSI of the splicing event will be censored, i.e. annotated as missing (NA). Coverage defined as the total number of reads supporting both included and excluded isoforms.

### Details

This function computes the PSI for each A5SS splicing event. Splicing events provided in `SpliceFeature` data frame will first be cross-checked against the splice junctions provided in `SpliceJunction` data frame. Only events whose junctions are found in `SpliceJunction` are retained. The formula for computing PSI is the number of junction reads supporting the included isoform divided by the total number of reads supporting both included and excluded isoforms.

### Value

An object of class S3 containing all the original slots as inputted by the user in addition to two new slots. `$SpliceFeatureValidated$A5SS` contains the validated splicing event metadata. `$PSI$A5SS` contains the computed PSI values for the validated splicing events.

### Author(s)

Sean Wen <sean.wenwx@gmail.com>

### Examples

```r
marvel <- readRDS(system.file("extdata/Data", "MarvelObject.rds", package="MARVEL"))

marvel <- ComputePSI.A5SS(MarvelObject=marvel,
                           CoverageThreshold=10)

marvel$SpliceFeatureValidated$A5SS
marvel$PSI$A5SS[,1:5]
```
ComputePSI.MXE

**Description**

ComputePSI.MXE computes percent spliced-in (PSI) mutually exclusive exons (MXE) splicing event.

**Usage**

ComputePSI.MXE(MarvelObject, CoverageThreshold)

**Arguments**

- **MarvelObject**: S3 object generated from CreateMarvelObject function.
- **CoverageThreshold**: Numeric value. Coverage threshold below which the PSI of the splicing event will be censored, i.e. annotated as missing (NA). Coverage defined as the total number of reads supporting both included and excluded isoforms.

**Details**

This function computes the PSI for each MXE splicing event. Splicing events provided in SpliceFeature data frame will first be cross-checked against the splice junctions provided in SpliceJunction data frame. Only events whose junctions are found in SpliceJunction are retained. The formula for computing PSI is the number of junction reads supporting the included isoform divided by the total number of reads supporting both included and excluded isoforms.

**Value**

An object of class S3 containing all the original slots as inputted by the user in addition to two new slots. `$SpliceFeatureValidated$MXE` contains the validated splicing event metadata. `$PSI$MXE` contains the computed PSI values for the validated splicing events.

**Author(s)**

Sean Wen <sean.wenwx@gmail.com>

**Examples**

```r
marvel <- readRDS(system.file("extdata/Data", "MarvelObject.rds", package="MARVEL"))

marvel <- ComputePSI.MXE(MarvelObject=marvel,
                         CoverageThreshold=10)

marvel$SpliceFeatureValidated$MXE
marvel$PSI$MXE[,1:5]
```
ComputePSI.RI

Compute Retained-intron (RI) Percent Spliced-in (PSI) Values

Description

ComputePSI.RI computes percent spliced-in (PSI) retained intron (RI) splicing event.

Usage

ComputePSI.RI(MarvelObject, CoverageThreshold, IntronCountsFile, thread)

Arguments

MarvelObject S3 object generated from CreateMarvelObject function.
CoverageThreshold Numeric value. Coverage threshold below which the PSI of the splicing event will be censored, i.e. annotated as missing (NA). Coverage defined as the total number of reads supporting both included and excluded isoforms.
IntronCountsFile Data frame containing total coverage of introns. First column should be named coord.intron and indicate intron coordinates in the form of chr:start:end. Subsequent columns should contain the total intron coverage for each sample. These coverage counts can be detected using external softwares such as Bedtools etc..
thread Numeric value. Set number of threads.

Details

This function computes the PSI for each RI splicing event. Splicing events provided in SpliceFeature data frame will first be cross-checked against the splice junctions provided in SpliceJunction data frame. Only events whose junctions are found in SpliceJunction are retained. Formula for computing PSI is the normalized intron coverage divided by the total number of reads supporting both included and excluded isoforms. Normalized intron coverage is computed by taking the total coverage over the intronic region adjusted (divided) by the intron length.

Value

An object of class S3 containing all the original slots as inputted by the user in addition to two new slots. $SpliceFeatureValidated$RI contains the validated splicing event metadata. $PSI$RI contains the computed PSI values for the validated splicing events.

Author(s)

Sean Wen <sean.wenwx@gmail.com>
Examples

```r
path_to_file <- system.file("extdata/Data", "Counts_by_Region.txt",
  package="MARVEL")
df.intron.counts <- read.table(path_to_file, sep="\t", header=TRUE,
  stringsAsFactors=FALSE, na.strings="NA")

marvel <- readRDS(system.file("extdata/Data", "MarvelObject.rds", package="MARVEL"))

marvel <- ComputePSI.RI(MarvelObject=marvel,
  CoverageThreshold=10,
  IntronCountsFile=df.intron.counts,
  thread=1
)

marvel$SpliceFeatureValidated$RI
marvel$PSI$RI[,1:5]
```

---

**ComputePSI.SE**

**Compute Skipped-exon (SE) Percent Spliced-in (PSI) Values**

**Description**

ComputePSI.SE computes percent spliced-in (PSI) skipped-exon (SE) splicing event.

**Usage**

```r
ComputePSI.SE(MarvelObject, CoverageThreshold)
```

**Arguments**

- **MarvelObject**  
  S3 object generated from CreateMarvelObject function.
- **CoverageThreshold**  
  Numeric value. Coverage threshold below which the PSI of the splicing event will be censored, i.e. annotated as missing (NA). Coverage defined as the total number of reads supporting both included and excluded isoforms.

**Details**

This function computes the PSI for each SE splicing event. Splicing events provided in SpliceFeature data frame will first be cross-checked against the splice junctions provided in SpliceJunction data frame. Only events whose junctions are found in SpliceJunction are retained. The formula for computing PSI is the number of junction reads supporting the included isoform divided by the total number of reads supporting both included and excluded isoforms.

**Value**

An object of class S3 containing all the original slots as inputted by the user in addition to two new slots. `$SpliceFeatureValidated$SE` contains the validated splicing event metadata. `$PSI$SE` contains the computed PSI values for the validated splicing events.
CreateMarvelObject

Author(s)
Sean Wen <sean.wenwx@gmail.com>

Examples

marvel <- readRDS(system.file("extdata/Data", "MarvelObject.rds", package="MARVEL"))

marvel <- ComputePSI.SE(MarvelObject=marvel, CoverageThreshold=10)

marvel$SpliceFeatureValidated$SE
marvel$PSI$SE[,1:5]

Description

CreateMarvelObject creates an S3 object named Marvel for downstream analysis.

Usage

CreateMarvelObject(SplicePheno = NULL, SpliceJunction = NULL, SpliceFeature = NULL, SpliceFeatureValidated = NULL, PSI = NULL, GenePheno = NULL, GeneFeature = NULL, Exp = NULL)

Arguments

SplicePheno Data frame containing sample metadata. This object should consist of at least 2 columns. Mandatory columns are sample.id and cell.type. sample.id indicates all unique sample IDs. cell.type indicates the group names of each sample. Additional columns, if present, may contain additional details of each sample such sequencing QC details etc..

SpliceJunction Data frame containing splice junction counts. First column should be named coord.intron and indicate the splice junction position in the form of chr:start:end. Subsequent columns should contain the splice junction counts for each sample. These junctions can be detected using external softwares such as STAR, featureCounts etc..

SpliceFeature List containing splicing event metadata. Each element in the list is a data frame for each splicing event type, i.e. SE, MXE, RI, A5SS, and A3SS. Names of each element should reflect the splicing event type. Mandatory columns in each data frame are tran_id and gene_id. tran_id indicates all the splicing event coordinates. These events can be detected using external softwares such as rMATS, MISO, BRIE etc.. Other columns, if present, may contain additional details of each splicing event such as gene name, gene type etc.
CreateMarvelObject

SpliceFeatureValidated
List containing validated splicing event metadata. Required when SpliceFeature and SpliceJunction not specified. Each element in the list is a data frame for each splicing event type, i.e. SE, MXE, RI, A5SS, and A3SS. Names of each element should reflect the splicing event type. Mandatory columns in each data frame are tran_id and gene_id. tran_id indicates all the splicing event coordinates. These events can be detected using external softwares such as rMATS, MISO, BRIE etc.. Additional columns, if present, may contain additional details of each splicing event such as gene name, gene type etc..

PSI
Data frame containing pre-computed PSI values. Required when SpliceFeature and SpliceJunction not specified. The first column should be named tran_id and second column onwards should be sample names containing the PSI values.

GenePheno
Data frame containing sample metadata. Optional but highly recommended. This object should consist of at least 2 columns. Mandatory columns are sample.id and cell.type. sample.id indicates all unique sample IDs. cell.type indicates the group names of each sample. Additional columns, if present, may contain additional details of each sample such sequencing QC details etc..

GeneFeature
Data frame containing the gene metadata. Optional but highly recommended. Mandatory column is gene_id. Additional columns, if present, may contain details of each gene, e.g. gene name, gene type etc..

Exp
Data frame containing normalised and log-transformed gene expression values. Optional but highly recommended. The first column should be named gene_id and second column onwards should be sample names containing the gene expression values.

Details
This function creates an S3 object named Marvel for downstream analysis. It can take both splicing and gene expression data. Gene expression data is highly encouraged so that users can compare and contrast splicing and gene expression profiles using other functionalities by MARVEL.

Value
An object of class S3. Each slot in the object is named after the CreateMarvelObject arguments.

Author(s)
Sean Wen <sean.wenwx@gmail.com>

Examples
path_to_file <- system.file("extdata/Data", "SJ_phenoData.txt", package="MARVEL")
df.pheno <- read.table(path_to_file, sep="\t", header=TRUE, stringsAsFactors=FALSE, na.strings="NA")

# Read splice junction file
path_to_file <- system.file("extdata/Data", "SJ_phenoData.txt", package="MARVEL")
sj <- read.table(paste(path_to_file), sep="\t", header=TRUE, stringsAsFactors=FALSE, na.strings="NA")
# Read splicing event metadata file
df.feature.list <- list()

path_to_file <- system.file("extdata/Data", "SE_featureData.txt", package="MARVEL")
df.feature.list[[1]] <- read.table(paste(path_to_file), sep="\t", header=TRUE,
stringsAsFactors=FALSE, na.strings="NA")

names(df.feature.list) <- "SE"

# Create MARVEL object
marvel <- CreateMarvelObject(SplicePheno=df.pheno,
SpliceJunction=sj,
SpliceFeature=df.feature.list
)

class(marvel)

---

**IsoSwitch**  
*Detect Isoform Switch*

**Description**

IsoSwitch compare changes in PSI values with corresponding gene expression values to detect genes that have undergone isoform switching.

**Usage**

```r
IsoSwitch(MarvelObject, psi.de.sig, cell.types, method, method.adjust,
gene.de.sig)
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MarvelObject</td>
<td>S3 object generated from CompareValues function.</td>
</tr>
<tr>
<td>psi.de.sig</td>
<td>Numeric value. Adjusted p-value below which the splicing event is considered differentially spliced and included for isoform switching analysis.</td>
</tr>
<tr>
<td>cell.types</td>
<td>Character string. To indicate which 2 groups of cells that will be used for differential splicing analysis. Group names should match those in cell.type column of $SplicePheno slot.</td>
</tr>
<tr>
<td>method</td>
<td>Character string. Statistical test for differential gene expression analysis. Can take values &quot;ks&quot;, &quot;wilcox&quot; or &quot;t.test&quot;. Please refer to CompareValues function for more details.</td>
</tr>
<tr>
<td>method.adjust</td>
<td>Character strings. Adjust for multiple testing as per p.adjust function.</td>
</tr>
<tr>
<td>gene.de.sig</td>
<td>Numeric value. Adjusted p-value below which the gene is considered differentially expressed.</td>
</tr>
</tbody>
</table>
Details
This function compare changes in PSI values with corresponding gene expression values to detect genes that have undergone isoform switching. Isoform switch occurs when there is significant change in PSI values between 2 groups of cells in the absence of any significant change in gene expression values. This function also detect PSI values that change in the same or opposite direction with gene expression values.

Value
An object of class S3 containing all the original slots as inputted by the user in addition to three new slots named $DE$Cor*. $DE$Cor Original data frame generated from CompareValues function with an additional column to indicate if isoform switching has taken place. $DE$CorProp Tabulated proportion for each PSI-gene expression relationship. $DE$CorPlot Doughnut plot representing the values in $DE$CorProp.

Author(s)
Sean Wen <sean.wenwx@gmail.com>

Examples
```r
marvel <- readRDS(system.file("extdata/Data", "MarvelObject.rds", package="MARVEL"))
marvel <- IsoSwitch(MarvelObject=marvel, psi.de.sig=0.05, cell.types=c("iPSC", "Endoderm"), method="t.test", method.adjust="fdr", gene.de.sig=0.05)
marvel$DE$Cor[1:5, ]
marvel$DE$CorPlot
marvel$DE$CorProp
```

ModalityChange

Classify Modality Changes

Description
ModalityChange Classifies the type of modality change for each splicing event that has taken place between 2 groups of cells.

Usage
```r
ModalityChange(MarvelObject, psi.de.sig, cell.types, n.cells, sigma.sq, bimodal.adjust, seed, modality.column)
```
Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MarvelObject</td>
<td>S3 object generated from CompareValues function.</td>
</tr>
<tr>
<td>psi.de.sig</td>
<td>Numeric value. Adjusted p-value below which the splicing event is considered</td>
</tr>
<tr>
<td></td>
<td>differentially spliced and included for isoform switching analysis.</td>
</tr>
<tr>
<td>cell.types</td>
<td>Character string. To indicate which 2 groups of cells that will be used for</td>
</tr>
<tr>
<td></td>
<td>differential splicing analysis. Group names should match those in cell.type</td>
</tr>
<tr>
<td></td>
<td>column of $SplicePheno$ slot.</td>
</tr>
<tr>
<td>n.cells</td>
<td>Numeric value. The minimum no. of cells expressing the splicing event for</td>
</tr>
<tr>
<td></td>
<td>the event to be included for differential splicing analysis. Please refer to</td>
</tr>
<tr>
<td></td>
<td>AssignModality function help page for more details.</td>
</tr>
<tr>
<td>sigma.sq</td>
<td>Numeric value. The variance threshold below which the included/excluded</td>
</tr>
<tr>
<td></td>
<td>modality will be defined as primary sub-modality, and above which it will be</td>
</tr>
<tr>
<td></td>
<td>defined as dispersed sub-modality. Please refer to AssignModality function</td>
</tr>
<tr>
<td></td>
<td>help page for more details.</td>
</tr>
<tr>
<td>bimodal.adjust</td>
<td>Logical. When set to TRUE, MARVEL will identify false bimodal modalities and</td>
</tr>
<tr>
<td></td>
<td>reassign them as included/excluded modality. Please refer to AssignModality</td>
</tr>
<tr>
<td></td>
<td>function help page for more details.</td>
</tr>
<tr>
<td>seed</td>
<td>Numeric value. Ensure the fitdist function returns the same values for alpha</td>
</tr>
<tr>
<td></td>
<td>and beta paramters each time this function is executed using the same random</td>
</tr>
<tr>
<td></td>
<td>number generator. Please refer to AssignModality function help page for</td>
</tr>
<tr>
<td></td>
<td>more details.</td>
</tr>
<tr>
<td>modality.column</td>
<td>Character string. Can take the value &quot;modality&quot;, &quot;modality.var&quot; or</td>
</tr>
<tr>
<td></td>
<td>&quot;modality.bimodal.adj&quot;. Please refer to AssignModality function help page</td>
</tr>
<tr>
<td></td>
<td>for more details.</td>
</tr>
</tbody>
</table>

Details

This function classifies the type of modality change for each splicing event that has taken place between 2 groups of cells. Explicit: When modality changes between one of the five main modalities, e.g. included to multimodal. Implicit: When modality changes between primary and dispersed sub-modalities, e.g. included-primary to included-dispersed. Restricted: No modality change, e.g. included to included.

Value

An object of class S3 containing all the original slots as inputted by the user in addition to three new slots named $DE$Modality*. $DE$Modality Original data frame generated from CompareValues function with an additional columns to indicate the type of modality changes that have taken place between the 2 groups of cells. $DE$ModalityProp Tabulated proportion for each type of modality change. $DE$ModalityPlot Doughnut plot representing the values in $DE$ModalityProp.

Author(s)

Sean Wen <sean.wenwx@gmail.com>
Examples

```r
marvel <- readRDS(system.file("extdata/Data", "MarvelObject.rds", package="MARVEL"))

marvel <- ModalityChange(MarvelObject=marvel,
  psi.de.sig=0.05,
  cell.types=c("iPSC", "Endoderm"),
  n.cells=25,
  sigma.sq=0.001,
  bimodal.adjust=TRUE,
  seed=1,
  modality.column="modality.bimodal.adj"
)

marvel$DE$ModalityProp
marvel$DE$ModalityPlot
```

---

**PlotValues**

*Plot Percent Spliced-in (PSI) and Gene Expression Values*

**Description**

PlotValues plots percent spliced-in (PSI) and gene expression values across different groups of cells.

**Usage**

```r
PlotValues(MarvelObject, cell.types, feature, maintitle,
  xlabels.size = NULL, level, n.cells = NULL, sigma.sq = NULL,
  bimodal.adjust = NULL, seed = NULL, modality.column = NULL)
```

**Arguments**

- **MarvelObject**: S3 object generated from CreateMarvelObject or ComputePSI function.
- **cell.types**: Character string. To indicate which groups of cells that will be used for plotting. Group names should match those in cell.type column of $SplicePheno or GenePheno slot for splicing or gene expression data, respectively.
- **feature**: Character string. tran_id or gene_id for plotting. Should match tran_id or gene_id column of $ValidatedSpliceFeature or GeneFeature slot when level set to "splicing" or "gene", respectively.
- **maintitle**: Character string. Column to use as plot main title as per ValidatedSpliceFeature or GeneFeature when level set to "splicing" or "gene", respectively.
- **xlabels.size**: Numeric value. Size of x-axis labels as per ggplot2 function.
- **level**: Character string. Indicate "splicing" or "gene" for PSI or gene expression value plotting, respectively.
- **n.cells**: Numeric value. Only applicable when level set to "splicing". The minimum no. of cells expressing the splicing event to be included for analysis.
**PlotValues**

sigma.sq  Numeric value. Only applicable when level set to "splicing". The variance threshold below which the included/excluded modality will be defined as primary sub-modality, and above which it will be defined as dispersed sub-modality. Please refer to AssignModality function help page for more details.

bimodal.adjust Logical. Only applicable when level set to "splicing". When set to TRUE, MARVEL will identify false bimodal modalities and reassign them as included/excluded modality. Please refer to AssignModality function help page for more details.

seed Numeric value. Only applicable when level set to "splicing". Ensure the fitdist function returns the same values for alpha and beta parameters each time this function is executed using the same random number generator. Please refer to AssignModality function help page for more details.

modality.column Character string. Only applicable when level set to "splicing". Can take the value "modality", "modality.var" or "modality.bimodal.adj". Please refer to AssignModality function help page for more details.

Details

This function plots percent spliced-in (PSI) and gene expression values across different groups of cells. Violin plot is used for PSI values while boxplot is used for gene expression values.

Value

An object of class S3 containing all the original slots as inputted by the user in addition to one new slot named $adhocPlot$PSI or $adhocPlot$Gene when level set to "splicing" or "gene", respectively.

Author(s)

Sean Wen <sean.wenwx@gmail.com>

Examples

```r
marvel <- readRDS(system.file("extdata/Data", "MarvelObject.rds", package="MARVEL"))

feature <- marvel$SpliceFeature$SE$tran_id[1]

marvel <- PlotValues(MarvelObject=marvel,
  cell.types=c("iPSC", "Endoderm"),
  feature=feature,
  maintitle="gene_short_name",
  xlabels.size=12,
  level="splicing",
  n.cells=25,
  sigma.sq=0.001,
  bimodal.adjust=TRUE,
  seed=1,
  modality.column="modality.bimodal.adj"
  )
```
PlotValues.Exp

Plot Gene Expression Values

Description

PlotValues.Exp plots gene expression values across different groups of cells.

Usage

PlotValues.Exp(MarvelObject, cell.types, feature, maintitle = NULL)

Arguments

MarvelObject S3 object generated from CreateMarvelObject or ComputePSI function.

cell.types Character string. To indicate which groups of cells that will be used for plotting. Group names should match those in cell.type column of $GenePheno slot.


maintitle Character string. Column to use as plot main title as per GeneFeature.

Details

This function plots gene expression values across different groups of cells. Boxplot is used for gene expression values.

Value

An object of class S3 containing all the original slots as inputted by the user in addition to one new slot named $adhocPlot$Gene.

Author(s)

Sean Wen <sean.wenwx@gmail.com>

Examples

marvel <- readRDS(system.file("extdata/Data", "MarvelObject.rds", package="MARVEL"))

feature <- marvel$GeneFeature$gene_id[1]

marvel <- PlotValues.Exp(MarvelObject=marvel,
cell.types=c("iPSC", "Endoderm"),
feature=feature,
maintitle="gene_short_name"
)

marvel$adhocPlot$Gene
Description
PlotValues.PSI plots percent spliced-in (PSI) values across different groups of cells.

Usage
PlotValues.PSI(MarvelObject, cell.types, feature, maintitle, xlabels.size, n.cells, sigma.sq, bimodal.adjust, seed, modality.column)

Arguments
MarvelObject S3 object generated from CreateMarvelObject or ComputePSI function.
cell.types Character string. To indicate which groups of cells that will be used for plotting. Group names should match those in cell.type column of $SplicePheno slot.
feature Character string. tran_id for plotting. Should match tran_id column of $ValidatedSpliceFeature slot.
maintitle Character string. Column to use as plot main title as per $ValidatedSpliceFeature.
xlabels.size Numeric value. Size of x-axis labels as per ggplot2 function.
n.cells Numeric value. The minimum no. of cells expressing the splicing event to be included for analysis.
sigma.sq Numeric value. The variance threshold below which the included/excluded modality will be defined as primary sub-modality, and above which it will be defined as dispersed sub-modality. Please refer to AssignModality function help page for more details.
bimodal.adjust Logical. When set to TRUE, MARVEL will identify false bimodal modalities and reassign them as included/excluded modality. Please refer to AssignModality function help page for more details.
seed Numeric value. Ensure the fitdist function returns the same values for alpha and beta parameters each time this function is executed using the same random number generator. Please refer to AssignModality function help page for more details.
modality.column Character string. Can take the value "modality", "modality.var" or "modality.bimodal.adjust". Please refer to AssignModality function help page for more details.

Details
This function plots percent spliced-in (PSI) across different groups of cells. Violin plot is used for PSI values.
Value

An object of class S3 containing all the original slots as inputted by the user in addition to one new slot named $adhocPlot$PSI.

Author(s)

Sean Wen <sean.wenwx@gmail.com>

Examples

```r
marvel <- readRDS(system.file("extdata/Data", "MarvelObject.rds", package="MARVEL"))

feature <- marvel$SpliceFeature$SE$tran_id[1]

marvel <- PlotValues.PSI(MarvelObject=marvel, cell.types=c("iPSC", "Endoderm"), feature=feature, maintitle="gene_short_name", xlabels.size=12, n.cells=25, sigma.sq=0.001, bimodal.adjust=TRUE, seed=1, modality.column="modality.bimodal.adj")

marvel$adhocPlot$PSI
```

<table>
<thead>
<tr>
<th>PropModality</th>
<th>Modality Proportion</th>
</tr>
</thead>
</table>

Description

PropModality tabulates and plots the proportion of each modality.

Usage

```
PropModality(MarvelObject, modality.column, modality.type, event.type, across.event.type, prop.test = NULL, prop.adj = NULL)
```

Arguments

- **MarvelObject**: S3 object generated from AssignModality function.
- **modality.column**: Character string. Can take the value "modality", "modality.var" or "modality.bimodal.adj". Please refer to AssignModality function help page for more details.
PropModality.Bar

### modality.type
Character string. `basic` indicates that only the main modalities (included, excluded, bimodal, middle, multimodal) are analysed. Sub-modalities (primary and dispersed) will be merged. `extended` indicates that both main and sub-modalities are analysed. Sub-modalities will not be merged.

### event.type
Character string. To indicate which event type to analyse. Can take the value "SE", "MXE", "RI", "A5SS" or "A3SS". Specify "all" to include all event types.

### across.event.type
Logical. If set to TRUE, the proportion of modality will be compared across the specified event types

### prop.test
Character string. Only applicable when `across.event.type` set to TRUE. `chisq` Chi-squared test used to compare the proportion of modalities across the different event splicing type. `fisher` Fisher test used to compare the proportion of modalities across the different splicing event type.

### prop.adj
Character string. Only applicable when `across.event.type` set to TRUE. Adjust p-values generated from `prop.test` for multiple testing. Options available as per `p.adjust` function.

### Details
This function tabulates and plots the proportion of each modality for a specified splicing event type(s) or compares proportion of each modality across specified splicing event types.

### Value
An object of class S3 containing all the original slots as inputted by the user in addition to one new slot. If `across.event.type` set to FALSE, results returned to `$Modality$Prop$DoughnutChart` slot. If `across.event.type` set to TRUE, results returned to `$Modality$Prop$BarChart` slot.

### Author(s)
Sean Wen <sean.wenwx@gmail.com>

---

**PropModality.Bar**  
*Modality Proportion across Event Types.*

---

**Description**

`PropModality.Bar` compares proportion of each modality across specified splicing event types.

**Usage**

```r
PropModality.Bar(MarvelObject, modality.column, modality.type, event.type, prop.test, prop.adj)
```
Arguments

MarvelObject  S3 object generated from AssignModality function.
modality.column  Character string. Can take the value "modality", "modality.var" or "modality.bimodal.adj". Please refer to AssignModality function help page for more details.
modality.type  Character string. basic indicates that only the main modalities (included, excluded, bimodal, middle, multimodal) are analysed. Sub-modalities (primary and dispersed) will be merged. extended indicates that both main and sub-modalities are analysed. Sub-modalities will not be merged.
event.type  Character string. To indicate which event type to analyse. Can take the value "SE", "MXE", "RI", "A5SS" or "A3SS". Specify "all" to include all event types.
prop.test  Character string. Only applicable when across.event.type set to TRUE. chisq Chi-squared test used to compare the proportion of modalities across the different event splicing type. fisher Fisher test used to compare the proportion of modalities across the different splicing event type.
prop.adj  Character string. Only applicable when across.event.type set to TRUE. Adjust p-values generated from prop.test for multiple testing. Options available as per p.adjust function.

Details

This function compares proportion of each modality across specified splicing event types.

Value

An object of class S3 containing all the original slots as inputted by the user in addition to one new slot named $Modality$Prop$BarChart$ slot.

Author(s)

Sean Wen <sean.wenwx@gmail.com>

Examples

marvel <- readRDS(system.file("extdata/Data", "MarvelObject.rds", package="MARVEL"))

marvel <- PropModality.Bar(MarvelObject=marvel,
       modality.column="modality.bimodal.adj",
       modality.type="basic",
       event.type="all",
       prop.test="fisher",
       prop.adj="fdr"
       )

marvel$Modality$Prop$BarChart$Plot
**PropModality.Doughnut**

*Modality Proportion for a Given Event Type(s).*

**Description**

PropModality.Doughnut tabulates and plots the proportion of each modality for a specified splicing event type(s).

**Usage**

```r
PropModality.Doughnut(MarvelObject, modality.column, modality.type, event.type)
```

**Arguments**

- **MarvelObject**  
  S3 object generated from AssignModality function.

- **modality.column**  
  Character string. Can take the value "modality", "modality.var" or "modality.bimodal.adj". Please refer to AssignModality function help page for more details.

- **modality.type**  
  Character string. basic indicates that only the main modalities (included, excluded, bimodal, middle, multimodal) are analysed. Sub-modalities (primary and dispersed) will be merged. extended indicates that both main and sub-modalities are analysed. Sub-modalities will not be merged.

- **event.type**  
  Character string. To indicate which event type to analyse. Can take the value "SE", "MXE", "RI", "A5SS" or "A3SS". Specify "all" to include all event types.

**Details**

This function tabulates and plots the proportion of each modality for a specified splicing event type(s).

**Value**

An object of class S3 containing all the original slots as inputted by the user in addition to one new slot name `$Modality$Prop$DoughnutChart`.

**Author(s)**

Sean Wen <sean.wenwx@gmail.com>

**Examples**

```r
marvel <- readRDS(system.file("extdata/Data", "MarvelObject.rds", package="MARVEL"))

marvel <- PropModality.Doughnut(MarvelObject=marvel, 
  modality.column="modality.bimodal.adj", 
  modality.type="basic", 
  event.type="all")
```
RunPCA

Principle Component Analysis

Description

RunPCA performs principle component analysis on splicing and gene expression data.

Usage

RunPCA(MarvelObject, cell.types, n.cells, features, point.size, level, 
  event.type = NULL, seed = NULL)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MarvelObject</td>
<td>S3 object generated from CreateMarvelObject or ComputePSI function.</td>
</tr>
<tr>
<td>cell.types</td>
<td>Character string. To indicate which groups of cells that will be used for analysis. Group names should match those in cell.type column of $SplicePheno or GenePheno slot for splicing or gene expression data, respectively.</td>
</tr>
<tr>
<td>n.cells</td>
<td>Numeric value. The minimum no. of cells expressing the splicing event or gene for the event or gene, respectively, to be included for analysis.</td>
</tr>
<tr>
<td>features</td>
<td>Character string. Vector of tran_ids or gene_ids for analysis. Should match tran_id or gene_id column of $ValidatedSpliceFeature or GeneFeature slot when level set to &quot;splicing&quot; or &quot;gene&quot;, respectively.</td>
</tr>
<tr>
<td>point.size</td>
<td>Numeric value. Size of data points on reduced dimension space.</td>
</tr>
<tr>
<td>level</td>
<td>Character string. Indicate &quot;splicing&quot; or &quot;gene&quot; for splicing or gene expression analysis, respectively</td>
</tr>
<tr>
<td>event.type</td>
<td>Character string. Only applicable when level set to &quot;splicing&quot;. Indicate which splicing event type to include for analysis. Can take value &quot;SE&quot;, &quot;MXE&quot;, &quot;RI&quot;, &quot;A5SS&quot;, or &quot;A3SS&quot; which represents skipped-exon (SE), mutually-exclusive exons (MXE), retained-intron (RI), alternative 5’ splice site (A5SS), and alternative 3’ splice site (A3SS), respectively.</td>
</tr>
<tr>
<td>seed</td>
<td>Numeric value. Only applicable when level set to &quot;splicing&quot;. Ensures imputed values for NA PSIs are reproducible.</td>
</tr>
</tbody>
</table>

Details

This function performs principle component analysis on splicing and gene expression data and visualise cells on a reducted dimension space, i.e. 2D scatterplot.
RunPCA.Exp

Value
An object of class S3 containing all the original slots as inputted by the user in addition to one new slot named MarvelObject$PCA$PSI or MarvelObject$PCA$Gene when level set to "splicing" or "gene", respectively. Contains 2D scatterplot in MarvelObject$PCA$PSI$Plot or MarvelObject$PCA$Gene$Plot and the corresponding x- and y-coordinates for each sample in MarvelObject$PCA$PSI$Results or MarvelObject$PCA$Gene$Results, respectively.

Author(s)
Sean Wen <sean.wenwx@gmail.com>

Examples
```
marvel <- readRDS(system.file("extdata/Data", "MarvelObject.rds", package="MARVEL"))

features <- do.call(rbind.data.frame, marvel$SpliceFeatureValidated)
features <- features$tran_id

marvel <- RunPCA(MarvelObject=marvel,
cell.types="all",
n.cells=3,
features=features,
point.size=2.5,
level="splicing",
event.type="all",
seed=1)

marvel$PCA$PSI$Results
marvel$PCA$PSI$Plot
```

---

RunPCA.Exp  Principle Component Analysis on Gene Expression Values

Description
RunPCA.Exp performs principle component analysis on gene expression values.

Usage
```
RunPCA.Exp(MarvelObject, cell.types, n.cells, features, point.size)
```

Arguments
- **MarvelObject**: S3 object generated from CreateMarvelObject or ComputePSI function.
- **cell.types**: Character string. To indicate which groups of cells that will be used for analysis. Group names should match those in cell.type column of $GenePheno slot.
RunPCA.PSI

```
n.cells    Numeric value. The minimum no. of cells expressing the splicing event or gene for the event or gene, respectively, to be included for analysis.
point.size Numeric value. Size of data points on reduced dimension space.
```

Details

This function performs principle component analysis on gene expression values and visualise cells on a reduced dimension space, i.e. 2D scatterplot.

Value

An object of class S3 containing all the original slots as inputted by the user in addition to one new slot named MarvelObject$PCA$Gene. Contains both 2D scatterplot in MarvelObject$PCA$Gene$Plot and the corresponding x- and y-coordinates for each sample in MarvelObject$PCA$Gene$Results.

Author(s)

Sean Wen <sean.wenwx@gmail.com>

Examples

```r
marvel <- readRDS(system.file("extdata/Data", "MarvelObject.rds", package="MARVEL"))
features <- marvel$GeneFeature$gene_id
marvel <- RunPCA.Exp(MarvelObject=marvel, cell.types="all", n.cells=3, features=features, point.size=2.5)
marvel$PCA$Gene$Results
marvel$PCA$Gene$Plot
```

---

RunPCA.PSI  

**Principle Component Analysis on Percent Spliced-in (PSI) Values**

Description

RunPCA.PSI performs principle component analysis on percent spliced-in (PSI) values.

Usage

```
RunPCA.PSI(MarvelObject, cell.types, n.cells, features, point.size, event.type, seed)
```
RunPCA.PSI

Arguments

MarvelObject  
S3 object generated from CreateMarvelObject or ComputePSI function.

cell.types  
Character string. To indicate which groups of cells that will be used for analysis. Group names should match those in cell.type column of $SplicePheno slot.

n.cells  
Numeric value. The minimum no. of cells expressing the splicing event or gene for the event or gene, respectively, to be included for analysis.

features  
Character string. Vector of tran_ids for analysis. Should match tran_id column of $ValidatedSpliceFeature slot.

point.size  
Numeric value. Size of data points on reduced dimension space.

event.type  
Character string. Indicate which splicing event type to include for analysis. Can take value "SE", "MXE", "RI", "A5SS", or "A3SS" which represents skipped-exon (SE), mutually-exclusive exons (MXE), retained-intron (RI), alternative 5' splice site (A5SS), and alternative 3' splice site (A3SS), respectively.

seed  
Numeric value. Ensures imputed values for NA PSIs are reproducible.

Details

This function performs principle component analysis on percent spliced-in (PSI) values and visualise cells on a reducted dimension space, i.e. 2D scatterplot.

Value

An object of class S3 containing all the original slots as inputted by the user in addition to one new slot named MarvelObject$PCA$PSI. Contains both 2D scatterplot in MarvelObject$PCA$PSI$Plot and the corresponding x- and y-coordinates for each sample in MarvelObject$PCA$PSI$Results.

Author(s)

Sean Wen <sean.wenwx@gmail.com>

Examples

marvel <- readRDS(system.file("extdata/Data", "MarvelObject.rds", package="MARVEL"))

features <- do.call(rbind.data.frame, marvel$SpliceFeatureValidated)
features <- features$tran_id

marvel <- RunPCA.PSI(MarvelObject=marvel,
  cell.types="all",
  n.cells=3,
  features=features,
  point.size=2.5,
  event.type="all",
  seed=1)

marvel$PCA$PSI$Results
marvel$PCA$PSI$Plot
Index

AssignModality, 2
BioPathways, 4
CompareValues, 5
CompareValues.Exp, 6
CompareValues.PSI, 7
ComputePSI, 8
ComputePSI.A3SS, 10
ComputePSI.A5SS, 11
ComputePSI.MXE, 12
ComputePSI.RI, 13
ComputePSI.SE, 14
CreateMarvelObject, 15

IsoSwitch, 17

ModalityChange, 18

PlotValues, 20
PlotValues.Exp, 22
PlotValues.PSI, 23
PropModality, 24
PropModality.Bar, 25
PropModality.Doughnut, 27

RunPCA, 28
RunPCA.Exp, 29
RunPCA.PSI, 30