Package 'MARVEL'

October 31, 2022

Title Revealing Splicing Dynamics at Single-Cell Resolution

Version 1.4.0

Description
Alternative splicing represents an additional and underappreciated layer of complexity underlying gene expression profiles. Nevertheless, there remains hitherto a paucity of software to investigate splicing dynamics at single-cell resolution. 'MARVEL' enables splicing analysis of single-cell RNA-sequencing data generated from plate- and droplet-based library preparation methods.

Imports ggplot2 (>= 3.3.2), Matrix (>= 1.3-3), methods, plyr (>= 1.8.4), scales (>= 1.1.1)

Suggests AnnotationDbi (>= 1.48.0), Biostrings (>= 2.56.0), BSgenome (>= 1.56.0), BSgenome.Hsapiens.NCBI.GRCh38 (>= 1.3.1000), clusterProfiler (>= 3.16.0), factoextra (>= 1.0.7), FactoMineR (>= 2.3), fitdistrplus (>= 1.1-1), GenomicRanges (>= 1.42.0), ggnewscale (>= 0.4.5), ggrepel (>= 0.9.1), gridExtra (>= 2.3), gtools (>= 3.9.2), IRanges (>= 2.24.1), kableExtra (>= 1.3.1), knitr (>= 1.29), kSamples (>= 1.2-9), markdown (>= 1.1), MAST (>= 1.16.0), org.Hs.eg.db (>= 3.10.0), org.Mm.eg.db (>= 3.11.4), parallel, pheatmap (>= 1.0.12), reshape2 (>= 1.4.4), rmarkdown (>= 2.3), S4Vectors (>= 0.26.1), stringr (>= 1.4.0), textclean (>= 0.9.3), twosamples (>= 1.1.1), wiggleplotr (>= 1.12.1)

Encoding UTF-8

License GPL-3

RoxygenNote 7.2.0

VignetteBuilder knitr

ByteCompile true

NeedsCompilation no

Author Sean Wen [aut, cre]

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

Depends R (>= 3.5.0)

Repository CRAN

Date/Publication 2022-10-31 10:22:50 UTC

R topics documented:

adhocGene.DE.Gene.10x *Differential gene expression analysis of specified gene*

Description

Performs differential gene expression analysis specified gene across for all possible pairs of cell groups. The gene and cell groups were defined earlier in adhocGene.TabulateExpression.Gene.10x function.

Usage

```
adhocGene.DE.Gene.10x(MarvelObject)
```

Arguments

MarvelObject	Marvel object. S3 object generated from adhocGene.TabulateExpression.Gene.10x function.

Value

An object of class S3 with new slots MarvelObject$adhocGene$DE$Gene$Data.

Examples

```
marvel.demo.10x <- readRDS(system.file("extdata/data",
                                "marvel.demo.10x.rds",
                                package="MARVEL")
                                )

marvel.demo.10x <- adhocGene.DE.Gene.10x(MarvelObject=marvel.demo.10x)

# Check output
marvel.demo.10x$adhocGene$DE$Gene$Data
```

adhocGene.DE.PSI.10x *Differential splice junction analysis of specified gene*

Description

Performs differential splice junction analysis specified gene across for all possible pairs of cell groups. The gene and cell groups were defined earlier in adhocGene.TabulateExpression.Gene.10x function.

Usage

```
adhocGene.DE.PSI.10x(MarvelObject)
```

Arguments

MarvelObject	Marvel object. S3 object generated from adhocGene.TabulateExpression.PSI.10x function.

Value

An object of class S3 with new slots MarvelObject$adhocGene$DEPSIData.

Examples

```
marvel.demo.10x <- readRDS(system.file("extdata/data",
                                "marvel.demo.10x.rds",
                                package="MARVEL")
                                )

marvel.demo.10x <- adhocGene.DE.PSI.10x(MarvelObject=marvel.demo.10x)

# Check output
marvel.demo.10x$adhocGene$DE$PSI$Data
```

---

adhocGene.PlotDEValues.10x

Plot differential splice junction analysis results for a specified gene

---

Description

Scatterplot of results from differential gene and splice junction analysis. x-axis represents the gene expression log2 fold change between the different pairs of cell groups. y-axis represents the PSI differences or log2 fold change between the different pairs of cell groups.

Usage

```
adhocGene.PlotDEValues.10x(
  MarvelObject,
  coord.intron,
  log2fc.gene = 0.5,
  delta.sj = 5,
  label.size = 2,
  point.size = 2,
  xmin = NULL,
  xmax = NULL,
  ymin = NULL,
  ymax = NULL
)
```

Arguments

MarvelObject	Marvel object. S3 object generated from adhocGene.DE.Gene.10x and adhocGene.DE.PSI.10x functions.
coord.intron	Character string. Coordinates of splice junction whose differential splice junction results will be plotted.
log2fc.gene	Numeric value. Absolute log2 fold change, above which, the gene is considered differentially expressed.
delta.sj	Numeric value. Absolute differences in average PSI values between the two cell groups, above which, the splice junction is considered differentially spliced.
label.size	Numeric value. The font size of the group comparison labels on the plot will be adjusted to the size specified here. Default is 2.
point.size	Numeric value. Size of data points. Default is 2.
xmin	Numeric value. Minimum x-axis value.
xmax	Numeric value. Maximum x-axis value.
ymin	Numeric value. Minimum y-axis value.
ymax	Numeric value. Maximum y-axis value.

Value

An object of class S3 with a new slots MarvelObject$adhocGene$DE$VolcanoPlot$Plot and MarvelObject$adhocGene$DE$VolcanoPlot$Table.

Examples

```
marvel.demo.10x <- readRDS(system.file("extdata/data",
                                "marvel.demo.10x.rds",
                                package="MARVEL")
                                )

# Define SJ to plot
coord.intron <- marvel.demo.10x$adhocGene$DE$PSI$Data$coord.intron[1]

# Plot SJ vs gene
marvel.demo.10x <- adhocGene.PlotDEValues.10x(
                        MarvelObject=marvel.demo.10x,
                        coord.intron=coord.intron,
                        log2fc.gene=0.5,
                        delta.sj=5,
                        label.size=2,
                        point.size=2,
                        xmin=-2.0,
                        xmax=2.0,
                        ymin=-25,
                        ymax=25
                        )

# Check output
marvel.demo.10x$adhocGene$DE$VolcanoPlot$Plot
```

adhocGene.PlotSJPosition.10x *Plots the locations of specified splice junction relative to isoforms*

Description

Plots the locations of specified splice junction relative to isoforms. List of isoforms are retrieved from GTF.

Usage

```
adhocGene.PlotSJPosition.10x(
  MarvelObject,
  coord.intron,
  coord.intron.ext = 50,
  rescale_introns = FALSE,
  show.protein.coding.only = TRUE,
  anno.label.size = 3,
  anno.colors = c("black", "gray", "red")
)
```

Arguments

`MarvelObject`	Marvel object. S3 object generated from `CheckAlignment.10x` function.
`coord.intron`	Character string. Coordinates of splice junction whose splice junction will be plotted.
`coord.intron.ext`	Numeric value. Number of bases to extend the splice junction start and end coordinates into the exons. Helpful to enhance splice junction locations on the plot. Default is `50`.
`rescale_introns`	Logical value. If set to `TRUE`, the intron length will be shorten. Helpful when introns are very long and focus visualisation of exons and splice junctions. Default is `FALSE`.
`show.protein.coding.only`	Logical value. If set to `TRUE` (default), only protein-coding isoforms will be displayed.
`anno.label.size`	Numeric value. Font size of isoform ID labels. Default is `3`.
`anno.colors`	Vector of character strings. Colors for non-coding UTRs, coding exons, and splice junctions, respectively. Default is `c("black", "gray", "red")`.

Value

An object of class S3 with new slots `MarvelObject$adhocGene$SJPosition$Plot`, `MarvelObject$adhocGene$SJPositic`
`MarvelObject$adhocGene$SJPosition$exonfile`, and `MarvelObject$adhocGene$SJPosition$cdsfile`.

Examples

```
marvel.demo.10x <- readRDS(system.file("extdata/data",
                                "marvel.demo.10x.rds",
                                package="MARVEL")
                                )

marvel.demo.10x <- adhocGene.PlotSJPosition.10x(
                         MarvelObject=marvel.demo.10x,
                         coord.intron="chr1:100:1001",
                         rescale_introns=FALSE,
                         show.protein.coding.only=TRUE,
                         anno.label.size=1.5
                         )
```

---

adhocGene.TabulateExpression.Gene.10x

Dotplot of gene expression values for a specified gene

Description

Creates a dotplot of average expression value of a specified gene across different cell groups.

Usage

```
adhocGene.TabulateExpression.Gene.10x(
  MarvelObject,
  cell.group.list,
  gene_short_name,
  log2.transform = TRUE,
  min.pct.cells = 10,
  downsample = FALSE,
  seed = 1
)
```

Arguments

`MarvelObject`	Marvel object. S3 object generated from `CheckAlignment.10x` function.
`cell.group.list`	List of character strings. Each element of the list is a vector of cell IDs corresponding to a cell group.
`gene_short_name`	Character string. Gene names whose expression will be plotted.
`log2.transform`	Logical value. If set to `TRUE` (default), normalised gene expression values will be off-set by 1 and then log2-transformed prior to plotting.

`min.pct.cells` Numeric value. Percentage of cell expressing the gene in a cell group, below which, the value be re-coded as missing and appear will be omitted from the plot. A gene is considered to be expressed in a given cell if it has non-zero normalised count.

`downsample` Logical value. If set to `TRUE`, the number of cells in each cell group will be down-sampled so that all cell groups will have the same number of cells. The number of cells to down-sample will be based on the smallest cell group. Default is `FALSE`.

`seed` Numeric value. Random number generator to be fixed for down-sampling.

Value

An object of class S3 with new slots `MarvelObject$adhocGene$Expression$Gene$Table`, `MarvelObject$adhocGene$Exp`
`MarvelObject$adhocGene$cell.group.list`, and `MarvelObject$adhocGene$gene_short_name`.

Examples

```
marvel.demo.10x <- readRDS(system.file("extdata/data",
                                "marvel.demo.10x.rds",
                                package="MARVEL")
                                )

# Define cell groups
    # Retrieve sample metadata
    sample.metadata <- marvel.demo.10x$sample.metadata

    # iPSC
    index <- which(sample.metadata$cell.type=="iPSC")
    cell.ids.1 <- sample.metadata[index, "cell.id"]
    length(cell.ids.1)

    # Cardio day 10
    index <- which(sample.metadata$cell.type=="Cardio day 10")
    cell.ids.2 <- sample.metadata[index, "cell.id"]
    length(cell.ids.2)

    # Save into list
    cell.group.list <- list("iPSC"=cell.ids.1,
                            "Cardio d10"=cell.ids.2
                            )

# Gene expression profiling
marvel.demo.10x <- adhocGene.TabulateExpression.Gene.10x(
                        MarvelObject=marvel.demo.10x,
                        cell.group.list=cell.group.list,
                        gene_short_name="TPM2",
                        min.pct.cells=10,
                        downsample=TRUE
                        )
```

```
# Check output
marvel.demo.10x$adhocGene$Expression$Gene$Plot
marvel.demo.10x$adhocGene$Expression$Gene$Table
```

---

adhocGene.TabulateExpression.PSI.10x

Dotplot of splice junction expression values for a specified gene

---

Description

Creates a dotplot of splice junction expression value of a specified gene across different cell groups. The gene and cell groups were defined earlier in adhocGene.TabulateExpression.Gene.10x function.

Usage

```
adhocGene.TabulateExpression.PSI.10x(MarvelObject, min.pct.cells = 10)
```

Arguments

MarvelObject	Marvel object. S3 object generated from adhocGene.TabulateExpression.Gene.10x function.
min.pct.cells	Numeric value. Percentage of cell expressing the splice junction in a cell group, below which, the value be re-coded as missing and appear will be omitted from the plot. A splice junction is considered to be expressed in a given cell if it has count >=1.

Value

An object of class S3 with new slots MarvelObject$adhocGene$ExpressionPSITable and MarvelObject$adhocGene$ExpressionPSIPlot.

Examples

```
marvel.demo.10x <- readRDS(system.file("extdata/data",
                                "marvel.demo.10x.rds",
                                package="MARVEL")
                                )

# SJ usage profiling
marvel.demo.10x <- adhocGene.TabulateExpression.PSI.10x(
                        MarvelObject=marvel.demo.10x,
                        min.pct.cells=10
                        )

# Check output
marvel.demo.10x$adhocGene$Expression$PSI$Plot
```

AnnotateGenes.10x *Annotate splice junctions*

Description

Annotates the each gene in the gene metadata with the gene type, e.g. protein-coding, antisense etc.. Annotations are retrieved from GTF. Only genes found in gene metadata and GTF will be retained.

Usage

```
AnnotateGenes.10x(MarvelObject)
```

Arguments

MarvelObject	Marvel object. S3 object generated from CreateMarvelObject.10x function.

Value

An object of class S3 containing the updated slots MarvelObject$gene.metadata and gene.norm.matrix.

Examples

```
# Load un-processed MARVEL object
marvel.demo.10x.raw <- readRDS(system.file("extdata/data",
                               "marvel.demo.10x.raw.rds",
                               package="MARVEL")
                               )

# Annotate gene metadata
marvel.demo.10x <- AnnotateGenes.10x(MarvelObject=marvel.demo.10x.raw)
```

AnnotateSJ.10x *Annotate splice junctions*

Description

Annotates the splice junctions by assigning the gene name to the start and end of the splice junction. Annotations are retrieved from GTF.

Usage

```
AnnotateSJ.10x(MarvelObject)
```

Arguments

MarvelObject	Marvel object. S3 object generated from AnnotateGenes.10x function.

Value

An object of class S3 containing the updated slot `MarvelObject$sj.metadata`.

Examples

```
# Load un-processed MARVEL object
marvel.demo.10x.raw <- readRDS(system.file("extdata/data",
                               "marvel.demo.10x.raw.rds",
                               package="MARVEL")
                               )

# Annotate gene metadata
marvel.demo.10x <- AnnotateGenes.10x(MarvelObject=marvel.demo.10x.raw)

# Annotate junction metadata
marvel.demo.10x <- AnnotateSJ.10x(MarvelObject=marvel.demo.10x)
```

`AnnoVolcanoPlot` *Annotate volcano plot with nonsense-mediated decay (NMD) genes*

Description

Annotate volcano plot generated from differential gene expression analysis with genes predicted to undergo splicing-induced NMD.

Usage

```
AnnoVolcanoPlot(
  MarvelObject,
  anno = FALSE,
  anno.gene_short_name = NULL,
  label.size = NULL,
  point.size = 1,
  xlabel.size = 8
)
```

Arguments

`MarvelObject`	Marvel object. S3 object generated from `CompareExpr` function.
`anno`	Logical value. If set to `TRUE`, selected gene names will be annotated on the plot as defined in `gene.label.x.below` and `gene.label.y.above`.
`anno.gene_short_name`	Vector of character strings. When `anno` set to `TRUE`, the gene names to annotate on the plot.
`label.size`	Numeric value. When `anno` set to `TRUE`, the size of gene labels.
`point.size`	Numeric value. Size of data points. Default value is `1`.
`xlabel.size`	Numeric value. Font size of the xtick labels. Default is `8`.

Value

An object of class S3 with new slots `MarvelObject$NMD$AnnoVolcanoPlot$Table` and `MarvelObject$NMD$AnnoVolcanoP`

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

marvel.demo <- AnnoVolcanoPlot(MarvelObject=marvel.demo)

# Check outputs
head(marvel.demo$NMD$AnnoVolcanoPlot$Table)
marvel.demo$NMD$AnnoVolcanoPlot$Plot
```

`AssignModality` *Assign modalities*

Description

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

Usage

```
AssignModality(
  MarvelObject,
  sample.ids,
  min.cells = 25,
  sigma.sq = 0.001,
  bimodal.adjust = TRUE,
  bimodal.adjust.fc = 3,
  bimodal.adjust.diff = 50,
  seed = 1,
  tran_ids = NULL
)
```

Arguments

`MarvelObject`	Marvel object. S3 object generated from `TransformExpValues` function.
`sample.ids`	Vector of character strings. Sample IDs that constitute the cell group.
`min.cells`	Numeric value. The minimum no. of cells expressing the splicing event for the event to be included for modality assignment.
`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.

`bimodal.adjust.fc`
: Numeric value. The ratio between the proportion of cells with >0.75 PSI vs <0.25 PSI (and vice versa) below which the splicing event will be classified as bimodal. Only applicable when `bimodal.adjust` set to `TRUE`. To be used in conjunction with `bimodal.adjust.diff`.

`bimodal.adjust.diff`
: Numeric value. The difference between the percentage of cells with >0.75 PSI vs <0.25 PSI (and vice versa) below which the splicing event will be classified as bimodal. Only applicable when `bimodal.adjust` set to `TRUE`. To be used in conjunction with `bimodal.adjust.fc`.

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

`tran_ids`
: Character strings. Specific vector of transcript IDs for modality assignment. This will be a subset of all transcripts expressed in sufficient number of cells as defined in `min.cells` option.

Value

An object of class S3 containing with new slot `MarvelObject$Modality$Results`.

Author(s)

Sean Wen <sean.wenwx@gmail.com>

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

df.pheno <- marvel.demo$SplicePheno
sample.ids <- df.pheno[which(df.pheno$cell.type=="iPSC"), "sample.id"]

# Assign modality
marvel.demo <- AssignModality(MarvelObject=marvel.demo,
                              sample.ids=sample.ids,
                              min.cells=5
                              )

# Check output
head(marvel.demo$Modality$Results)
```

`BioPathways` *Pathway enrichment analysis*

Description

Performs pathway enrichment analysis on differentially spliced genes or user-specified custom set of genes.

Usage

```
BioPathways(
  MarvelObject,
  method = NULL,
  pval = NULL,
  delta = 0,
  n.top = NULL,
  method.adjust = "fdr",
  custom.genes = NULL,
  species = "human"
)
```

Arguments

MarvelObject	Marvel object. S3 object generated from CompareValues function.
method	Character string. The statistical method used for differential splicing analysis.
pval	Numeric value. Alternative to n.top and custom.genes, i.e. choose one of these three options. Adjusted p-value below which the splicing events are considered differentially spliced and their corresponding genes are included for gene ontology analysis. If this argument is specified, then n.top must not be specified.
delta	Numeric value. The absolute difference between the means PSI values of cell group 1 and 2, above which, the splicing event is considered differentially spliced and their corresponding genes are included for gene ontology analysis.
n.top	Numeric value. Alternative to pval to custom.genes, i.e. choose one of these three options.. Indicate the top n splicing events with the smallest adjusted p-values are differentially spliced and their corresponding genes are included for gene ontology analysis. If this argument is specified, then pval must not be specified.
method.adjust	Character string. Adjust p-values for multiple testing. Options available as per p.adjust function.
custom.genes	Character strings. Alternative to pval and n.top, i.e. choose one of these three options.. Vector of gene names to be assessed for enrichment of biological pathways.
species	Character strings. Takes the value "human" or "mouse", which corresponds to human and mouse genes, respectively. Default value is "human". This will enable MARVEL to retrieve the relevant database for GO analysis.

Value

An object of class S3 with new slot MarvelObjectDEBioPathways$Table.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

marvel.demo <- BioPathways(MarvelObject=marvel.demo,
```

```
method="ad",
custom.genes=c("RPL26", "SNRPN")
)
```

`BioPathways.10x` *Pathway enrichment analysis*

Description

Performs pathway enrichment analysis on differentially spliced genes or user-specified custom set of genes.

Usage

```
BioPathways.10x(
  MarvelObject,
  pval = 0.05,
  log2fc = NULL,
  delta = 5,
  min.gene.norm = 0,
  method.adjust = "fdr",
  custom.genes = NULL,
  species = "human",
  remove.ribo = FALSE
)
```

Arguments

`MarvelObject`	Marvel object. S3 object generated from `CompareValues.Genes.10x` function.
`pval`	Numeric value. p-value, above which, the splice junction is considered differentially spliced. Default is `0.05`.
`log2fc`	Numeric value. Absolute log2 fold change from differential splicing analysis, above which, the splice junction is considered differentially spliced. This option should be `NULL` if `delta` has been specified.
`delta`	Numeric value. Absolute difference in average PSI values between the two cell groups, above which, the splice junction is considered differentially spliced. This option should be `NULL` if `log2fc` has been specified.
`min.gene.norm`	Numeric value. The average normalised gene expression across the two cell groups above which the splice junction is considered differentially spliced. Default is `0`.
`method.adjust`	Character string. Adjust p-values for multiple testing. Options available as per `p.adjust` function.
`custom.genes`	Character strings. Alternative to `pval` and `delta`. Vector of gene names to be assessed for enrichment of biological pathways.
`species`	Character strings. Takes the value `"human"` or `"mouse"`, which corresponds to human and mouse genes, respectively. Default value is `"human"`.

remove.ribo	Logical value. If set to TRUE, ribosomal genes will be removed prior to GO analysis. This may prevent high-expressing ribosomal genes from overshadowing more biological relevant genes for GO analysis. Default value is FALSE.
method	Character string. The statistical method used for differential splicing analysis.

Value

An object of class S3 containing new slot MarvelObjectDEBioPathways$Table.

Examples

```
marvel.demo.10x <- readRDS(system.file("extdata/data",
                                "marvel.demo.10x.rds",
                                package="MARVEL")
                                )

marvel.demo.10x <- BioPathways.10x(
                        MarvelObject=marvel.demo.10x,
                        custom.genes=c("TPM2", "GNAS"),
                        species="human"
                        )
```

BioPathways.Plot *Plot pathway enrichment analysis results*

Description

Plots user-specified enriched pathways.

Usage

```
BioPathways.Plot(
  MarvelObject,
  go.terms,
  y.label.size = 10,
  offset = 0.5,
  x.axis = "enrichment"
)
```

Arguments

MarvelObject	Marvel object. S3 object generated from BioPathways function.
go.terms	Vector of character strings. Names of pathways to plot. Should match pathway names in column Description of MarvelObjectDEBioPathways$Table.
y.label.size	Numeric value. Size of y-axis tick labels, i.e. gene set names.
offset	Numeric value. The -log10(p-value) on the x-axis to substract or add to increase the plot margins.

x.axis	Character string. If set to "enrichment" (default) the pathway enrichment will be displayed on the x-axis while the color intensity of the data points will reflect the -log10(adjusted p-value). If set to "pval" the -log10(adjusted p-value) will be displayed on the x-axis while the color intensity of the data points will reflect the pathway enrichment.

Details

This function plots selected gene sets returned from gene ontology analysis performed previously using BioPathways

Value

An object of class S3 with new slot MarvelObjectDEBioPathways$Plot.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

# Define go terms to plot
df <- marvel.demo$DE$BioPathways$Table
go.terms <- df$Description[c(1:10)]

# Plot
marvel.demo <- BioPathways.Plot(MarvelObject=marvel.demo,
                                go.terms=go.terms,
                                offset=10
                                )

# Check output
marvel.demo$DE$BioPathways$Plot
```

BioPathways.Plot.10x *Plot pathway enrichment analysis results*

Description

Plots user-specified enriched pathways.

Usage

```
BioPathways.Plot.10x(MarvelObject, go.terms, y.label.size = 10, offset = 0.5)
```

Arguments

MarvelObject	Marvel object. S3 object generated from BioPathways.10x function.
go.terms	Vector of character strings. Names of pathways to plot. Should match pathway names in column Description of MarvelObjectDEBioPathways$Table.

y.label.size	Numeric value. Size of y-axis tick labels, i.e. pathway names.
offset	Numeric value. The value on the x-axis to substract or add to increase the plot margins.

Value

An object of class S3 containing with new slot MarvelObjectDEBioPathways$Plot.

Examples

```
marvel.demo.10x <- readRDS(system.file("extdata/data",
                                "marvel.demo.10x.rds",
                                package="MARVEL")
                                )

# Define top pathways to plot
go.terms <- marvel.demo.10x$DE$BioPathways$Table$Description
go.terms <- go.terms[c(1:10)]

# Plot
marvel.demo.10x <- BioPathways.Plot.10x(
                            MarvelObject=marvel.demo.10x,
                            go.terms=go.terms
                            )
# Check outpout
marvel.demo.10x$DE$BioPathways$Plot
```

CheckAlignment *Pre-flight check*

Description

Checks if the metadata aligns with the columns and rows of the matrix for splicing or gene data. This is a wrapper function for CheckAlignment.PSI, CheckAlignment.Exp, CheckAlignment.PSI.Exp, and CheckAlignment.SJ.

Usage

```
CheckAlignment(MarvelObject, level)
```

Arguments

MarvelObject	Marvel object. S3 object generated from CreateMarvelObject function.
level	Character string. Indicate "SJ", "splicing" or "gene" for splice junction, splicing or gene data, respectively. "SJ" typically specified before computing PSI values. "splicing" or "gene" typically specified after computing PSI values.

Value

An object of class S3 with updated slots `MarvelObject$SpliceJunction`, `MarvelObject$IntronCoverage`, `MarvelObject$SplicePheno`, `MarvelObject$SpliceFeatureValidated`, and `MarvelObject$PSI` or `MarvelObject$GenePheno`, `MarvelObject$GeneFeature`, and `MarvelObject$Gene` are updated for splicing or gene data, respectively.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

marvel.demo <- CheckAlignment(MarvelObject=marvel.demo,
                              level="SJ"
                              )
```

CheckAlignment.10x *Pre-flight check*

Description

Ensures only overlapping cells found in both gene and splice junction data are retained. Also ensures matrix columns matches cell IDs in sample metadata and matrix rows matches gene name or splice junction coordinates in feature metadata.

Usage

```
CheckAlignment.10x(MarvelObject)
```

Arguments

`MarvelObject`	Marvel object. S3 object generated from `FilterGenes.10x` function.

Value

An object of class S3 containing updated slots `MarvelObject$gene.norm.matrix`, `MarvelObject$sample.metadata`, `MarvelObject$gene.metadata`, `MarvelObject$gene.count.matrix`, `MarvelObject$sj.count.matrix`, `MarvelObject$sj.metadata`.

Examples

```
# Load un-processed MARVEL object
marvel.demo.10x.raw <- readRDS(system.file("extdata/data",
                               "marvel.demo.10x.raw.rds",
                               package="MARVEL")
                               )

# Annotate gene metadata
marvel.demo.10x <- AnnotateGenes.10x(MarvelObject=marvel.demo.10x.raw)
```

```
# Annotate junction metadata
marvel.demo.10x <- AnnotateSJ.10x(MarvelObject=marvel.demo.10x)

# Validate junctions
marvel.demo.10x <- ValidateSJ.10x(MarvelObject=marvel.demo.10x)

# Subset CDS genes
marvel.demo.10x <- FilterGenes.10x(MarvelObject=marvel.demo.10x,
                          gene.type="protein_coding"
                          )

# Pre-flight check
marvel.demo.10x <- CheckAlignment.10x(MarvelObject=marvel.demo.10x)
```

`CheckAlignment.Exp` *Check gene data*

Description

Checks if the metadata aligns with the columns and rows of the matrix for gene data.

Usage

```
CheckAlignment.Exp(MarvelObject)
```

Arguments

`MarvelObject`	Marvel object. S3 object generated from `CreateMarvelObject` function.

Value

An object of class S3 with updated slots `MarvelObject$SplicePheno`, `MarvelObject$SpliceFeature`, and `MarvelObject$PSI`.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

marvel.demo <- CheckAlignment.Exp(MarvelObject=marvel.demo)
```

CheckAlignment.PSI *Check splicing data*

Description

Checks if the metadata aligns with the columns and rows of the matrix for splicing data.

Usage

```
CheckAlignment.PSI(MarvelObject)
```

Arguments

MarvelObject	Marvel object. S3 object generated from CreateMarvelObject function.

Value

An object of class S3 with updated slots MarvelObject$SplicePheno, MarvelObject$SpliceFeature, and MarvelObject$PSI.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

marvel.demo <- CheckAlignment.PSI(MarvelObject=marvel.demo)
```

CheckAlignment.PSI.Exp *Check splicing and gene data against each other*

Description

Subsets overlapping samples between splicing and gene data.

Usage

```
CheckAlignment.PSI.Exp(MarvelObject)
```

Arguments

MarvelObject	S3 object generated from CheckAlignment.PSI and CheckAlignment.Exp function.

Value

An object of class S3 with updated slots MarvelObject$SplicePheno, MarvelObject$PSI, MarvelObject$GenePheno, and MarvelObject$Exp.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

marvel.demo <- CheckAlignment.PSI.Exp(MarvelObject=marvel.demo)
```

`CheckAlignment.SJ` *Check splice junction data*

Description

Checks if the metadata aligns with the columns and rows of the matrix for splice junction data prior to PSI computation.

Usage

```
CheckAlignment.SJ(MarvelObject)
```

Arguments

`MarvelObject`	Marvel object. S3 object generated from `CreateMarvelObject` function.

Value

An object of class S3 with updated slots `MarvelObject$SplicePheno`, `MarvelObject$PSI` and `MarvelObject$IntronCounts`.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

marvel.demo <- CheckAlignment.SJ(MarvelObject=marvel.demo)
```

`CompareExpr` *Compares gene expression changes based on nonsense-mediated decay (NMD) status*

Description

Compares gene expression changes based on NMD status for each splicing event type.

Usage

```
CompareExpr(MarvelObject, xlabels.size = 8)
```

Arguments

MarvelObject	Marvel object. S3 object generated from FindPTC function.
xlabels.size	Numeric value. Size of the x-axis tick labels. Default is 8.

Value

An object of class S3 new slots MarvelObjectNMDNMD.Expr$Table, MarvelObject$NMD$NMD.Expr$Plot, and MarvelObjectNMDNMD.Expr$Plot.Stats.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

marvel.demo <- CompareExpr(MarvelObject=marvel.demo)

# Check outputs
head(marvel.demo$NMD$NMD.Expr$Table)
marvel.demo$NMD$NMD.Expr$Plot
marvel.demo$NMD$NMD.Expr$Plot.Stats
```

CompareValues *Differential splicing and gene expression analysis*

Description

Performs differential splicing and gene expression analysis between 2 groups of cells. This is a wrapper function for CompareValues.PSI and CompareValues.Exp functions.

Usage

```
CompareValues(
  MarvelObject,
  cell.group.g1 = NULL,
  cell.group.g2 = NULL,
  downsample = FALSE,
  seed = 1,
  min.cells = 25,
  pct.cells = NULL,
  method = NULL,
  nboots = 1000,
  n.permutations = 1000,
  method.adjust = "fdr",
  level,
  event.type = NULL,
  show.progress = TRUE,
  annotate.outliers = TRUE,
  n.cells.outliers = 10,
```

```
  assign.modality = TRUE,
  custom.gene_ids = NULL,
  psi.method = NULL,
  psi.pval = NULL,
  psi.delta = NULL,
  method.de.gene = NULL,
  method.adjust.de.gene = NULL,
  mast.method = "bayesglm",
  mast.ebayes = TRUE
)
```

Arguments

MarvelObject	Marvel object. S3 object generated from TransformExpValues function.
cell.group.g1	Vector of character strings. Cell IDs corresponding to Group 1 (reference group).
cell.group.g2	Vector of character strings. Cell IDs corresponding to Group 2.
downsample	Logical value. If set to TRUE, the number of cells in each cell group will be downsampled to the sample size of the smaller cell group so that both cell groups will have the sample size prior to differential expression analysis. Default is FALSE.
seed	Numeric value. The seed number for the random number generator to ensure reproducibility during during down-sampling of cells when downsample set to TRUE, during permutation testing when method set to "permutation", and during modality assignment which will be performed automatically.
min.cells	Numeric value. The minimum no. of cells expressing the splicing event or genes for the event or genes to be included for differential splicing analysis.
pct.cells	Numeric value. The minimum percentage of cells expressing the splicing event or genes for the event or genes to be included for differential splicing analysis. If pct.cells is specified, then pct.cells will be used as threshold instead of min.cells.
method	Character string. Statistical test to compare the 2 groups of cells. "ks", "kuiper", "ad", "dts", "wilcox", and "t.test" for Kolmogorov-Smirnov, Kuiper, Anderson-Darling, DTS, Wilcox, and t-test, respectively. Additional "mast" option is available for differential gene expression analysis. If "mast" is specified, the log2fc and p-values will be corrected using the gene detection rate as per the MAST package tutorial.
nboots	Numeric value.Only applicable when level set to "splicing". When method set to "dts", the number of bootstrap iterations for computing the p-value.
n.permutations	Numeric value. Only applicable when level set to "splicing". When method set to "permutation", this argument indicates the number of permutations to perform for generating the null distribution for subsequent p-value inference. Default is 1000 times.
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.

event.type Character string. Only applicable when level set to "splicing". 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.

show.progress Logical value. If set to TRUE, progress bar will be displayed so that users can estimate the time needed for differential analysis. Default value is TRUE.

annotate.outliers
Numeric value. Only applicable when level set to "splicing". When set to TRUE, statistical difference in PSI values between the two cell groups that is driven by outlier cells will be annotated.

n.cells.outliers
Numeric value. Only applicable when level set to "splicing". When method set to "dts", the minimum number of cells with non-1 or non-0 PSI values for included-to-included or excluded-to-excluded modality change, respectively. The p-values will be re-coded to 1 when both cell groups have less than this minimum number of cells. This is to avoid false positive results.

assign.modality
Logical value. Only applicable when level set to "splicing". If set to TRUE (default), modalities will be assigned to each cell group.

custom.gene_ids
Character string. Only applicable when level set to "gene". Instead of specified the genes to include for DE analysis with min.cells, users may input a custom vector of gene IDs to include for DE analysis.

psi.method Vector of character string(s). Only applicable when level set to "gene.spliced" and when CompareValues function has been ran with level set to "splicing" earlier. To include significant events from these method(s) for differential gene expression analysis.

psi.pval Vector of numeric value(s). Only applicable when level set to "gene.spliced" and when CompareValues function has been ran with level set to "splicing" earlier. The adjusted p-value, below which, the splicing event is considered differentially spliced, and the corresponding genes will be included for differential gene expression analysis.

psi.delta Numeric value. Only applicable when level set to "gene.spliced" and when CompareValues function has been ran with level set to "splicing" earlier. The absolute difference in mean PSI values between cell.group.g1 and cell.group.g1, above which, the splicing event is considered differentially spliced, and the corresponding genes will be included for differential gene expression analysis.

method.de.gene Character string. Only applicable when level set to "gene.spliced" and when CompareValues function has been ran with level set to "splicing" earlier. Same as method.

method.adjust.de.gene
Character string. Only applicable when level set to "gene.spliced" and when CompareValues function has been ran with level set to "splicing" earlier. Same as method.adjust.

mast.method — Character string. Only applicable when level set to "gene" or "gene.spliced". As per the method option of the zlm function from the MAST package. Default is "bayesglm", other options are "glm" and "glmer".

mast.ebayes — Logical value. Only applicable when level set to "gene" or "gene.spliced". As per the ebayes option of the zlm function from the MAST package. Default is TRUE.

Value

An object of class S3 containing with new slot MarvelObjectDEPSI$Table[["method"]] or MarvelObjectDEExp$Table when level option specified as "splicing" or "gene", respectively.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

# Define cell groups for analysis
df.pheno <- marvel.demo$SplicePheno
cell.group.g1 <- df.pheno[which(df.pheno$cell.type=="iPSC"), "sample.id"]
cell.group.g2 <- df.pheno[which(df.pheno$cell.type=="Endoderm"), "sample.id"]

# DE
marvel.demo <- CompareValues(MarvelObject=marvel.demo,
                             cell.group.g1=cell.group.g1,
                             cell.group.g2=cell.group.g2,
                             min.cells=5,
                             method="t.test",
                             method.adjust="fdr",
                             level="splicing",
                           event.type=c("SE", "MXE", "RI", "A5SS", "A3SS", "AFE", "ALE"),
                             show.progress=FALSE
                             )

# Check output
head(marvel.demo$DE$PSI$Table[["ad"]])
```

CompareValues.Exp *Differential gene expression analysis*

Description

Performs differential gene expression analysis between 2 groups of cells.

Usage

```
CompareValues.Exp(
  MarvelObject,
  cell.group.g1 = NULL,
```

```
  cell.group.g2 = NULL,
  downsample = FALSE,
  seed = 1,
  min.cells = 25,
  pct.cells = NULL,
  method,
  method.adjust,
  show.progress = TRUE,
  nboots = 1000,
  custom.gene_ids = NULL,
  mast.method = "bayesglm",
  mast.ebayes = TRUE
)
```

Arguments

MarvelObject	Marvel object. S3 object generated from TransformExpValues function.
cell.group.g1	Vector of character strings. Cell IDs corresponding to Group 1 (reference group).
cell.group.g2	Vector of character strings. Cell IDs corresponding to Group 2.
downsample	Logical value. If set to TRUE, the number of cells in each cell group will be downsampled to the sample size of the smaller cell group so that both cell groups will have the sample size prior to differential expression analysis. Default is FALSE.
seed	Numeric value. The seed number for the random number generator to ensure reproducibility during during down-sampling of cells when downsample set to TRUE.
min.cells	Numeric value. The minimum no. of cells expressing the gene for the gene to be included for differential splicing analysis.
pct.cells	Numeric value. The minimum no. of cells expressing the gene for the gene to be included for differential splicing analysis. If pct.cells is specified, then pct.cells will be used as threshold instead of min.cells.
method	Character string. Statistical test to compare the 2 groups of cells. "ks", "kuiper", "ad", "dts", "wilcox", and "t.test" for Kolmogorov-Smirnov, Kuiper, Anderson-Darling, DTS, Wilcox, and t-test, respectively. Additional option is "mast". If set to "mast" is specified, the log2fc and p-values will be corrected using the gene detection rate as per the MAST package tutorial.
method.adjust	Character string. Adjust p-values for multiple testing. Options available as per p.adjust function.
show.progress	Logical value. If set to TRUE, progress bar will be displayed so that users can estimate the time needed for differential analysis. Default value is TRUE.
nboots	Numeric value. When method set to "dts", the number of bootstrap iterations for computing the p-value.
custom.gene_ids	Character string. Instead of specified the genes to include for DE analysis with min.cells, users may input a custom vector of gene IDs to include for DE analysis.

mast.method — Character string. As per the method option of the zlm function from the MAST package. Default is "bayesglm", other options are "glm" and "glmer".

mast.ebayes — Logical value. As per the ebayes option of the zlm function from the MAST package. Default is TRUE.

Value

An object of class S3 new slot MarvelObjectDEExp$Table.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

# Define cell groups for analysis
df.pheno <- marvel.demo$SplicePheno
cell.group.g1 <- df.pheno[which(df.pheno$cell.type=="iPSC"), "sample.id"]
cell.group.g2 <- df.pheno[which(df.pheno$cell.type=="Endoderm"), "sample.id"]

# DE
marvel.demo <- CompareValues.Exp(MarvelObject=marvel.demo,
                                 cell.group.g1=cell.group.g1,
                                 cell.group.g2=cell.group.g2,
                                 min.cells=5,
                                 method="t.test",
                                 method.adjust="fdr",
                                 show.progress=FALSE
                                 )

# Check output
head(marvel.demo$DE$Exp$Table)
```

CompareValues.Exp.Spliced

Differential gene expression analysis for differentially spliced genes

---

Description

Performs differential gene expression analysis between 2 groups of cells only on differentially spliced genes.

Usage

```
CompareValues.Exp.Spliced(
  MarvelObject,
  cell.group.g1 = NULL,
  cell.group.g2 = NULL,
  psi.method,
  psi.pval,
  psi.delta,
```

```
  method.de.gene = "wilcox",
  method.adjust.de.gene = "fdr",
  downsample = FALSE,
  seed = 1,
  show.progress = TRUE,
  mast.method = "bayesglm",
  mast.ebayes = TRUE
)
```

Arguments

MarvelObject	Marvel object. S3 object generated from TransformExpValues function.
cell.group.g1	Vector of character strings. Cell IDs corresponding to Group 1 (reference group).
cell.group.g2	Vector of character strings. Cell IDs corresponding to Group 2.
psi.method	Vector of character string(s). To include significant events from these method(s) for differential gene expression analysis.
psi.pval	Vector of numeric value(s). The adjusted p-value, below which, the splicing event is considered differentially spliced, and the corresponding genes will be included for differential gene expression analysis.
psi.delta	Numeric value. The absolute difference in mean PSI values between cell.group.g1 and cell.group.g1, above which, the splicing event is considered differentially spliced, and the corresponding genes will be included for differential gene expression analysis.
method.de.gene	Character string. Same as method in CompareValues function.
method.adjust.de.gene	Character string. Same as method in CompareValues function.
downsample	Logical value. If set to TRUE, the number of cells in each cell group will be downsampled to the sample size of the smaller cell group so that both cell groups will have the sample size prior to differential expression analysis. Default is FALSE.
seed	Numeric value. The seed number for the random number generator to ensure reproducibility during during down-sampling of cells when downsample set to TRUE.
show.progress	Logical value. If set to TRUE, progress bar will be displayed so that users can estimate the time needed for differential analysis. Default value is TRUE.
mast.method	Character string. As per the method option of the zlm function from the MAST package. Default is "bayesglm", other options are "glm" and "glmer".
mast.ebayes	Logical value. As per the ebayes option of the zlm function from the MAST package. Default is TRUE.

Value

An object of class S3 new slot MarvelObjectDEExp$Table.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

# Define cell groups for analysis
df.pheno <- marvel.demo$SplicePheno
cell.group.g1 <- df.pheno[which(df.pheno$cell.type=="iPSC"), "sample.id"]
cell.group.g2 <- df.pheno[which(df.pheno$cell.type=="Endoderm"), "sample.id"]

# DE
marvel.demo <- CompareValues.Exp.Spliced(MarvelObject=marvel.demo,
                                         cell.group.g1=cell.group.g1,
                                         cell.group.g2=cell.group.g2,
                                         psi.method="ad",
                                         psi.pval=0.10,
                                         psi.delta=0,
                                         method.de.gene="t.test",
                                         method.adjust.de.gene="fdr",
                                         show.progress=FALSE
                                         )

# Check output
head(marvel.demo$DE$Exp.Spliced$Table)
```

CompareValues.Genes.10x

Differential gene expression analysis

Description

Performs differential gene expression analysis between two groups of cells. Only among cells and genes previously included for splice junction analysis.

Usage

```
CompareValues.Genes.10x(
  MarvelObject,
  log2.transform = TRUE,
  show.progress = TRUE,
  method = "wilcox",
  mast.method = "bayesglm",
  mast.ebayes = TRUE
)
```

Arguments

`MarvelObject`	Marvel object. S3 object generated from `CompareValues.SJ.10x` function.
`log2.transform`	Logical value. If set to `TRUE` (default), normalised gene expression values will be off-set by 1 and then log2-transformed prior to analysis. This option is automatically set to `TRUE` if `method` option is set to `"mast"`.

show.progress	Logical value. If set to TRUE (default), the progress bar will appear.
method	Character string. Statistical test to compare the 2 groups of cells. Default is "wilcox" as recommended by Seurat. Another option is "mast". If "mast" is specified, the log2fc and p-values will be corrected using the gene detection rate as per the MAST package tutorial.
mast.method	Character string. As per the method option of the zlm function from the MAST package. Default is "bayesglm", other options are "glm" and "glmer".
mast.ebayes	Logical value. As per the ebayes option of the zlm function from the MAST package. Default is TRUE.

Value

An object of class S3 with a updated slot MarvelObjectDESJ$Table.

Examples

```
marvel.demo.10x <- readRDS(system.file("extdata/data",
                                "marvel.demo.10x.rds",
                                package="MARVEL")
                                )

marvel.demo.10x <- CompareValues.Genes.10x(
                        MarvelObject=marvel.demo.10x,
                        show.progress=FALSE
                        )

# Check output
head(marvel.demo.10x$DE$SJ$Table)
```

CompareValues.PSI *Differential splicing analysis*

Description

Performs differentially splicing analysis between 2 groups of cells.

Usage

```
CompareValues.PSI(
  MarvelObject,
  cell.group.g1,
  cell.group.g2,
  downsample = FALSE,
  seed = 1,
  min.cells = 25,
  pct.cells = NULL,
```

```
  method,
  nboots = 1000,
  n.permutations = 1000,
  method.adjust = "fdr",
  event.type,
  show.progress = TRUE,
  annotate.outliers = TRUE,
  n.cells.outliers = 10,
  assign.modality = TRUE
)
```

Arguments

MarvelObject	Marvel object. S3 object generated from TransformExpValues function.
cell.group.g1	Vector of character strings. Cell IDs corresponding to Group 1 (reference group).
cell.group.g2	Vector of character strings. Cell IDs corresponding to Group 2.
downsample	Logical value. If set to TRUE, the number of cells in each cell group will be downsampled to the sample size of the smaller cell group so that both cell groups will have the sample size prior to differential expression analysis. Default is FALSE.
seed	Numeric value. The seed number for the random number generator to ensure reproducibility during during down-sampling of cells when downsample set to TRUE, during permutation testing when method set to "permutation", and during modality assignment which will be performed automatically.
min.cells	Numeric value. The minimum no. of cells expressing the splicing event for the event to be included for differential splicing analysis.
pct.cells	Numeric value. The minimum percentage of cells expressing the splicing event for the event to be included for differential splicing analysis. If pct.cells is specified, then pct.cells will be used as threshold instead of min.cells.
method	Character string. Statistical test to compare the 2 groups of cells. "ks", "kuiper", "ad", "dts", "wilcox", "t.test", and "permutation" for Kolmogorov-Smirnov, Kuiper, Anderson-Darling, DTS, Wilcox, t-test, and, permutation approach respectively.
nboots	Numeric value. When method set to "dts", the number of bootstrap iterations for computing the p-value.
n.permutations	Numeric value. When method set to "permutation", this argument indicates the number of permutations to perform for generating the null distribution for subsequent p-value inference. Default is 1000 times.
method.adjust	Character string. Adjust p-values for multiple testing. Options available as per p.adjust function.
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.
show.progress	Logical value. If set to TRUE, progress bar will be displayed so that users can estimate the time needed for differential analysis. Default value is TRUE.

annotate.outliers
: Numeric value. When set to TRUE, statistical difference in PSI values between the two cell groups that is driven by outlier cells will be annotated.

n.cells.outliers
: Numeric value. When annotate.outliers set to TRUE, the minimum number of cells with non-1 or non-0 PSI values for included-to-included or excluded-to-excluded modality change, respectively. The p-values will be re-coded to 1 when both cell groups have less than this minimum number of cells. This is to avoid false positive results.

assign.modality
: Logical value. If set to TRUE (default), modalities will be assigned to each cell group.

Value

An object of class data frame containing the output of the differential splicing analysis.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

# Define cell groups for analysis
df.pheno <- marvel.demo$SplicePheno
cell.group.g1 <- df.pheno[which(df.pheno$cell.type=="iPSC"), "sample.id"]
cell.group.g2 <- df.pheno[which(df.pheno$cell.type=="Endoderm"), "sample.id"]

# DE
results <- CompareValues.PSI(MarvelObject=marvel.demo,
                             cell.group.g1=cell.group.g1,
                             cell.group.g2=cell.group.g2,
                             min.cells=5,
                             method="t.test",
                             method.adjust="fdr",
                            event.type=c("SE", "MXE", "RI", "A5SS", "A3SS", "AFE", "ALE"),
                             show.progress=FALSE
                             )

# Check output
head(results)
```

CompareValues.SJ.10x *Differential splice junction analysis*

Description

Performs differential splice junction analysis between two groups of cells.

Usage

```
CompareValues.SJ.10x(
  MarvelObject,
  coord.introns = NULL,
  cell.group.g1,
  cell.group.g2,
  min.pct.cells.genes = 10,
  min.pct.cells.sj = 10,
  min.gene.norm = 1,
  seed = 1,
  n.iterations = 100,
  downsample = FALSE,
  show.progress = TRUE
)
```

Arguments

MarvelObject	Marvel object. S3 object generated from CheckAlignment.10x function.
coord.introns	Character strings. Specific splice junctions to be included for analysis. Default is NULL.
cell.group.g1	Vector of Character strings. Cell IDs corresponding to Group 1 (reference group).
cell.group.g2	Vector of Character strings. Cell IDs corresponding to Group 2.
min.pct.cells.genes	Numeric value. Minimum percentage of cells in which the gene is expressed for that gene to be included for splice junction expression distribution analysis. Expressed genes defined as genes with non-zero normalised UMI counts. This threshold may be determined from PlotPctExprCells.SJ.10x function. Default is 10.
min.pct.cells.sj	Numeric value. Minimum percentage of cells in which the splice junction is expressed for that splice junction to be included for splice junction expression distribution analysis. Expressed splice junctions defined as splice junctions with raw UMI counts >= 1. This threshold may be determined from PlotPctExprCells.SJ.10x function. Default is 10.
min.gene.norm	Numeric value. The average normalised gene expression across the two cell groups above which the splice junction will be included for analysis. Default is 1.0.
seed	Numeric value. Random number generator to be fixed for permutations test and down-sampling.
n.iterations	Numeric value. Number of times to shuffle the cell group labels when building the null distribution. Default is 100.
downsample	Logical value. If set to TRUE, both cell groups will be down-sampled so that both cell groups will have the same number of cells. The number of cells to downsample will be based on the smallest cell group. Default is FALSE.
show.progress	Logical value. If set to TRUE (default), the progress bar will appear.

Value

An object of class S3 with a new slots `MarvelObject$DE$SJ$Table`, `MarvelObject$DE$SJ$cell.group.g1`, and `MarvelObject$DE$SJ$cell.group.g2`.

Examples

```
marvel.demo.10x <- readRDS(system.file("extdata/data",
                                "marvel.demo.10x.rds",
                                package="MARVEL")
                                )

# Define cell groups
    # Retrieve sample metadata
    sample.metadata <- marvel.demo.10x$sample.metadata

    # Group 1 (reference)
    index <- which(sample.metadata$cell.type=="iPSC")
    cell.ids.1 <- sample.metadata[index, "cell.id"]
    length(cell.ids.1)

    # Group 2
    index <- which(sample.metadata$cell.type=="Cardio day 10")
    cell.ids.2 <- sample.metadata[index, "cell.id"]
    length(cell.ids.2)

# DE
marvel.demo.10x <- CompareValues.SJ.10x(
                        MarvelObject=marvel.demo.10x,
                        cell.group.g1=cell.ids.1,
                        cell.group.g2=cell.ids.2,
                        min.pct.cells.genes=10,
                        min.pct.cells.sj=10,
                        min.gene.norm=1.0,
                        seed=1,
                        n.iterations=100,
                        downsample=TRUE,
                        show.progress=FALSE
                        )

# Check output
head(marvel.demo.10x$DE$SJ$Table)
```

`ComputePSI` *Compute percent spliced-in (PSI) values*

Description

Validate splicing events and subsequently computes percent spliced-in (PSI) values these high-quality splicing events. This is a wrapper function for `ComputePSI.SE`, `ComputePSI.MXE`, `ComputePSI.A5SS`, `ComputePSI.A3SS`, `ComputePSI.RI`, `ComputePSI.AFE`, and `ComputePSI.ALE` functions.

Usage

```
ComputePSI(
  MarvelObject,
  CoverageThreshold,
  EventType,
  thread = NULL,
  UnevenCoverageMultiplier = 10,
  read.length = 1
)
```

Arguments

MarvelObject	Marvel object. 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.
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.
thread	Numeric value. Only applicable when EventType set to "RI" Set number of threads..
UnevenCoverageMultiplier	Numeric value. Maximum allowable fold difference between two included junction counts for SE or two included or two excluded junction counts for MXE. Only applicable when EventType set to "SE" or "MXE", respectively.
read.length	Numeric value. The length of read.Only applicable when EventType set to "RI". This number will be specific to the sequencing mode. E.g. read length should be set to 150 when samples were sequenced in 150bp paired-end or single-end. This option should only be specified when users used read-counting approach for computing intron counts. The option should be left with its default value 1 when users tabulated the per-base count and summed them up to arrive at the intron counts.

Value

An object of class S3 with new slots $SpliceFeatureValidated and $PSI.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

marvel.demo <- ComputePSI(MarvelObject=marvel.demo,
                          CoverageThreshold=10,
                          EventType="SE",
                          UnevenCoverageMultiplier=10
                          )
```

ComputePSI.A3SS *Compute Alternative 3' Splice Site (A3SS) Percent Spliced-in (PSI) Values*

Description

Validate A3SS splicing events and subsequently computes percent spliced-in (PSI) values these high-quality splicing events.

Usage

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

Arguments

MarvelObject	Marvel object. 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.

Value

An object of class S3 containing with new slots $SpliceFeatureValidated$A3SS and PSIA3SS.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

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

ComputePSI.A5SS *Compute alternative 5' splice site (A5SS) percent spliced-in (PSI) values*

Description

Validate A5SS splicing events and subsequently computes percent spliced-in (PSI) values these high-quality splicing events.

Usage

```
ComputePSI.A5SS(MarvelObject, CoverageThreshold)
```

Arguments

MarvelObject	Marvel object. 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.

Value

An object of class S3 with new slots $SpliceFeatureValidated$A5SS and PSIA5SS.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

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

ComputePSI.AFE *Compute alternative first exon (AFE) percent spliced-in (PSI) values*

Description

Computes percent spliced-in (PSI) for alternative first exon (ALE) splicing events.

Usage

```
ComputePSI.AFE(MarvelObject, CoverageThreshold = 10)
```

Arguments

MarvelObject	Marvel object. S3 object generated from DetectEvents 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.

Value

An object of class S3 containing with new slots $SpliceFeatureValidated$AFE and PSIAFE.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

marvel.demo <- ComputePSI.AFE(MarvelObject=marvel.demo,
                              CoverageThreshold=10
                              )
```

ComputePSI.ALE *Compute alternative last exon (ALE) percent spliced-in (PSI) values*

Description

Computes percent spliced-in (PSI) for alternative last exon (ALE) splicing events.

Usage

```
ComputePSI.ALE(MarvelObject, CoverageThreshold = 10)
```

Arguments

MarvelObject	Marvel object. S3 object generated from DetectEvents 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.

Value

An object of class S3 containing with new slots $SpliceFeatureValidated$ALE and PSIALE.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

marvel.demo <- ComputePSI.ALE(MarvelObject=marvel.demo,
                              CoverageThreshold=10
                              )
```

ComputePSI.MXE *Compute mutually exclusive exons (MXE) percent spliced-in (PSI) values*

Description

Validate MXE splicing events and subsequently computes percent spliced-in (PSI) values these high-quality splicing events.

Usage

```
ComputePSI.MXE(MarvelObject, CoverageThreshold, UnevenCoverageMultiplier = 10)
```

Arguments

MarvelObject	Marvel object. 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.
UnevenCoverageMultiplier	Numeric value. Maximum allowable fold difference between two included or two excluded junction counts for MXE.

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 with new slots $SpliceFeatureValidated$MXE and PSIMXE.

Author(s)

Sean Wen <sean.wenwx@gmail.com>

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

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

ComputePSI.RI *Compute retained-intron (RI) percent spliced-in (PSI) values*

Description

Validate RI splicing events and subsequently computes percent spliced-in (PSI) values these high-quality splicing events.

Usage

```
ComputePSI.RI(
  MarvelObject,
  CoverageThreshold,
  IntronCounts,
  thread,
  read.length = 1
)
```

Arguments

`MarvelObject`	Marvel object. 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.
`IntronCounts`	Data frame. Columns indicate sample IDs, rows indicate intron coordinates, and values indicate total intron coverage. The first column needs to be named `coord.intron`. These values will be combined with splice junction counts in the MARVEL object to compute PSI values.
`thread`	Numeric value. Set number of threads.
`read.length`	Numeric value. The length of read. This number will be specific to the sequencing mode. E.g. read length should be set to `150` when samples were sequenced in 150bp paired-end or single-end. This option should only be specified when users used read-counting approach for computing intron counts. The option should be left with its default value `1` when users tabulated the per-base count and summed them up to arrive at the intron counts.

Value

An object of class S3 with new slots `$SpliceFeatureValidated$RI` and `$PSI$RI`.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

marvel.demo <- ComputePSI.RI(MarvelObject=marvel.demo,
                             CoverageThreshold=10,
                             IntronCounts=marvel.demo$IntronCounts,
                             thread=1
                             )
```

ComputePSI.SE *Compute skipped-exon (SE) percent spliced-in (PSI) values*

Description

Validate SE splicing events and subsequently computes percent spliced-in (PSI) values these high-quality splicing events.

Usage

```
ComputePSI.SE(MarvelObject, CoverageThreshold, UnevenCoverageMultiplier = 10)
```

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.
UnevenCoverageMultiplier	Numeric value. Maximum allowable fold difference between two included junction counts.

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 with new slots $SpliceFeatureValidated$SE PSISE.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

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

CountEvents *Tabulate the number of expressed splicing events*

Description

Tabulates and plots the number of expressed splicing events for each splicing event category for a specified cell group.

Usage

```
CountEvents(MarvelObject, sample.ids, min.cells, event.group.colors = NULL)
```

Arguments

MarvelObject	Marvel object. S3 object generated from TransformExpValues function.
sample.ids	Vector of character strings. Sample IDs that constitute the cell group.
min.cells	Numeric value. Minimum number of cells expressing the splicing event for the event to be included for tabulation. A splicing event is defined as expressed when it has a non-missing PSI value.
event.group.colors	Vector of character strings. Colors for the event groups. If not specified, default ggplot2 colors will be used.

Value

An object of class S3 with new slots MarvelObject$N.Events$Table and MarvelObject$N.Events$Plot.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

# Define cell group for analysis
df.pheno <- marvel.demo$SplicePheno
sample.ids <- df.pheno[which(df.pheno$cell.type=="iPSC"), "sample.id"]

# Tabulate expressed events
marvel.demo <- CountEvents(MarvelObject=marvel.demo,
                           sample.ids=sample.ids,
                           min.cells=5,
                           event.group.colors=NULL
                           )

# Check outputs
marvel.demo$N.Events$Table
marvel.demo$N.Events$Plot
```

`CreateMarvelObject` *Create Marvel object for plate-based RNA-sequencing data*

Description

Creates an S3 object named `Marvel` for downstream analysis, specifically for plate-based RNA-sequencing data.

Usage

```
CreateMarvelObject(
  SplicePheno = NULL,
  SpliceJunction = NULL,
  IntronCounts = NULL,
  SpliceFeature = NULL,
  SpliceFeatureValidated = NULL,
  PSI = NULL,
  GeneFeature = NULL,
  Exp = NULL,
  GTF = NULL
)
```

Arguments

`SplicePheno`	Data frame. Sample metadata.
`SpliceJunction`	Data frame. Splice junction counts matrix.
`IntronCounts`	Data frame. Intron coverage matrix.
`SpliceFeature`	List of data frames. Each data frame is the exon-level alternative splicing event metadata.
`SpliceFeatureValidated`	List of data frames. Each data frame is the validated (high-quality) exon-level alternative splicing event metadata.
`PSI`	Data frame. PSI matrix.
`GeneFeature`	Data frame. Gene metadata.
`Exp`	Data frame. Normalised, non-log2-transformed gene expression matrix.
`GTF`	Data frame. GTF used for generating the exon-level alternative splicing event metadata.

Value

An object of class S3.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

SpliceJunction <- marvel.demo$SpliceJunction
SpliceJunction[1:5,1:5]

SplicePheno <- marvel.demo$SplicePheno
SplicePheno[1:5,]

SpliceFeature <- marvel.demo$SpliceFeature
SpliceFeature[["SE"]][1:5, ]

IntronCounts <- marvel.demo$IntronCounts
IntronCounts[1:5,1:5]

GeneFeature <- marvel.demo$GeneFeature
GeneFeature[1:5, ]

Exp <- marvel.demo$Exp
Exp[1:5,1:5]

marvel <- CreateMarvelObject(SpliceJunction=SpliceJunction,
                             SplicePheno=SplicePheno,
                             SpliceFeature=SpliceFeature,
                             IntronCounts=IntronCounts,
                             GeneFeature=GeneFeature,
                             Exp=Exp
                             )
class(marvel)
```

CreateMarvelObject.10x

Create Marvel object for droplet-based RNA-sequencing data

Description

Creates an S3 object named `Marvel` for downstream analysis, specifically for droplet-based RNA-sequencing data.

Usage

```
CreateMarvelObject.10x(
  gene.norm.matrix = NULL,
  gene.norm.pheno = NULL,
  gene.norm.feature = NULL,
  gene.count.matrix = NULL,
  gene.count.pheno = NULL,
  gene.count.feature = NULL,
  sj.count.matrix = NULL,
```

```
  sj.count.pheno = NULL,
  sj.count.feature = NULL,
  pca = NULL,
  gtf = NULL
)
```

Arguments

gene.norm.matrix	Sparse matrix. UMI-collapsed, normalised, non-log2-transformed gene expression matrix.
gene.norm.pheno	Data frame. Sample metadata for annotating gene.norm.matrix columns with cell IDs.
gene.norm.feature	Data frame. Gene metadata for annotating gene.norm.matrix rows with gene names.
gene.count.matrix	Sparse matrix. UMI-collapsed, non-normalised (raw counts), non-log2-transformed gene expression matrix.
gene.count.pheno	Data frame. Sample metadata for annotating gene.count.matrix columsn with cell IDs.
gene.count.feature	Data frame. Gene metadata for annotating gene.count.matrix rows with gene names.
sj.count.matrix	Sparse matrix. UMI-collapsed, non-normalised (raw counts), non-log2-transformed splice junction expression matrix.
sj.count.pheno	Data frame. Sample metadata for annotating sj.count.matrix columsn with cell IDs.
sj.count.feature	Data frame. Splice junction metadata for annotating sj.count.matrix rows with splice junction coordinates.
pca	Data frame. Coordinates of PCA/tSNE/UMAP.
gtf	Data frame. GTF used in cellranger. Will be used for annotating splice junctions downstream.

Value

An object of class S3.

Examples

```
# Retrieve, observe format of pre-saved input files
marvel.demo.10x.raw <- readRDS(system.file("extdata/data",
                               "marvel.demo.10x.raw.rds",
                               package="MARVEL")
```

```
                                )
# Gene expression (Normalised)
    # Matrix
    df.gene.norm <- marvel.demo.10x.raw$gene.norm.matrix
    df.gene.norm[1:5, 1:5]

    # phenoData
    df.gene.norm.pheno <- marvel.demo.10x.raw$sample.metadata
    head(df.gene.norm.pheno)

    # featureData
    df.gene.norm.feature <- data.frame("gene_short_name"=rownames(df.gene.norm),
                                       stringsAsFactors=FALSE
                                       )
    head(df.gene.norm.feature)

# Gene expression (Counts)
    # Matrix
    df.gene.count <- marvel.demo.10x.raw$gene.count.matrix
    df.gene.count[1:5, 1:5]

    # phenoData
    df.gene.count.pheno <- data.frame("cell.id"=colnames(df.gene.count),
                                       stringsAsFactors=FALSE
                                       )
    head(df.gene.count.pheno)

    # featureData
    df.gene.count.feature <- data.frame("gene_short_name"=rownames(df.gene.count),
                                       stringsAsFactors=FALSE
                                       )
    head(df.gene.count.feature)

# SJ (Counts)
    # Matrix
    df.sj.count <- marvel.demo.10x.raw$sj.count.matrix
    df.sj.count[1:5, 1:5]

    # phenoData
    df.sj.count.pheno <- data.frame("cell.id"=colnames(df.sj.count),
                                    stringsAsFactors=FALSE
                                    )
    head(df.sj.count.pheno)

    # featureData
    df.sj.count.feature <- data.frame("coord.intron"=rownames(df.sj.count),
                                       stringsAsFactors=FALSE
                                       )
    head(df.sj.count.feature)

# tSNE coordinates
df.coord <- marvel.demo.10x.raw$pca
head(df.coord)
```

```
# GTF
gtf <- marvel.demo.10x.raw$gtf
head(gtf)

# Create MARVEL object
marvel.demo.10x <- CreateMarvelObject.10x(gene.norm.matrix=df.gene.norm,
                     gene.norm.pheno=df.gene.norm.pheno,
                     gene.norm.feature=df.gene.norm.feature,
                     gene.count.matrix=df.gene.count,
                     gene.count.pheno=df.gene.count.pheno,
                     gene.count.feature=df.gene.count.feature,
                     sj.count.matrix=df.sj.count,
                     sj.count.pheno=df.sj.count.pheno,
                     sj.count.feature=df.sj.count.feature,
                     pca=df.coord,
                     gtf=gtf
                     )
```

`DetectEvents` *Detect Splicing Events*

Description

Detects splicing events, specifically alternative first and last exons (AFE, ALE) from GTF. This is a wrapper function for `DetectEvents.ALE` and `DetectEvents.AFE` functions.

Usage

```
DetectEvents(
  MarvelObject,
  min.cells = 50,
  min.expr = 1,
  track.progress = FALSE,
  EventType
)
```

Arguments

`MarvelObject`	Marvel object. S3 object generated from `CreateMarvelObject` function.
`min.cells`	Numeric value. The minimum number of cells in which the gene is expressed for the gene to included for splicing event detected and quantification. To be used in conjunction with `min.expr` argument. Default value is `50`.
`min.expr`	Numeric value. The minimum expression value for the gene to be considered to be expressed in a cell. Default value is `1`.
`track.progress`	Logical. If set to `TRUE`, progress bar will appear to track the progress of the rate-limiting step of this function, which is the extraction of the final exon-exon junctions. Default value is `FALSE`. Only applicable when `EventType` set to `"ALE"` or `"AFE"`.

EventType	Character string. Indicate which splicing event type to calculate the PSI values for. Can take value "ALE" or "AFE".

Value

An object of class S3 with new slot MarvelObject$SpliceFeature$ALE or MarvelObject$SpliceFeature$AFE.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

marvel.demo <- DetectEvents(MarvelObject=marvel.demo,
                            min.cells=5,
                            min.expr=1,
                            track.progress=FALSE,
                            EventType="AFE"
                            )
```

DetectEvents.AFE *Detect alternative first exons*

Description

Detects alternative first exons from GTF. This is a wrapper function for DetectEvents.AFE.PosStrand and DetectEvents.AFE.NegStrand functions.

Usage

```
DetectEvents.AFE(
  MarvelObject,
  min.cells = 50,
  min.expr = 1,
  track.progress = FALSE
)
```

Arguments

MarvelObject	Marvel object. S3 object generated from CreateMarvelObject function.
min.cells	Numeric value. The minimum number of cells in which the gene is expressed for the gene to included for splicing event detected and quantification. To be used in conjunction with min.expr argument. Default value is 50.
min.expr	Numeric value. The minimum expression value for the gene to be considered to be expressed in a cell. Default value is 1.
track.progress	Logical. If set to TRUE, progress bar will appear to track the progress of the rate-limiting step of this function, which is the extraction of the final exon-exon junctions. Default value is FALSE.

Value

An object of class S3 with new slot `MarvelObject$SpliceFeature$AFE`.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

marvel.demo <- DetectEvents.AFE(MarvelObject=marvel.demo,
                                min.cells=5,
                                min.expr=1,
                                track.progress=FALSE
                                )
```

DetectEvents.AFE.NegStrand

Detect alternative first exons on negative strand

Description

Detects alternative first exons, specifically for genes transcribed on the negative strand of the DNA.

Usage

```
DetectEvents.AFE.NegStrand(
  MarvelObject,
  parsed.gtf = NULL,
  min.cells = 50,
  min.expr = 1,
  track.progress = FALSE
)
```

Arguments

`MarvelObject`	S3 object generated from `CreateMarvelObject` function.
`parsed.gtf`	Data frame. GTF file with the gene_id parsed. Generated from the `DetectEvents.AFE` function.
`min.cells`	Numeric value. The minimum number of cells in which the gene is expressed for the gene to included for splicing event detected and quantification. To be used in conjunction with `min.expr` argument. Default value is `50`.
`min.expr`	Numeric value. The minimum expression value for the gene to be considered to be expressed in a cell. Default value is `1`.
`track.progress`	Logical. If set to `TRUE`, progress bar will appear to track the progress of the rate-limiting step of this function, which is the extraction of the final exon-exon junctions. Default value is `FALSE`.

Value

An object of class S3 with new slot `MarvelObject$SpliceFeature$AFE.NegStrand`.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

marvel.demo <- DetectEvents.AFE.NegStrand(MarvelObject=marvel.demo,
                                          parsed.gtf=NULL,
                                          min.cells=5,
                                          min.expr=1,
                                          track.progress=FALSE
                                          )
```

DetectEvents.AFE.PosStrand

Detect alternative first exons on positive strand

Description

Detects alternative first exons, specifically for genes transcribed on the positive strand of the DNA.

Usage

```
DetectEvents.AFE.PosStrand(
  MarvelObject,
  parsed.gtf = NULL,
  min.cells = 50,
  min.expr = 1,
  track.progress = FALSE
)
```

Arguments

`MarvelObject`	S3 object generated from `CreateMarvelObject` function.
`parsed.gtf`	Data frame. GTF file with the gene_id parsed. Generated from the `DetectEvents.AFE` function.
`min.cells`	Numeric value. The minimum number of cells in which the gene is expressed for the gene to included for splicing event detected and quantification. To be used in conjunction with `min.expr` argument. Default value is `50`.
`min.expr`	Numeric value. The minimum expression value for the gene to be considered to be expressed in a cell. Default value is `1`.
`track.progress`	Logical. If set to `TRUE`, progress bar will appear to track the progress of the rate-limiting step of this function, which is the extraction of the final exon-exon junctions. Default value is `FALSE`.

Value

An object of class S3 with new slot `MarvelObject$SpliceFeature$AFE.PosStrand`.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

marvel.demo <- DetectEvents.AFE.PosStrand(MarvelObject=marvel.demo,
                                          parsed.gtf=NULL,
                                          min.cells=5,
                                          min.expr=1,
                                          track.progress=FALSE
                                          )
```

DetectEvents.ALE *Detect alternative last exons*

Description

Detects alternative last exons from GTF. This is a wrapper function for `DetectEvents.ALE.PosStrand` and `DetectEvents.ALE.NegStrand` functions.

Usage

```
DetectEvents.ALE(
  MarvelObject,
  min.cells = 50,
  min.expr = 1,
  track.progress = FALSE
)
```

Arguments

`MarvelObject`	Marvel object. S3 object generated from `CreateMarvelObject` function.
`min.cells`	Numeric value. The minimum number of cells in which the gene is expressed for the gene to included for splicing event detected and quantification. To be used in conjunction with `min.expr` argument. Default value is `50`.
`min.expr`	Numeric value. The minimum expression value for the gene to be considered to be expressed in a cell. Default value is `1`.
`track.progress`	Logical. If set to `TRUE`, progress bar will appear to track the progress of the rate-limiting step of this function, which is the extraction of the final exon-exon junctions. Default value is `FALSE`.

Value

An object of class S3 with new slot `MarvelObject$SpliceFeature$ALE`.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

marvel.demo <- DetectEvents.ALE(MarvelObject=marvel.demo,
                                min.cells=5,
                                min.expr=1,
                                track.progress=FALSE
                                )
```

DetectEvents.ALE.NegStrand

Detect alternative last exons on negative strand

Description

Detects alternative last exons, specifically for genes transcribed on the negative strand of the DNA.

Usage

```
DetectEvents.ALE.NegStrand(
  MarvelObject,
  parsed.gtf = NULL,
  min.cells = 50,
  min.expr = 1,
  track.progress = FALSE
)
```

Arguments

MarvelObject	S3 object generated from CreateMarvelObject function.
parsed.gtf	Data frame. GTF file with the gene_id parsed. Generated from the DetectEvents.ALE function.
min.cells	Numeric value. The minimum number of cells in which the gene is expressed for the gene to included for splicing event detected and quantification. To be used in conjunction with min.expr argument. Default value is 50.
min.expr	Numeric value. The minimum expression value for the gene to be considered to be expressed in a cell. Default value is 1.
track.progress	Logical. If set to TRUE, progress bar will appear to track the progress of the rate-limiting step of this function, which is the extraction of the final exon-exon junctions. Default value is FALSE.

Value

An object of class S3 with new slot MarvelObject$SpliceFeature$ALE.NegStrand.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

marvel.demo <- DetectEvents.ALE.NegStrand(MarvelObject=marvel.demo,
                                          parsed.gtf=NULL,
                                          min.cells=5,
                                          min.expr=1,
                                          track.progress=FALSE
                                          )
```

DetectEvents.ALE.PosStrand

Detect alternative last exons on positive strand

Description

Detects alternative last exons, specifically for genes transcribed on the positive strand of the DNA.

Usage

```
DetectEvents.ALE.PosStrand(
  MarvelObject,
  parsed.gtf = NULL,
  min.cells = 50,
  min.expr = 1,
  track.progress = FALSE
)
```

Arguments

`MarvelObject`	S3 object generated from `CreateMarvelObject` function.
`parsed.gtf`	Data frame. GTF file with the gene_id parsed. Generated from the `DetectEvents.ALE` function.
`min.cells`	Numeric value. The minimum number of cells in which the gene is expressed for the gene to included for splicing event detected and quantification. To be used in conjunction with `min.expr` argument. Default value is `50`.
`min.expr`	Numeric value. The minimum expression value for the gene to be considered to be expressed in a cell. Default value is `1`.
`track.progress`	Logical. If set to `TRUE`, progress bar will appear to track the progress of the rate-limiting step of this function, which is the extraction of the final exon-exon junctions. Default value is `FALSE`.

Value

An object of class S3 with new slot `MarvelObject$SpliceFeature$ALE.PosStrand`.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

marvel.demo <- DetectEvents.ALE.PosStrand(MarvelObject=marvel.demo,
                                          parsed.gtf=NULL,
                                          min.cells=5,
                                          min.expr=1,
                                          track.progress=FALSE
                                          )
```

`FilterGenes.10x` *Filter specific gene types*

Description

Retain genes of specific type, e.g., protein-coding genes.

Usage

```
FilterGenes.10x(MarvelObject, gene.type = "protein_coding")
```

Arguments

`MarvelObject`	Marvel object. S3 object generated from `AnnotateGenes.10x` function.
`gene.type`	Character string. Gene type to keep. Specification should match that of GTF.

Value

An object of class S3 containing the updated slots `MarvelObject$gene.metadata`, `MarvelObject$gene.norm.matrix`, `MarvelObject$sj.metadata`, and `MarvelObject$sj.count.matrix`.

Examples

```
# Load un-processed MARVEL object
marvel.demo.10x.raw <- readRDS(system.file("extdata/data",
                               "marvel.demo.10x.raw.rds",
                               package="MARVEL")
                               )

# Annotate gene metadata
marvel.demo.10x <- AnnotateGenes.10x(MarvelObject=marvel.demo.10x.raw)

# Annotate junction metadata
marvel.demo.10x <- AnnotateSJ.10x(MarvelObject=marvel.demo.10x)

# Validate junctions
marvel.demo.10x <- ValidateSJ.10x(MarvelObject=marvel.demo.10x)
```

```
# Subset CDS genes
marvel.demo.10x <- FilterGenes.10x(MarvelObject=marvel.demo.10x,
                          gene.type="protein_coding"
                          )
```

FindPTC *Find premature terminal codons (PTCs)*

Description

Finds PTC(s) introduced by alternative exons into protein-coding transcripts.

Usage

```
FindPTC(MarvelObject, method, pval, delta, custom.tran_ids = NULL)
```

Arguments

MarvelObject	Marvel object. S3 object generated from CompareValues.PSI and ParseGTF functions.
method	Character string. The statistical method used for differential splicing analysis.
pval	Numeric value. Adjusted p-value below which the splicing event will be analysed for PTCs.
delta	Numeric value. Positive delta percent spliced-in (PSI) value above which the splicing event will be analysed for PTCs. "Positive" because only an increased in PSI value leads to increased alternative exon inclusion in the transcript.
custom.tran_ids	Vector of character strings. Subset of tran_ids to be brought forward for analysis after filtering based on pval and delta.

Value

An object of class S3 with new slot MarvelObjectNMDPrediction.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

marvel.demo <- FindPTC(MarvelObject=marvel.demo,
                       method="ad",
                       pval=0.1,
                       delta=90
                       )
```

FindPTC.A3SS.NegStrand

Find premature terminal codon (PTC) for alternative 3' splice site (A3SS) located on the negative strand of the transcript

Description

Finds PTC(s) introduced by alternative exons into protein-coding transcripts.

Usage

```
FindPTC.A3SS.NegStrand(MarvelObject, tran_id, gene_id)
```

Arguments

MarvelObject	Marvel object. S3 object generated from CompareValues.PSI and ParseGTF function.
tran_id	Character string. Vector of tran_id to look for PTCs.
gene_id	Character string. Vector of gene_id corresponding to the tran_id argument.

Value

A data frame of transcripts containing splicing events meeting the psi.de.sig and psi.de.diff criteria are categorised based on the presence or absence of PTCs.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

# Define relevant event type
results <- marvel.demo$DE$PSI$Table[["ad"]]
index.1 <- which(results$event_type=="A3SS")
index.2 <- grep(":-@", results$tran_id, fixed=TRUE)
index <- intersect(index.1, index.2)
results <- results[index, ]
tran_id <- results$tran_id[3]
gene_id <- results$gene_id[3]

# Find PTC
results <- FindPTC.A3SS.NegStrand(MarvelObject=marvel.demo,
                                  tran_id=NULL,
                                  gene_id=gene_id
                                  )

# Check output
head(results)
```

FindPTC.A3SS.PosStrand *Find premature terminal codon (PTC) for alternative 3' splice site (A3SS) located on the positive strand of the transcript*

Description

Finds PTC(s) introduced by alternative exons into protein-coding transcripts.

Usage

```
FindPTC.A3SS.PosStrand(MarvelObject, tran_id, gene_id)
```

Arguments

MarvelObject	Marvel object. S3 object generated from CompareValues.PSI and ParseGTF function.
tran_id	Character string. Vector of tran_id to look for PTCs.
gene_id	Character string. Vector of gene_id corresponding to the tran_id argument.

Value

A data frame of transcripts containing splicing events meeting the psi.de.sig and psi.de.diff criteria are categorised based on the presence or absence of PTCs.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

# Define relevant event type
results <- marvel.demo$DE$PSI$Table[["ad"]]
index.1 <- which(results$event_type=="A3SS")
index.2 <- grep(":+@", results$tran_id, fixed=TRUE)
index <- intersect(index.1, index.2)
results <- results[index, ]
tran_id <- results$tran_id[1]
gene_id <- results$gene_id[1]

# Find PTC
results <- FindPTC.A3SS.PosStrand(MarvelObject=marvel.demo,
                                  tran_id=tran_id,
                                  gene_id=gene_id
                                  )

# Check output
head(results)
```

FindPTC.A5SS.NegStrand *Find premature terminal codon (PTC) for alternative 5' splice site (A5SS) located on the negative strand of the transcript*

Description

Finds PTC(s) introduced by alternative exons into protein-coding transcripts.

Usage

```
FindPTC.A5SS.NegStrand(MarvelObject, tran_id, gene_id)
```

Arguments

MarvelObject	Marvel object. S3 object generated from CompareValues.PSI and ParseGTF function.
tran_id	Character string. Vector of tran_id to look for PTCs.
gene_id	Character string. Vector of gene_id corresponding to the tran_id argument.

Value

A data frame of transcripts containing splicing events meeting the psi.de.sig and psi.de.diff criteria are categorised based on the presence or absence of PTCs.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

results <- marvel.demo$DE$PSI$Table[["ad"]]
index.1 <- which(results$event_type=="A5SS")
index.2 <- grep(":-@", results$tran_id, fixed=TRUE)
index <- intersect(index.1, index.2)
results <- results[index, ]
tran_id <- results$tran_id[1]
gene_id <- results$gene_id[1]

# Find PTC
results <- FindPTC.A5SS.NegStrand(MarvelObject=marvel.demo,
                                  tran_id=tran_id,
                                  gene_id=gene_id
                                  )

# Check output
head(results)
```

FindPTC.A5SS.PosStrand *Find premature terminal codon (PTC) for alternative 5' splice site (A5SS) located on the positive strand of the transcript*

Description

Finds PTC(s) introduced by alternative exons into protein-coding transcripts.

Usage

```
FindPTC.A5SS.PosStrand(MarvelObject, tran_id, gene_id)
```

Arguments

MarvelObject	S3 object generated from CompareValues.PSI and ParseGTF function.
tran_id	Character string. Vector of tran_id to look for PTCs.
gene_id	Character string. Vector of gene_id corresponding to the tran_id argument.

Value

A data frame of transcripts containing splicing events meeting the psi.de.sig and psi.de.diff criteria are categorised based on the presence or absence of PTCs.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

# Define relevant event type
results <- marvel.demo$DE$PSI$Table[["ad"]]
index.1 <- which(results$event_type=="A5SS")
index.2 <- grep(":+@", results$tran_id, fixed=TRUE)
index <- intersect(index.1, index.2)
results <- results[index, ]
tran_id <- results$tran_id[1]
gene_id <- results$gene_id[1]

# Find PTC
results <- FindPTC.A5SS.PosStrand(MarvelObject=marvel.demo,
                                  tran_id=tran_id,
                                  gene_id=gene_id
                                  )

# Check output
head(results)
```

`FindPTC.RI.NegStrand` *Find premature terminal codon (PTC) for retained-intron (RI) located on the negative strand of the transcript*

Description

Finds PTC(s) introduced by alternative exons into protein-coding transcripts.

Usage

```
FindPTC.RI.NegStrand(MarvelObject, tran_id, gene_id)
```

Arguments

`MarvelObject`	S3 object generated from `CompareValues.PSI` and `ParseGTF` function.
`tran_id`	Character string. Vector of `tran_id` to look for PTCs.
`gene_id`	Character string. Vector of `gene_id` corresponding to the `tran_id` argument.

Value

A data frame of transcripts containing splicing events meeting the `psi.de.sig` and `psi.de.diff` criteria are categorised based on the presence or absence of PTCs.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

# Define relevant event type
results <- marvel.demo$DE$PSI$Table[["ad"]]
index.1 <- which(results$event_type=="RI")
index.2 <- grep(":-@", results$tran_id, fixed=TRUE)
index <- intersect(index.1, index.2)
results <- results[index, ]
tran_id <- results$tran_id[1]
gene_id <- results$gene_id[1]

# Find PTC
results <- FindPTC.RI.NegStrand(MarvelObject=marvel.demo,
                                tran_id=tran_id,
                                gene_id=gene_id
                                )

# Check output
head(results)
```

FindPTC.RI.PosStrand *Find premature terminal codon (PTC) for retained-Intron (RI) located on the positive strand of the transcript*

Description

Finds PTC(s) introduced by alternative exons into protein-coding transcripts.

Usage

```
FindPTC.RI.PosStrand(MarvelObject, tran_id, gene_id)
```

Arguments

MarvelObject	Marvel object. S3 object generated from CompareValues.PSI and ParseGTF function.
tran_id	Character string. Vector of tran_id to look for PTCs.
gene_id	Character string. Vector of gene_id corresponding to the tran_id argument.

Value

A data frame of transcripts containing splicing events meeting the psi.de.sig and psi.de.diff criteria are categorised based on the presence or absence of PTCs.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

# Define relevant event type
results <- marvel.demo$DE$PSI$Table[["ad"]]
index.1 <- which(results$event_type=="RI")
index.2 <- grep(":+@", results$tran_id, fixed=TRUE)
index <- intersect(index.1, index.2)
results <- results[index, ]
tran_id <- results$tran_id[1]
gene_id <- results$gene_id[1]

# Find PTC
results <- FindPTC.RI.PosStrand(MarvelObject=marvel.demo,
                                tran_id=tran_id,
                                gene_id=gene_id
                                )

# Check output
head(results)
```

FindPTC.SE.NegStrand *Find premature terminal codon (PTC) for skipped-exon (SE) located on the negative Strand of the yranscript*

Description

Finds PTC(s) introduced by alternative exons into protein-coding transcripts.

Usage

```
FindPTC.SE.NegStrand(MarvelObject, tran_id, gene_id)
```

Arguments

MarvelObject	Marvel object. S3 object generated from CompareValues.PSI and ParseGTF function.
tran_id	Character string. Vector of tran_id to look for PTCs.
gene_id	Character string. Vector of gene_id corresponding to the tran_id argument.#'

Value

A data frame of transcripts containing splicing events meeting the psi.de.sig and psi.de.diff criteria are categorised based on the presence or absence of PTCs.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

# Define relevant event type
results <- marvel.demo$DE$PSI$Table[["ad"]]
index.1 <- which(results$event_type=="SE")
index.2 <- grep(":-@", results$tran_id, fixed=TRUE)
index <- intersect(index.1, index.2)
results <- results[index, ]
tran_id <- results$tran_id[1]
gene_id <- results$gene_id[1]

# Find PTC
results <- FindPTC.SE.NegStrand(MarvelObject=marvel.demo,
                                tran_id=NULL,
                                gene_id=gene_id
                                )

# Check output
head(results)
```

`FindPTC.SE.PosStrand` *Find premature terminal codon (PTC) for skipped-exon (SE) located on the positive strand of the transcript*

Description

Finds PTC(s) introduced by alternative exons into protein-coding transcripts.

Usage

```
FindPTC.SE.PosStrand(MarvelObject, tran_id, gene_id)
```

Arguments

`MarvelObject`	S3 object generated from `CompareValues.PSI` and `ParseGTF` function.
`tran_id`	Character string. Vector of `tran_id` to look for PTCs.
`gene_id`	Character string. Vector of `gene_id` corresponding to the `tran_id` argument.

Details

This function finds PTC(s) introduced by alternative exons into protein-coding transcripts. It also records the distance between a PTCs and the final splice junction for a given protein-coding transcript. Non-protein-coding transcripts or transcripts in which splicing events are located outside of the transcripts' open-reading frame (ORF) are not analysed for PTCs but are noted.

Value

A data frame of transcripts containing splicing events meeting the `psi.de.sig` and `psi.de.diff` criteria are categorised based on the presence or absence of PTCs.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

# Define relevant event type
results <- marvel.demo$DE$PSI$Table[["ad"]]
index.1 <- which(results$event_type=="SE")
index.2 <- grep(":+@", results$tran_id, fixed=TRUE)
index <- intersect(index.1, index.2)
results <- results[index, ]
tran_id <- results$tran_id[1]
gene_id <- results$gene_id[1]

# Find PTC
results <- FindPTC.SE.PosStrand(MarvelObject=marvel.demo,
                                tran_id=NULL,
                                gene_id=gene_id
                                )
```

```
# Check output
head(results)
```

`IsoSwitch` *Classify gene-splicing relationship*

Description

Classify gene-splicing relative changes to each other from cell group 1 to group 2. Classifications are coordinated, opposing, isoform-switching, and complex. In coordinated relationship, both gene and splicing changes in the same direction from cell group 1 to group 2. In opposing relationship, gene changes in the opposite direction relative to splicing from cell group 1 to group 2. In isoform-switching, there is differential splice junction usage without differential expression of the corresponding gene between cell group 1 and group 2. Complex relationship involves genes with both coordinated and opposing relationships with splicing. Only differentially spliced junctions are included for analysis here.

Usage

```
IsoSwitch(
  MarvelObject,
  method,
  psi.pval = 0.1,
  psi.delta = 0,
  gene.pval = 0.1,
  gene.log2fc = 0.5,
  event.type = NULL,
  custom.tran_ids = NULL
)
```

Arguments

`MarvelObject`	Marvel object. S3 object generated from `CompareValues.Genes.10x` function.
`method`	Character string. The statistical method used for differential splicing analysis.
`psi.pval`	Numeric value. Adjusted p-value below which the splicing event is considered differentially spliced and included for isoform switching analysis. To be used in conjunction with `psi.delta`.
`psi.delta`	Numeric value. The absolute mininum difference in PSI values between the two cell groups above which the splicing event is considered differentially spliced nd included for isoform switching analysis. To be used in conjunction with `psi.pval`. Specify `0` (default) to switch this threshold off.
`gene.pval`	Numeric value. Adjusted p-value below which the gene is considered differentially expressed. Default value is `0.1`.

gene.log2fc — Numeric value. The absolute log2 fold change in mean gene expression values between the two cell groups above which the gene is considered differentially expressed. To be used in conjunction with gene.pval. Specify 0 to switch this threshold off. Default value is 0.5.

event.type — Character string. Indicate which splicing event type to include for analysis. Can take any combination of values: "SE", "MXE", "RI", "A5SS", "A3SS", "AFE}, or \code{"ALE.

custom.tran_ids — Vector of character strings. Subset of tran_ids to be brought forward for analysis after filtering based on psi.pval and psi.delta.

Value

An object of class S3 containing with new slots MarvelObjectDECor$Table, MarvelObject$DECorPlot, and MarvelObjectDECor$Plot.Stats.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

marvel.demo <- IsoSwitch(MarvelObject=marvel.demo,
                         method="ad",
                         psi.pval=0.1,
                         psi.delta=0,
                         gene.pval=0.1,
                         gene.log2fc=0.5
                         )

# Check outputs
head(marvel.demo$DE$Cor$Table_Raw)
head(marvel.demo$DE$Cor$Table)
marvel.demo$DE$Cor$Plot
marvel.demo$DE$Cor$Plot.Stats
```

IsoSwitch.10x *Classify gene-splicing relationship*

Description

Classify gene-splicing relative changes to each other from cell group 1 to group 2. Classifications are coordinated, opposing, isoform-switching, and complex. In coordinated relationship, both gene and splicing changes in the same direction from cell group 1 to group 2. In opposing relationship, gene changes in the opposite direction relative to splicing from cell group 1 to group 2. In isoform-switching, there is differential splice junction usage without differential expression of the corresponding gene between cell group 1 and group 2. Complex relationship involves genes with both coordinated and opposing relationships with splicing. Only differentially spliced junctions are included for analysis here.

Usage

```
IsoSwitch.10x(
  MarvelObject,
  pval.sj = 0.05,
  log2fc.sj = NULL,
  delta.sj = 5,
  min.gene.norm = 0,
  pval.adj.gene = 0.05,
  log2fc.gene = 0.5
)
```

Arguments

MarvelObject	Marvel object. S3 object generated from CompareValues.Genes.10x function.
pval.sj	Numeric value. p-value from differential splicing analysis, below which, the splice junction is considered differentially spliced. Default is 0.05.
log2fc.sj	Numeric value. Absolute log2 fold change from differential splicing analysis, above which, the splice junction is considered differentially spliced. This option should be NULL if delta.sj has been specified.
delta.sj	Numeric value. Absolute difference in average PSI values between the two cell groups, above which, the splice junction is considered differentially spliced. This option should be NULL if log2fc.sj has been specified.
min.gene.norm	Numeric value. The average normalised gene expression across the two cell groups above which the splice junction is considered differentially spliced. Default is 0.
pval.adj.gene	Numeric value. Adjusted p-value from differential gene expression analysis, below which, the gene is considered differentially expressed. Default is 0.05.
log2fc.gene	Numeric value. Absolute log2 fold change from differential gene expression analysis, above which, the gene is considered differentially expressed. This option should be NULL if delta.sj has been specified.

Value

An object of class S3 containing new slots MarvelObject$SJ.Gene.Cor$Data, MarvelObject$SJ.Gene.Cor$Proportion$
and MarvelObject$SJ.Gene.Cor$Proportion$Table.

Examples

```
marvel.demo.10x <- readRDS(system.file("extdata/data",
                                "marvel.demo.10x.rds",
                                package="MARVEL")
                                )

marvel.demo.10x <- readRDS(system.file("extdata/data",
                                "marvel.demo.10x.rds",
                                package="MARVEL")
                                )
```

```
marvel.demo.10x <- IsoSwitch.10x(
                        MarvelObject=marvel.demo.10x,
                        pval.sj=0.05,
                        delta.sj=5,
                        min.gene.norm=1.0,
                        pval.adj.gene=0.05,
                        log2fc.gene=0.5
                        )

# Check outputs
marvel.demo.10x$SJ.Gene.Cor$Proportion$Plot
marvel.demo.10x$SJ.Gene.Cor$Proportion$Table
cols <- c("coord.intron", "gene_short_name", "cor.complete")
head(marvel.demo.10x$SJ.Gene.Cor$Data[,cols])
```

`IsoSwitch.PlotExpr` *Plot gene-splicing relative change*

Description

Plots delta PSI vs gene log2-fold change

Usage

```
IsoSwitch.PlotExpr(MarvelObject, anno = FALSE)
```

Arguments

`MarvelObject`	Marvel object. S3 object generated from `CompareValues.Genes.10x` function.
`anno`	Logical value. If set to `TRUE`, genes with coordinated, opposing or complex change relative to splicing change will be annotated on the plot. Default value is `FALSE`.

Value

An object of class S3 containing with new slots `MarvelObject$DE$Cor$PSIvsExpr$Plot`.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

marvel.demo <- IsoSwitch.PlotExpr(MarvelObject=marvel.demo, anno=TRUE)

# Check output
marvel.demo$DE$Cor$PSIvsExpr$Plot
```

ModalityChange *Classify modality changes*

Description

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

Usage

```
ModalityChange(MarvelObject, method, psi.pval, psi.delta = 0)
```

Arguments

MarvelObject	Marvel object. S3 object generated from CompareValues function.
method	Character string. The statistical method used for differential splicing analysis.
psi.pval	Numeric value. Adjusted p-value below which the splicing event is considered differentially spliced and included for modality analysis.
psi.delta	Numeric value. The absolute difference between the means PSI values of cell group 1 and 2, above which, the splicing event is considered differentially spliced and included for modality analysis.

Value

An object of class S3 with new slots MarvelObjectDEModality$Table, MarvelObject$DE$Modality$Plot, and MarvelObjectDEModality$Plot.Stats.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

marvel.demo <- ModalityChange(MarvelObject=marvel.demo,
                              method="ad",
                              psi.pval=0.1,
                              psi.delta=0
                              )

# Check outputs
head(marvel.demo$DE$Modality$Table)
marvel.demo$DE$Modality$Plot
marvel.demo$DE$Modality$Plot.Stats
```

ParseGTF *Parse gene transfer file (GTF)*

Description

Parses the gene transfer file (GTF) for downstream nonsense-mediated decay (NMD) prediction.

Usage

```
ParseGTF(MarvelObject)
```

Arguments

MarvelObject	Marvel object. S3 object generated from CompareValues.PSI function.

Details

This function parses the GTF in order to generate new columns for gene IDs, transcript IDs, and transcript type. These information are extracted from the attribute (9th) column for a standard GTF. These information will be used for downstream NMD prediction.

Value

An object of class S3 with new slot MarvelObjectNMDGTF.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

marvel.demo <- ParseGTF(MarvelObject=marvel.demo)
```

PctASE *Tabulate differentially spliced splicing event*

Description

Tabulates the percentage or absoluate number of significant splicing events for each splicing type.

Usage

```
PctASE(
  MarvelObject,
  method,
  psi.pval,
  psi.mean.diff,
  ylabels.size = 8,
  barlabels.size = 3,
  x.offset = 0,
  direction.color = NULL,
  mode = "percentage"
)
```

Arguments

MarvelObject	Marvel object. S3 object generated from CompareValues function.
method	Character string. The statistical method used for differential splicing analysis.
psi.pval	Numeric value. Adjusted p-value below which the splicing event is considered differentially spliced and included for tabulation.
psi.mean.diff	Numeric value. The minimum absolute differences in PSI values between the two cell groups above which the splicing event is considered differentially spliced and included for tabulation.
ylabels.size	Numeric value. Size of the xtick labels. Default is 8.
barlabels.size	Numeric value. Size of the labels above each bar. Default is 3
x.offset	Numeric value. The values on the x-axis to offset by. Useful when right margin overshadow the numbers above the bars. Default value is 0.
direction.color	Character strings. Vector of length 2 to specify the colors for significanly down- and up-regulated splicing events. Default is NULL, which corresponds to default ggplot2 color scheme.
mode	Character strings. When set to "percentage" (default), percentage of significant splicing events over total splicing events detected will be tabulate. When set to absolute, the number of significant splicing events will be tabulated.

Value

An object of class S3 with new slots MarvelObjectDEAbsASE$Table and MarvelObject$DE$AbsASE$Plot.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

marvel.demo <- PctASE(MarvelObject=marvel.demo,
                      method="ad",
                      psi.pval=0.1,
                      psi.mean.diff=0
                      )
```

PlotDEValues *Plot differential splicing and gene expression analysis results*

Description

Volcano plot of differential splicing and gene expression analysis results. This is a wrapper function for PlotDEValues.PSI.Mean, PlotDEValues.Exp.Global, and PlotDEValues.Exp.Spliced.

Usage

```
PlotDEValues(
  MarvelObject,
  method = NULL,
  pval,
  level,
  delta = NULL,
  log2fc = NULL,
  psi.pval = NULL,
  psi.delta = NULL,
  gene.pval = NULL,
  gene.log2fc = NULL,
  point.size = 1,
  xlabel.size = 8,
  point.alpha = 1,
  anno = FALSE,
  anno.gene_short_name = NULL,
  anno.tran_id = NULL,
  label.size = 2.5,
  y.upper.offset = 5,
  event.types = c("SE", "MXE", "RI", "A5SS", "A3SS", "AFE", "ALE"),
  event.types.colors = NULL
)
```

Arguments

MarvelObject	Marvel object. S3 object generated from CompareValues function.
method	Character string. The statistical method used for differential splicing analysis.
pval	Numeric value. Only applicable when level set to "splicing.mean", "splicing.distance", and "gene.global". Adjusted p-value below which the splcing events or genes are considered as statistically significant and will consequently be color-annotated on the plot.
level	Character string. Indicate "splicing.distance" if the percent spliced-in (PSI) values' distribution was previously tested between 2 groups of cells using the CompareValues function. Statistical tests for distribution include Kolmogorov-Smirnov, Kuiper, and Anderson-Darling test. Indicate "splicing.mean" or gene if the PSI or gene expression values' mean was previously tested between

2 groups of cells using the CompareValues function. Statistical tests for comparing mean are t-test and Wilcoxon rank-sum test.

delta — Numeric value. Only applicable when level set to "splicing.mean". The positive (and negative) value specified above (and below) which the splicing events are considered to be statistically significant and will consequently be color-annotated on the plot.

log2fc — Numeric value. Only applicable when level set to "gene.global". The positive (and negative) value specified above (and below) which the genes are considered to be statistically significant and will consequently be color-annotated on the plot.

psi.pval — Numeric value. Only applicable when level set to "gene.spliced". The adjusted p-value from differential splicing analysis, below which, the splicing event is considered differentially spliced. Default is 0.1.

psi.delta — Numeric value. Only applicable when level set to "gene.spliced". The absolute differences in average PSI value between two cell groups from differential splicing analysis, above which, the splicing event is considered differentially spliced. Default is 0.

gene.pval — Numeric value. Only applicable when level set to "gene.spliced". The adjusted p-value from differential gene expression analysis, below which, the gene is considered differentially expressed. Default is 0.1.

gene.log2fc — Numeric value. Only applicable when level set to "gene.spliced". The absolute log2 fold change in gene expression betwene two cell groups from differential splicing analysis, above which, the gene is considered differentially expressed. Default is 0.5.

point.size — Numeric value. Size of data points. Default is 1.

xlabel.size — Numeric value. Font size of the xtick labels. Default is 8.

point.alpha — Numeric value. Only applicable when level set to "splicing.mean.g2vsg1". Transpancy of data points. Default is 1.

anno — Logical value. If set to TRUE, the specific gene names or splicing events will be annotated on the plot.

anno.gene_short_name — Vector of character strings. When anno set to TRUE, the gene names to be annotated on the plot.

anno.tran_id — Vector of character strings. When anno set to TRUE, the coordinates of the splicing events to be annotated on the plot.

label.size — Numeric value. Only applicable if anno set to TRUE. Size of the gene name labels.

y.upper.offset — Numeric value. The value in -log10(p-value) to increase the upper limit of the y-axis. To be used when anno set to TRUE so that gene labels will not be truncated at the upper limit of the y-axis.

event.types — Vector of character string(s). Only applicable when level set to "splicing.mean.g2vsg1". The specific splicing event to plot. May take any one or more of the following values "SE", "MXE", "RI", "A5SS", "A3SS", "AFE", and "ALE".

event.types.colors
: Vector of character string(s). Only applicable when level set to "splicing.mean.g2vsg1". Customise colors as per splicing event type specified in event.types option. Should be of same length as event.types option.

Value

An object of class S3 with new slot MarvelObjectDEPSI$Plot[["method"]] when level set to "splicing.mean" or "splicing.distance" or MarvelObjectDEExp.Global$Table and MarvelObject$DE$Exp.Global$Plot when level set to "gene.global" or MarvelObjectDEExp.Spliced$Table and MarvelObject$DE$Exp.Spliced$Plot when level set to "gene.spliced".

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

marvel.demo <- PlotDEValues(MarvelObject=marvel.demo,
                            method="ad",
                            pval=0.10,
                            level="splicing.distance"
                            )

# Check output
marvel.demo$DE$PSI$Plot[["ad"]]
```

PlotDEValues.Exp.Global

Plot global differential gene expression analysis results

Description

Volcano plot of differential splicing analysis results based on all expressed genes between 2 groups of cells. x-axis represents the log2 fold change in gene expression. y-axis represents the adjusted p-values.

Usage

```
PlotDEValues.Exp.Global(
  MarvelObject,
  pval = 0.1,
  log2fc = 0.5,
  point.size = 1,
  anno = FALSE,
  anno.gene_short_name = NULL,
  label.size = 2.5,
  y.upper.offset = 5,
  xlabel.size = 8
)
```

Arguments

MarvelObject	Marvel object. S3 object generated from CompareValues function.
pval	Numeric value. Adjusted p-value below which the genes are considered as statistically significant and will consequently be color-annotated on the plot.
log2fc	Numeric value. The positive (and negative) value specified above (and below) which the genes are considered to be statistically significant and will consequently be color-annotated on the plot.
point.size	Numeric value. The point size for the data points. Default value is 1.
anno	Logical value. If set to TRUE, the specific gene names will be annotated on the plot as defined in anno.gene_short_name option.
anno.gene_short_name	Vector of character strings. When anno set to TRUE, the gene names to be annotated on the plot.
label.size	Numeric value. Only applicable if anno set to TRUE. Size of the gene name labels.
y.upper.offset	Numeric value. The value in -log10(p-value) to increase the upper limit of the y-axis. To be used when anno set to TRUE so that gene labels will not be truncated at the upper limit of the y-axis.
xlabel.size	Numeric value. Font size of the xtick labels. Default is 8.

Value

An object of class S3 with new slots MarvelObjectDEExp.Global$Table, MarvelObject$DE$Exp.Global$Summary, and MarvelObjectDEExp.Global$Plot

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

marvel.demo <- PlotDEValues.Exp.Global(MarvelObject=marvel.demo,
                                       pval=0.10,
                                       log2fc=0.5
                                       )

# Check output
head(marvel.demo$DE$Exp.Global$Table)
marvel.demo$DE$Exp.Global$Plot
marvel.demo$DE$Exp.Global$Summary
```

PlotDEValues.Exp.Spliced

Plot differential gene expression analysis of differentially spliced genes

---

Description

Volcano plot of differential splicing analysis results based on differentially spliced genes between 2 groups of cells. x-axis represents the log2 fold change in gene expression. y-axis represents the adjusted p-values.

Usage

```
PlotDEValues.Exp.Spliced(
  MarvelObject,
  method,
  psi.pval = 0.1,
  psi.delta = 0,
  gene.pval = 0.1,
  gene.log2fc = 0.5,
  point.size = 1,
  anno = FALSE,
  anno.gene_short_name = NULL,
  label.size = 2.5,
  y.upper.offset = 5,
  xlabel.size = 8
)
```

Arguments

MarvelObject	S3 object generated from CompareValues function.
method	(Vector of) Character string(s). The method specified in CompareValues function when level option set to "splicing".
psi.pval	Numeric value. The adjusted p-value from differential splicing analysis, below which, the splicing event is considered differentially spliced. Default is 0.1.
psi.delta	Numeric value. The absolute differences in average PSI value between two cell groups from differential splicing analysis, above which, the splicing event is considered differentially spliced. Default is 0.
gene.pval	Numeric value. The adjusted p-value from differential gene expression analysis, below which, the gene is considered differentially expressed. Default is 0.1.
gene.log2fc	Numeric value. The absolute log2 fold change in gene expression betwene two cell groups from differential splicing analysis, above which, the gene is considered differentially expressed. Default is 0.5.
point.size	Numeric value. Size of data points. Default is 1.
anno	Logical value. If set to TRUE, the specific gene names will be annotated on the plot as defined in anno.gene_short_name option.
anno.gene_short_name	Vector of character strings. When anno set to TRUE, the gene names to be annotated on the plot.
label.size	Numeric value. Only applicable if anno set to TRUE. Size of the gene name labels.

y.upper.offset	Numeric value. The value in -log10(p-value) to increase the upper limit of the y-axis. To be used when anno set to TRUE so that gene labels will not be truncated at the upper limit of the y-axis.
xlabel.size	Numeric value. Font size of the xtick labels. Default is 8.

Value

An object of class S3 with new slots MarvelObjectDEExp.Spliced$Table, MarvelObject$DE$Exp.Spliced$Summary, and MarvelObjectDEExp.Spliced$Plot.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

marvel.demo <- PlotDEValues.Exp.Spliced(MarvelObject=marvel.demo,
                                        method="ad",
                                        psi.pval=0.1,
                                        psi.delta=0,
                                        gene.pval=0.1,
                                        gene.log2fc=0.5
                                        )
# Check output
marvel.demo$DE$Exp.Spliced$Summary
marvel.demo$DE$Exp.Spliced$Plot
```

PlotDEValues.Genes.10x

Plot differential gene analysis results

Description

Volcano plot of results from differential gene expression analysis. x-axis represents the log2 fold change between two cell groups. y-axis represents -log10(adjusted p-value). Only genes whose splice junctions were considered to be differentially spliced are included for plotting.

Usage

```
PlotDEValues.Genes.10x(
  MarvelObject,
  pval.sj = 0.05,
  log2fc.sj = NULL,
  delta.sj = 5,
  min.gene.norm = 0,
  pval.adj.gene = 0.05,
  log2fc.gene = 0.5,
  anno = FALSE,
  anno.gene_short_name = NULL,
  label.size = 2
)
```

Arguments

MarvelObject	Marvel object. S3 object generated from CompareValues.Genes.10x function.
pval.sj	Numeric value. p-value from differential splicing analysis, below which, the splice junction is considered differentially spliced. Default is 0.05.
log2fc.sj	Numeric value. Absolute log2 fold change from differential splicing analysis, above which, the splice junction is considered differentially spliced. This option should be NULL if delta.sj has been specified.
delta.sj	Numeric value. Absolute difference in average PSI values between the two cell groups, above which, the splice junction is considered differentially spliced. This option should be NULL if log2fc.sj has been specified.
min.gene.norm	Numeric value. The average normalised gene expression across the two cell groups above which the splice junction is considered differentially spliced. Default is 0.
pval.adj.gene	Numeric value. Adjusted p-value from differential gene expression analysis, below which, the gene is considered differentially expressed. Default is 0.05.
log2fc.gene	Numeric value. Absolute log2 fold change from differential gene expression analysis, above which, the gene is considered differentially expressed. This option should be NULL if delta.sj has been specified.
anno	Logical value. If set to TRUE, user-specific genes in anno.gene_short_name will be annotated on the plot. Default is FALSE.
anno.gene_short_name	Vector of character strings. If anno set to TRUE, genes specified here will be annotated on the plot.
label.size	Numeric value. If anno set to TRUE, the font size of the annotations on the plot will be adjusted to the size specified here. Default is 2.

Value

An object of class S3 with a new slots MarvelObjectDESJ$VolcanoPlot$Gene$Plot and MarvelObject$DESJVolcano

Examples

```
marvel.demo.10x <- readRDS(system.file("extdata/data",
                                "marvel.demo.10x.rds",
                                package="MARVEL")
                                )

marvel.demo.10x <- PlotDEValues.Genes.10x(
                        MarvelObject=marvel.demo.10x,
                        pval.sj=0.05,
                        delta.sj=5,
                        min.gene.norm=1.0,
                        pval.adj.gene=0.05,
                        log2fc.gene=0.5
                        )
```

```
# Check outputs
marvel.demo.10x$DE$SJ$VolcanoPlot$Gene$Plot
head(marvel.demo.10x$DE$SJ$VolcanoPlot$Gene$Data)
```

`PlotDEValues.PSI.Distance` *Plot differential splicing analysis results based on distance statistics.*

Description

Ranked plot for differential splicing analysis results based on distance statistics. Only statistical test that assess the overall PSI distribution between two cell groups will be eligible for plotting here, e.g., Anderson-Darling and DTS. x-axis represents the distance statistics. y-axis represents the adjusted p-values.

Usage

```
PlotDEValues.PSI.Distance(
  MarvelObject,
  method,
  pval,
  point.size = 1,
  xlabel.size = 8,
  anno = FALSE,
  anno.tran_id = NULL,
  label.size = 2.5,
  y.upper.offset = 5
)
```

Arguments

`MarvelObject`	Marvel object. S3 object generated from `CompareValues` function.
`method`	Character string. The statistical method used for differential splicing analysis.
`pval`	Numeric value. Adjusted p-value below which the splcing events are considered as statistically significant and will consequently be color-annotated on the plot.
`point.size`	Numeric value. The point size for the data points. Default value is `1`.
`xlabel.size`	Numeric value. Font size of the xtick labels. Default is `8`.
`anno`	Logical value. If set to `TRUE`, the specific gene names will be annotated on the plot. Speficified together with `anno.tran_id`.
`anno.tran_id`	Vector of character strings. When `anno` set to `TRUE`, the coordinates of the splicing events to be annotated on the plot.
`label.size`	Numeric value. Only applicable if `anno` set to TRUE. Size of the gene name labels.
`y.upper.offset`	Numeric value. The value in -log10(p-value) to increase the upper limit of the y-axis. To be used when `anno` set to TRUE so that gene labels will not be truncated at the upper limit of the y-axis.

Value

An object of class S3 containing with new slot `MarvelObject$DE$PSI$Plot[["method"]]`.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

marvel.demo <- PlotDEValues.PSI.Distance(MarvelObject=marvel.demo,
                                         method="ad",
                                         pval=0.10
                                         )

# Check output
marvel.demo$DE$PSI$Plot[["ad"]]
```

PlotDEValues.PSI.Mean *Plot differential splicing analysis results based on mean PSI difference*

Description

Volcano plot of differential splicing analysis results based on mean PSI difference between 2 groups of cells. x-axis represents the mean delta PSI. y-axis represents the adjusted p-values.

Usage

```
PlotDEValues.PSI.Mean(
  MarvelObject,
  method,
  pval = 0.1,
  delta = 5,
  point.size = 1,
  xlabel.size = 8,
  anno = FALSE,
  anno.tran_id = NULL,
  label.size = 2.5,
  y.upper.offset = 5
)
```

Arguments

`MarvelObject`	Marvel object. S3 object generated from `CompareValues` function.
`method`	Character string. The statistical method used for differential splicing analysis.
`pval`	Numeric value. Adjusted p-value below which the splcing event are considered as statistically significant and will consequently be color-annotated on the plot.
`delta`	Numeric value. The positive (and negative) value specified above (and below) which the splicing events are considered to be statistically significant and will consequently be color-annotated on the plot.

point.size	Numeric value. The point size for the data points. Default value is 1.
xlabel.size	Numeric value. Font size of the xtick labels. Default is 8.
anno	Logical value. If set to TRUE, the specific gene names will be annotated on the plot. Speficified together with anno.tran_id.
anno.tran_id	Vector of character strings. When anno set to TRUE, the coordinates of the splicing events to be annotated on the plot.
label.size	Numeric value. Only applicable if anno set to TRUE. Size of the gene name labels.
y.upper.offset	Numeric value. The value in -log10(p-value) to increase the upper limit of the y-axis. To be used when anno set to TRUE so that gene labels will not be truncated at the upper limit of the y-axis.

Value

An object of class S3 containing with new slot MarvelObjectDEPSI$Plot[["method"]].

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

marvel.demo <- PlotDEValues.PSI.Mean(MarvelObject=marvel.demo,
                                     method="ad",
                                     pval=0.10,
                                     delta=5
                                     )

# Check output
marvel.demo$DE$PSI$Plot[["ad"]]
```

PlotDEValues.PSI.Mean.g2vsg1

Plot differential splicing analysis results based on mean PSI difference

---

Description

Scatterplot of differential splicing analysis results based on mean PSI difference between 2 groups of cells. x-axis represents the mean PSI values of cell group 1. y-axis represents the mean PSI values of cell group 2.

Usage

```
PlotDEValues.PSI.Mean.g2vsg1(
  MarvelObject,
  method,
  pval,
  delta = 5,
  point.size = 1,
```

```
  xlabel.size = 8,
  anno = FALSE,
  anno.tran_id = NULL,
  label.size = 2.5,
  point.alpha = 1,
  event.types = c("SE", "MXE", "RI", "A5SS", "A3SS", "AFE", "ALE"),
  event.types.colors = NULL
)
```

Arguments

MarvelObject	Marvel object. S3 object generated from CompareValues function.
method	Character string. The statistical method used for differential splicing analysis.
pval	Numeric value. Adjusted p-value below which the splcing event are considered as statistically significant and will consequently be color-annotated on the plot.
delta	Numeric value. The positive (and negative) value specified above (and below) which the splicing events are considered to be statistically significant and will consequently be color-annotated on the plot.
point.size	Numeric value. The point size for the data points. Default value is 1.
xlabel.size	Numeric value. Font size of the xtick labels. Default is 8.
anno	Logical value. If set to TRUE, the specific gene names will be annotated on the plot. Speficified together with anno.tran_id.
anno.tran_id	Vector of character strings. When anno set to TRUE, the coordinates of the splicing events to be annotated on the plot.
label.size	Numeric value. Only applicable if anno set to TRUE. Size of the gene name labels.
point.alpha	Numeric value. Transpancy of data points. Default is 1.
event.types	Vector of character string(s). The specific splicing event to plot. May take any one or more of the following values "SE", "MXE", "RI", "A5SS", "A3SS", "AFE", and "ALE".
event.types.colors	Vector of character string(s). Customise colors as per splicing event type specified in event.types option. Should be of same length as event.types option.

Value

An object of class S3 containing with new slot MarvelObjectDEPSI$Plot[["method"]].

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

marvel.demo <- PlotDEValues.PSI.Mean.g2vsg1(MarvelObject=marvel.demo,
                                    method="ad",
                                    pval=0.10,
                                    delta=5
                                    )
```

```
# Check output
marvel.demo$DE$PSI$Plot
marvel.demo$DE$PSI$Summary
```

`PlotDEValues.SJ.10x` *Plot differential splice junction analysis results*

Description

Volcano plot of results from differential splice junction analysis. x-axis represents the average normalised gene expression across the two cell groups. y-axis represents the differences or log2 fold change between the two cell groups.

Usage

```
PlotDEValues.SJ.10x(
  MarvelObject,
  pval = 0.05,
  log2fc = NULL,
  delta = 5,
  min.gene.norm = 0,
  anno = FALSE,
  anno.coord.intron = NULL,
  label.size = 2
)
```

Arguments

`MarvelObject`	Marvel object. S3 object generated from `CompareValues.Genes.10x` function.
`pval`	Numeric value. p-value, below which, the splice junction is considered differentially spliced. To be used in conjunction with `log2fc`, `delta`, and `min.gene.norm`. Default is `0.05`.
`log2fc`	Numeric value. Absolute log2 fold change, above which, the splice junction is considered differentially spliced. This option should be `NULL` if `delta` has been specified.
`delta`	Numeric value. Absolute differences in average PSI values between the two cell groups, above which, the splice junction is considered differentially spliced. This option should be `NULL` if `log2fc` has been specified.
`min.gene.norm`	Numeric value. The average normalised gene expression across the two cell groups above which the splice junction is considered differentially spliced. Default is `0`.
`anno`	Logical value. If set to `TRUE`, user-specific spliced genes in `anno.coord.intron` will be annotated on the plot. Default is `FALSE`.

anno.coord.intron
: Vector of character strings. If anno set to TRUE, splice junction coordinates specified here will be annotated on the plot.

label.size
: Numeric value. If anno set to TRUE, the font size of the annotations on the plot will be adjusted to the size specified here. Default is 2.

Value

An object of class S3 with a new slots MarvelObjectDESJ$VolcanoPlot$SJ$Plot and MarvelObject$DESJVolcanoPl

Examples

```
marvel.demo.10x <- readRDS(system.file("extdata/data",
                                "marvel.demo.10x.rds",
                                package="MARVEL")
                                )

marvel.demo.10x <- PlotDEValues.SJ.10x(
                        MarvelObject=marvel.demo.10x,
                        pval=0.05,
                        delta=5,
                        min.gene.norm=1.0,
                        anno=FALSE
                        )

# Check outputs
marvel.demo.10x$DE$SJ$VolcanoPlot$SJ$Plot
head(marvel.demo.10x$DE$SJ$VolcanoPlot$SJ$Data)
```

PlotPctExprCells.Genes.10x

Plot gene expression distribution

Description

Generates a plot of gene expression distribution (percentage of cells expressing a particular gene) to determine normalised gene expression threshold for downstream differential splice junction analysis.

Usage

```
PlotPctExprCells.Genes.10x(
  MarvelObject,
  cell.group.g1,
  cell.group.g2,
  min.pct.cells = 1
)
```

Arguments

MarvelObject	Marvel object. S3 object generated from CheckAlignment.10x function.
cell.group.g1	Vector of character strings. Cell IDs corresponding to Group 1 (reference group) of downstream differential splice junction analysis.
cell.group.g2	Vector of character strings. Cell IDs corresponding to Group 2 of downstream differential splice junction analysis.
min.pct.cells	Numeric value. Minimum percentage of cells in which the gene is expressed for that gene to be included for gene expression distribution analysis. Expressed genes defined as genes with non-zero normalised UMI counts.

Value

An object of class S3 with a new slots MarvelObject$pct.cells.expr$Gene$Plot and MarvelObject$pct.cells.expr$G

Examples

```
marvel.demo.10x <- readRDS(system.file("extdata/data",
                                "marvel.demo.10x.rds",
                                package="MARVEL")
                                )

# Define cell groups
    # Retrieve sample metadata
    sample.metadata <- marvel.demo.10x$sample.metadata

    # Group 1 (reference)
    index <- which(sample.metadata$cell.type=="iPSC")
    cell.ids.1 <- sample.metadata[index, "cell.id"]
    length(cell.ids.1)

    # Group 2
    index <- which(sample.metadata$cell.type=="Cardio day 10")
    cell.ids.2 <- sample.metadata[index, "cell.id"]
    length(cell.ids.2)

# Explore % of cells expressing genes
marvel.demo.10x <- PlotPctExprCells.Genes.10x(
                        MarvelObject=marvel.demo.10x,
                        cell.group.g1=cell.ids.1,
                        cell.group.g2=cell.ids.2,
                        min.pct.cells=5
                        )

# Check output
marvel.demo.10x $pct.cells.expr$Gene$Plot
head(marvel.demo.10x $pct.cells.expr$Gene$Data)
```

`PlotPctExprCells.SJ.10x` *Plot splice junction expression distribution*

Description

Generates a plot of splice junction expression distribution (percentage of cells expressing a particular splice junction) to determine splice junction expression threshold for downstream differential splice junction analysis.

Usage

```
PlotPctExprCells.SJ.10x(
  MarvelObject,
  cell.group.g1,
  cell.group.g2,
  min.pct.cells.genes = 10,
  min.pct.cells.sj = 10,
  downsample = FALSE,
  downsample.pct.sj = 10,
  seed = 1
)
```

Arguments

`MarvelObject`	Marvel object. S3 object generated from `CheckAlignment.10x` function.
`cell.group.g1`	Vector of character strings. Cell IDs corresponding to Group 1 (reference group) of downstream differential splice junction analysis.
`cell.group.g2`	Vector of character strings. Cell IDs corresponding to Group 2 of downstream differential splice junction analysis.
`min.pct.cells.genes`	Numeric value. Minimum percentage of cells in which the gene is expressed for that gene to be included for splice junction expression distribution analysis. Expressed genes defined as genes with non-zero normalised UMI counts. This threshold may be determined from `PlotPctExprCells.SJ.10x` function.
`min.pct.cells.sj`	Numeric value. Minimum percentage of cells in which the splice junction is expressed for that splice junction to be included for splice junction expression distribution analysis. Expressed splice junctions defined as splice junctions with raw UMI counts >= 1.
`downsample`	Logical value. If set to `TRUE`, the splice junctions will be downsampled so that only a smaller number of splice junctions will be included for expression exploration analysis here. Default value is `FALSE`.
`downsample.pct.sj`	Numeric value. If `downsample` set to `TRUE`, the minimum percentage of splice junctions to include for expression exploration analysis here.

seed — Numeric value. To ensure the splice junctions downsampled will always be reproducible.

Value

An object of class S3 with a new slots `MarvelObject$pct.cells.expr$SJ$Plot` and `MarvelObject$pct.cells.expr$SJ$`

Examples

```
marvel.demo.10x <- readRDS(system.file("extdata/data",
                                "marvel.demo.10x.rds",
                                package="MARVEL")
                                )

# Define cell groups
    # Retrieve sample metadata
    sample.metadata <- marvel.demo.10x$sample.metadata

    # Group 1 (reference)
    index <- which(sample.metadata$cell.type=="iPSC")
    cell.ids.1 <- sample.metadata[index, "cell.id"]
    length(cell.ids.1)

    # Group 2
    index <- which(sample.metadata$cell.type=="Cardio day 10")
    cell.ids.2 <- sample.metadata[index, "cell.id"]
    length(cell.ids.2)

# Explore % of cells expressing SJ
marvel.demo.10x <- PlotPctExprCells.SJ.10x(
                    MarvelObject=marvel.demo.10x,
                    cell.group.g1=cell.ids.1,
                    cell.group.g2=cell.ids.2,
                    min.pct.cells.genes=5,
                    min.pct.cells.sj=5,
                    downsample=TRUE,
                    downsample.pct.sj=100
                    )

marvel.demo.10x$pct.cells.expr$SJ$Plot
head(marvel.demo.10x$pct.cells.expr$SJ$Data)
```

`PlotValues` *Plot percent spliced-in (PSI) or gene expression values*

Description

Plots percent spliced-in (PSI) or gene expression values across different groups of cells. This is a wrapper function for `PlotValues.Exp` and `PlotValues.PSI`.

Usage

```
PlotValues(
  MarvelObject,
  cell.group.list,
  feature,
  maintitle = "gene_short_name",
  xlabels.size = 8,
  level,
  min.cells = NULL,
  sigma.sq = 0.001,
  bimodal.adjust = NULL,
  seed = NULL,
  modality.column = "modality.bimodal.adj",
  scale.y.log = FALSE,
  max.cells.jitter = 10000,
  max.cells.jitter.seed = 1,
  cell.group.colors = NULL,
  point.alpha = 0.2
)
```

Arguments

MarvelObject	Marvel object. S3 object generated from TransformExpValues function.
cell.group.list	List of character strings. Each element of the list is a vector of cell IDs corresponding to a cell group. The name of the element will be the cell group label.
feature	Character string. tran_id or gene_id for plotting. Should match tran_id or gene_id column of MarvelObject$ValidatedSpliceFeature or MarvelObject$GeneFeature slot when level set to "splicing" or "gene", respectively.
maintitle	Character string. Column to use as plot main title as per MarvelObject$ValidatedSpliceFeature or MarvelObject$GeneFeature when level set to "splicing" or "gene", respectively. Default is "gene_short_name" column.
xlabels.size	Numeric value. Size of x-axis labels as per ggplot2 function. Default is 8.
level	Character string. Indicate "splicing" or "gene" for PSI or gene expression value plotting, respectively.
min.cells	Numeric value. Only applicable when level set to "splicing". The minimum no. of cells expressing the splicing event to be included for analysis.
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. Default is 0.001.
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 paramters 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. Default is "modality.bimodal.adj".

scale.y.log — Logical value. Only applicable when level set to "splicing". If set to TRUE, the y-axis of will log10-scaled. Useful when most PSI values are extremely small (< 0.02) or big (> 0.98). Default is FALSE.

max.cells.jitter — Numeric value. Only applicable when level set to "splicing". Maximum number of cells for jitter points. Cells are randomly downsampled to show on jitter plot. Useful when there are large number of cells so that individual jitter points do not overcrowd the violin plot. Specified together with max.cells.jitter.seed. To disable this option, specify a value large than the number of cells in each cell group.

max.cells.jitter.seed — Numeric value. Only applicable when level set to "splicing". Cells downsampled are reproducible. Specified together with max.cells.jitter.

cell.group.colors — Character string. Vector of colors for the cell groups specified for PCA analysis using cell.type.columns, cell.type.variable, and cell.type.labels. If not specified, default ggplot2 colors will be used.

point.alpha — Numeric value. Transparency of the data points. Takes any values between 0-1. Default value is 0.2.

Value

An object of class S3 with new slot $adhocPlot$PSI or MarvelObject$adhocPlot$Exp when level set to "splicing" or "gene", respectively.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

# Define cell groups to plot
df.pheno <- marvel.demo$SplicePheno
cell.group.g1 <- df.pheno[which(df.pheno$cell.type=="iPSC"), "sample.id"]
cell.group.g2 <- df.pheno[which(df.pheno$cell.type=="Endoderm"), "sample.id"]
cell.group.list <- list(cell.group.g1, cell.group.g2)
names(cell.group.list) <- c("iPSC", "Endoderm")

# Plot
marvel.demo <- PlotValues(MarvelObject=marvel.demo,
                          cell.group.list=cell.group.list,
                        feature="chr17:8383254:8382781|8383157:-@chr17:8382143:8382315",
                          level="splicing",
```

```
                          min.cells=5,
                          xlabels.size=5
                          )

# Check output
marvel.demo$adhocPlot$PSI
```

PlotValues.Exp *Plot gene expression values*

Description

Boxplot of gene expression values across different groups of cells.

Usage

```
PlotValues.Exp(
  MarvelObject,
  cell.group.list,
  feature,
  maintitle = "gene_short_name",
  xlabels.size = 8,
  cell.group.colors = NULL,
  point.alpha = 0.2
)
```

Arguments

MarvelObject	Marvel object. S3 object generated from TransformExpValues function.
cell.group.list	List of character strings. Each element of the list is a vector of cell IDs corresponding to a cell group. The name of the element will be the cell group label.
feature	Character string. gene_id for plotting. Should match gene_id column of MarvelObject$GeneFeature slot.
maintitle	Character string. Column to use as plot main title as per MarvelObject$GeneFeature. Default is "gene_short_name" column.
xlabels.size	Numeric value. Size of x-axis labels as per ggplot2 function. Default is 8.
cell.group.colors	Character string. Vector of colors for the cell groups specified for PCA analysis using cell.type.columns, cell.type.variable, and cell.type.labels. If not specified, default ggplot2 colors will be used.
point.alpha	Numeric value. Transparency of the data points. Takes any values between 0-1. Default value is 0.2.

Value

An object of class S3 with new slot MarvelObject$adhocPlot$Exp.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

df.pheno <- marvel.demo$SplicePheno
cell.group.g1 <- df.pheno[which(df.pheno$cell.type=="iPSC"), "sample.id"]
cell.group.g2 <- df.pheno[which(df.pheno$cell.type=="Endoderm"), "sample.id"]
cell.group.list <- list(cell.group.g1, cell.group.g2)
names(cell.group.list) <- c("iPSC", "Endoderm")

# Plot
marvel.demo <- PlotValues.Exp(MarvelObject=marvel.demo,
                              cell.group.list=cell.group.list,
                              feature="ENSG00000161970.15",
                              xlabels.size=8
                              )

# Check output
marvel.demo$adhocPlot$Exp
```

PlotValues.PCA.CellGroup.10x

Annotate reduced dimension space with cell feature

Description

Annotates reduced dimension space, e.g., UMAP and tSNE, with cell features such as cell group, donor ID, sample ID, etc.

Usage

```
PlotValues.PCA.CellGroup.10x(
  MarvelObject,
  cell.group.list,
  legendtitle = "Cell group",
  alpha = 0.75,
  point.size = 1,
  point.stroke = 0.1,
  point.colors = NULL,
  point.size.legend = 2,
  type
)
```

Arguments

`MarvelObject`	Marvel object. S3 object generated from `CheckAlignment.10x` function.
`cell.group.list`	List of character strings. Each element of the list is a vector of cell IDs corresponding to a feature, e.g. cell group. The names of the element will be the cell feature label.

legendtitle	Character string. Legend title. Default is "Cell group".
alpha	Numeric value. Transparency of the data points. Takes any values between 0-1 whereby 0 is totally transparent and 1 is opaque. Default is 0.75.
point.size	Numeric value. Size of data points. Default is 1.
point.stroke	Numeric value. Outline thickness of data points. Default is 0.1.
point.colors	Vector of character strings. Colors of cell groups and should be same length as cell.group.list. Default ggplot2 colors are used.
point.size.legend	Numeric value. Size of legend keys. Default is 2.
type	Character string. Type of reduced dimension space. Options are "umap" and "tsne".

Value

An object of class S3 with new slot MarvelObject$adhocPlot$PCA$CellGroup.

Examples

```
marvel.demo.10x <- readRDS(system.file("extdata/data",
                              "marvel.demo.10x.rds",
                              package="MARVEL")
                              )

# Define cell groups
    # Retrieve sample metadata
    sample.metadata <- marvel.demo.10x$sample.metadata

    # iPSC
    index <- which(sample.metadata$cell.type=="iPSC")
    cell.ids.1 <- sample.metadata[index, "cell.id"]
    length(cell.ids.1)

    # Cardio day 10
    index <- which(sample.metadata$cell.type=="Cardio day 10")
    cell.ids.2 <- sample.metadata[index, "cell.id"]
    length(cell.ids.2)

    # Save into list
    cell.group.list <- list("iPSC"=cell.ids.1,
                            "Cardio d10"=cell.ids.2
                            )

# Plot cell groups
marvel.demo.10x <- PlotValues.PCA.CellGroup.10x(
                            MarvelObject=marvel.demo.10x,
                            cell.group.list=cell.group.list,
                            legendtitle="Cell group",
                            type="tsne"
                            )
```

```
# Check output
marvel.demo.10x$adhocPlot$PCA$CellGroup
```

`PlotValues.PCA.Gene.10x` *Annotate reduced dimension space with gene expression values*

Description

Annotates reduced dimension space, e.g., UMAP and tSNE, with gene expression values. Values will be automatically be log2-transformed prior to plotting.

Usage

```
PlotValues.PCA.Gene.10x(
  MarvelObject,
  cell.ids = NULL,
  gene_short_name,
  log2.transform = TRUE,
  point.size = 0.1,
  color.gradient = c("grey90", "blue", "red"),
  type
)
```

Arguments

`MarvelObject`	Marvel object. S3 object generated from `CheckAlignment.10x` function.
`cell.ids`	Vector of character strings. Specify specific cells to plot.
`gene_short_name`	Character string. Gene name whose expression will be plotting.
`log2.transform`	Logical value. If set to `TRUE` (default), normalised gene expression values will be off-set by 1 and then log2-transformed prior to analysis.
`point.size`	Numeric value. Size of data points. Default is `1`.
`color.gradient`	Vector of character strings. Colors to indicate low, moderate, and high expression. Default is `c("grey90","blue","red")`.
`type`	Character string. Type of reduced dimension space. Options are `"umap"` and `"tsne"`.

Value

An object of class S3 with new slot `MarvelObject$adhocPlot$PCA$Gene`.

Examples

```
marvel.demo.10x <- readRDS(system.file("extdata/data",
                                "marvel.demo.10x.rds",
                                package="MARVEL")
                                )

# Define cell groups
    # Retrieve sample metadata
    sample.metadata <- marvel.demo.10x$sample.metadata

    # iPSC
    index <- which(sample.metadata$cell.type=="iPSC")
    cell.ids.1 <- sample.metadata[index, "cell.id"]
    length(cell.ids.1)

    # Cardio day 10
    index <- which(sample.metadata$cell.type=="Cardio day 10")
    cell.ids.2 <- sample.metadata[index, "cell.id"]
    length(cell.ids.2)

    # Save into list
    cell.group.list <- list("iPSC"=cell.ids.1,
                            "Cardio d10"=cell.ids.2
                            )

# Plot expression
marvel.demo.10x <- PlotValues.PCA.Gene.10x(
                      MarvelObject=marvel.demo.10x,
                      gene_short_name="TPM2",
                      color.gradient=c("grey","cyan","green","yellow","red"),
                      type="tsne"
                      )

# Check output
marvel.demo.10x$adhocPlot$PCA$Gene
```

PlotValues.PCA.PSI.10x

Annotate reduced dimension space with PSI values

Description

Annotates reduced dimension space, e.g., UMAP and tSNE, with PSI values.

Usage

```
PlotValues.PCA.PSI.10x(
  MarvelObject,
```

```
  cell.ids = NULL,
  coord.intron,
  min.gene.count = 3,
  point.size = 0.1,
  log2.transform = FALSE,
  color.gradient = c("grey90", "blue", "red"),
  type
)
```

Arguments

MarvelObject	Marvel object. S3 object generated from CheckAlignment.10x function.
cell.ids	Vector of character strings. Specific set of cells to plot.
coord.intron	Character string. Coordinates of splice junction whose expression will be plotted.
min.gene.count	Numeric value. Minimum raw gene count, above which, the PSI value will be calculate for the cell. Default is 3.
point.size	Numeric value. Size of data points. Default is 1.
log2.transform	Logical value. If set to TRUE, PSI values will be log2-transformed. Useful for highlighting small changes in PSI values between cell groups. Default is FALSE.
color.gradient	Vector of character strings. Colors to indicate low, moderate, and high expression. Default is c("grey90","blue","red").
type	Character string. Type of reduced dimension space. Options are "umap" and "tsne".

Value

An object of class S3 with new slot MarvelObject$adhocPlot$PCA$PSI.

Examples

```
marvel.demo.10x <- readRDS(system.file("extdata/data",
                                "marvel.demo.10x.rds",
                                package="MARVEL")
                                )

# Define cell groups
    # Retrieve sample metadata
    sample.metadata <- marvel.demo.10x$sample.metadata

    # iPSC
    index <- which(sample.metadata$cell.type=="iPSC")
    cell.ids.1 <- sample.metadata[index, "cell.id"]
    length(cell.ids.1)

    # Cardio day 10
    index <- which(sample.metadata$cell.type=="Cardio day 10")
    cell.ids.2 <- sample.metadata[index, "cell.id"]
```

```
    length(cell.ids.2)

    # Save into list
    cell.group.list <- list("iPSC"=cell.ids.1,
                            "Cardio d10"=cell.ids.2
                            )

# Plot expression
marvel.demo.10x <- PlotValues.PCA.PSI.10x(
                        MarvelObject=marvel.demo.10x,
                        coord.intron="chr1:23693914:23694659",
                        min.gene.count=3,
                        log2.transform=FALSE,
                        color.gradient=c("grey","cyan","green","yellow","red"),
                        type="tsne"
                        )

# Check output
marvel.demo.10x$adhocPlot$PCA$PSI
```

PlotValues.PSI *Plot percent spliced-in (PSI) values*

Description

Violin plot of percent spliced-in (PSI) values across different groups of cells.

Usage

```
PlotValues.PSI(
  MarvelObject,
  cell.group.list,
  feature,
  maintitle = "gene_short_name",
  xlabels.size = 8,
  max.cells.jitter = 10000,
  max.cells.jitter.seed = 1,
  min.cells = 25,
  sigma.sq = 0.001,
  bimodal.adjust = TRUE,
  seed = 1,
  modality.column = "modality.bimodal.adj",
  scale.y.log = FALSE,
  cell.group.colors = NULL,
  point.alpha = 0.2
)
```

Arguments

MarvelObject	Marvel object. S3 object generated from TransformExpValues function.
cell.group.list	List of character strings. Each element of the list is a vector of cell IDs corresponding to a cell group. The name of the element will be the cell group label.
feature	Character string. Coordinates of splicing event to plot.
maintitle	Character string. Column to use as plot main title as per MarvelObject$ValidatedSpliceFeature. Default is "gene_short_name" column.
xlabels.size	Numeric value. Size of x-axis labels as per ggplot2 function. Default is 8.
max.cells.jitter	Numeric value. Maximum number of cells for jitter points. Cells are randomly downsampled to show on jitter plot. Useful when there are large number of cells so that individual jitter points do not overcrowd the violin plot.
max.cells.jitter.seed	Numeric value. Cells downsampled are reproducible.
min.cells	Numeric value. The minimum no. of cells expressing the splicing event to be included for analysis. Please refer to AssignModality function help page for more details.
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. Default is 0.001.
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 paramters 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.adj". Please refer to AssignModality function help page for more details. Default is "modality.bimodal.adj".
scale.y.log	Logical value. Only applicable when level set to "splicing". If set to TRUE, the y-axis of will log10-scaled. Useful when most PSI values are extremely small (< 0.02) or big (> 0.98). Default is FALSE.
cell.group.colors	Character string. Vector of colors for the cell groups specified for PCA analysis using cell.type.columns, cell.type.variable, and cell.type.labels. If not specified, default ggplot2 colors will be used.
point.alpha	Numeric value. Transparency of the data points. Takes any values between 0-1. Default value is 0.2.

Value

An object of class S3 with new slot MarvelObject$adhocPlot$PSI.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

# Define cell groups to plot
df.pheno <- marvel.demo$SplicePheno
cell.group.g1 <- df.pheno[which(df.pheno$cell.type=="iPSC"), "sample.id"]
cell.group.g2 <- df.pheno[which(df.pheno$cell.type=="Endoderm"), "sample.id"]
cell.group.list <- list(cell.group.g1, cell.group.g2)
names(cell.group.list) <- c("iPSC", "Endoderm")

# Plot
marvel.demo <- PlotValues.PSI(MarvelObject=marvel.demo,
                              cell.group.list=cell.group.list,
                        feature="chr17:8383254:8382781|8383157:-@chr17:8382143:8382315",
                              min.cells=5,
                              xlabels.size=5
                              )

# Check output
marvel.demo$adhocPlot$PSI
```

`PropModality` *Tabulate modality proportion*

Description

Tabulates and plots the proportion of each modality. This is a wrapper function for `PropModality.Doughnut` and `PropModality.Bar` functions.

Usage

```
PropModality(
  MarvelObject,
  modality.column,
  modality.type,
  event.type,
  across.event.type,
  prop.test = NULL,
  prop.adj = NULL,
  xlabels.size = 8,
  zoom = FALSE,
  yinterval = NULL
)
```

Arguments

`MarvelObject`	Marvel object. 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. `complete` 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.

`xlabels.size`
: Numeric value. Only applicable when `across.event.type` set to `TRUE`. Size of x-axis labels as per `ggplot2` function. Default is 8.

`zoom`
: Logical value. Only applicable if `across.event.type` set to `TRUE`. If set to `TRUE`, users can specify the range of the y-axis using `yinterval` argument. Useful when scrutinasing low-frequency event types, e.g. middle and multimodal.

`yinterval`
: Logical value. Only applicable if `across.event.type` set to `TRUE` and `zoom` set to `TRUE`.

Value

An object of class S3 containing with new slot `$Modality$Prop$DoughnutChart` or `$Modality$Prop$BarChart`.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

marvel.demo <- PropModality(MarvelObject=marvel.demo,
                            modality.column="modality.bimodal.adj",
                            modality.type="extended",
                           event.type=c("SE", "MXE", "RI", "A5SS", "A3SS", "AFE", "ALE"),
                            across.event.type=FALSE
                            )

# Check outputs
marvel.demo$Modality$Prop$DoughnutChart$Table
marvel.demo$Modality$Prop$DoughnutChart$Plot
```

PropModality.Bar *Modality proportion broken down by event type*

Description

Tabulates and plots the proportion of each modality broken down by splicing event type.

Usage

```
PropModality.Bar(
  MarvelObject,
  modality.column,
  modality.type,
  event.type,
  xlabels.size = 8,
  zoom = FALSE,
  yinterval = NULL,
  prop.test,
  prop.adj
)
```

Arguments

MarvelObject	Marvel object. 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.
xlabels.size	Numeric value. Size of x-axis labels as per ggplot2 function. Default is 8.
zoom	Logical value. If set to TRUE, users can specify the range of the y-axis using yinterval argument. Useful when scrutinasing low-frequency event types, e.g. middle and multimodal.
yinterval	Logical value. Only applicable when zoom is set to TRUE.
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.

Value

An object of class S3 containing new slots `MarvelObject$Modality$Prop$BarChart$Table` and `MarvelObject$Modality$Prop$BarChart$Stats`.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

marvel.demo <- PropModality.Bar(MarvelObject=marvel.demo,
                                modality.column="modality.bimodal.adj",
                                modality.type="extended",
                           event.type=c("SE", "MXE", "RI", "A5SS", "A3SS", "AFE", "ALE"),
                                prop.test="fisher",
                                prop.adj="fdr"
                                )

# Check outputs
head(marvel.demo$Modality$Prop$BarChart$Table)
marvel.demo$Modality$Prop$BarChart$Plot
marvel.demo$Modality$Prop$BarChart$Stats
```

PropModality.Doughnut *Overall modality proportion*

Description

Tabulates and plots the proportion of each modality without breaking down by splicing event type.

Usage

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

Arguments

`MarvelObject`	Marvel object. 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. `complete` 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.

Value

An object of class S3 with new slots `MarvelObject$Modality$Prop$DoughnutChart$Table` and `MarvelObject$Modality$Prop$DoughnutChart$Plot`.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

marvel.demo <- PropModality.Doughnut(MarvelObject=marvel.demo,
                                     modality.column="modality.bimodal.adj",
                                     modality.type="extended",
                                event.type=c("SE", "MXE", "RI", "A5SS", "A3SS", "AFE", "ALE")
                                     )

# Check outputs
marvel.demo$Modality$Prop$DoughnutChart$Table
marvel.demo$Modality$Prop$DoughnutChart$Plot
```

PropPTC *Tabulate proportion of transcripts with PTC*

Description

Tabulates and plots the proportion of transcripts with PTC for each splicing event type.

Usage

```
PropPTC(MarvelObject, xlabels.size = 8, show.NovelSJ.NoCDS = TRUE, prop.test)
```

Arguments

MarvelObject	Marvel object. S3 object generated from FindPTC function.
xlabels.size	Numeric value. Size of the x-axis tick labels. Default is 8.
show.NovelSJ.NoCDS	Logical value. If set to TRUE transcripts not analysed for premature terminal codon (PTC), e.g. non-protein-coding transcripts are tabulated and plotted.
prop.test	Character string. chisq Chi-squared test used to compare the proportion of transcripts with PTC across the different event splicing type. fisher Fisher test used to compare the proportion of transcripts with PTC across the different splicing event type.

Value

An object of class S3 with new slots MarvelObjectNMDPTC.Prop$Table, MarvelObject$NMD$PTC.Prop$Plot, and MarvelObjectNMDPTC.Prop$Plot.Stats.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

marvel.demo <- PropPTC(MarvelObject=marvel.demo,
                       xlabels.size=8,
                       show.NovelSJ.NoCDS=TRUE,
```

```
                      prop.test="fisher"
                      )

# Check outputs
head(marvel.demo$NMD$PTC.Prop$Table)
marvel.demo$NMD$PTC.Prop$Plot
marvel.demo$NMD$PTC.Prop$Plot.Stats
```

RunPCA *Principle component analysis*

Description

Performs principle component analysis on splicing or gene data. This is a wrapper function for `RunPCA.PSI` and `RunPCA.Exp`.

Usage

```
RunPCA(
  MarvelObject,
  cell.group.column,
  cell.group.order = NULL,
  cell.group.colors = NULL,
  sample.ids = NULL,
  min.cells = 25,
  features,
  point.size = 0.5,
  point.alpha = 0.75,
  point.stroke = 0.1,
  seed = 1,
  method.impute = "random",
  cell.group.column.impute = NULL,
  level
)
```

Arguments

`MarvelObject` Marvel object. S3 object generated from `TransformExpValues` function.

`cell.group.column`
: Character string. The name of the sample metadata column in which the variables will be used to label the cell groups on the PCA.

`cell.group.order`
: Character string. The order of the variables under the sample metadata column specified in `cell.group.column` to appear in the PCA cell group legend.

`cell.group.colors`
: Character string. Vector of colors for the cell groups specified for PCA analysis using `cell.type.columns` and `cell.group.order`. If not specified, default `ggplot2` colors will be used.

sample.ids	Character strings. Specific cells to plot.
min.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_id or gene_id for analysis. Should match tran_id or gene_id column of MarvelObject$ValidatedSpliceFeature or MarvelObject$GeneFeature when level set to "splicing" or "gene", respectively.
point.size	Numeric value. Size of data points on reduced dimension space.
point.alpha	Numeric value. Transparency of the data points on reduced dimension space. Take any values between 0 to 1. The smaller the value, the more transparent the data points will be.
point.stroke	Numeric value. The thickness of the outline of the data points. The larger the value, the thicker the outline of the data points.
seed	Numeric value. Only applicable when level set to "splicing". Ensures imputed values for NA PSIs are reproducible.
method.impute	Character string. Only applicable when level set to "splicing". Indicate the method for imputing missing PSI values (low coverage). "random" method randomly assigns any values between 0-1. "population.mean" method uses the mean PSI value for each cell population. Default option is "population.mean".
cell.group.column.impute	Character string. Only applicable when method.impute set to "population.mean". The name of the sample metadata column in which the variables will be used to impute missing values.
level	Character string. Indicate "splicing" or "gene" for splicing or gene expression analysis, respectively

Value

An object of class S3 with new slots MarvelObjectPCAPSI$Results, MarvelObject$PCAPSIPlot, and MarvelObjectPCAPSI$Plot.Elbow or MarvelObjectPCAExp$Results, MarvelObject$PCAExpPlot, and MarvelObjectPCAExp$Plot.Elbow, when level option specified as "splicing" or "gene", respectively.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

# Define splicing events for analysis
df <- do.call(rbind.data.frame, marvel.demo$PSI)
tran_ids <- df$tran_id

# PCA
marvel.demo <- RunPCA(MarvelObject=marvel.demo,
                      sample.ids=marvel.demo$SplicePheno$sample.id,
                      cell.group.column="cell.type",
                      cell.group.order=c("iPSC", "Endoderm"),
                      cell.group.colors=NULL,
```

```
                    min.cells=5,
                    features=tran_ids,
                    level="splicing",
                    point.size=2
                    )

# Check outputs
head(marvel.demo$PCA$PSI$Results$ind$coord)
marvel.demo$PCA$PSI$Plot
```

RunPCA.Exp *Principle component analysis for gene Data*

Description

Performs principle component analysis using gene expression values.

Usage

```
RunPCA.Exp(
  MarvelObject,
  sample.ids = NULL,
  cell.group.column,
  cell.group.order = NULL,
  cell.group.colors = NULL,
  features,
  min.cells = 25,
  point.size = 0.5,
  point.alpha = 0.75,
  point.stroke = 0.1
)
```

Arguments

MarvelObject	Marvel object. S3 object generated from TransformExpValues function.
sample.ids	Character strings. Specific cells to plot.
cell.group.column	Character string. The name of the sample metadata column in which the variables will be used to label the cell groups on the PCA.
cell.group.order	Character string. The order of the variables under the sample metadata column specified in cell.group.column to appear in the PCA cell group legend.
cell.group.colors	Character string. Vector of colors for the cell groups specified for PCA analysis using cell.type.columns and cell.group.order. If not specified, default ggplot2 colors will be used.

features	Character string. Vector of gene_id for analysis. Should match gene_id column of MarvelObject$GeneFeature.
min.cells	Numeric value. The minimum no. of cells expressing the gene to be included for analysis.
point.size	Numeric value. Size of data points on reduced dimension space.
point.alpha	Numeric value. Transparency of the data points on reduced dimension space. Take any values between 0 to 1. The smaller the value, the more transparent the data points will be.
point.stroke	Numeric value. The thickness of the outline of the data points. The larger the value, the thicker the outline of the data points.

Value

An object of class S3 containing with new slots MarvelObjectPCAExp$Results, MarvelObject$PCAExpPlot, and MarvelObjectPCAExp$Plot.Elbow.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

# Define genes for analysis
gene_ids <- marvel.demo$Exp$gene_id

# PCA
marvel.demo <- RunPCA.Exp(MarvelObject=marvel.demo,
                          sample.ids=marvel.demo$SplicePheno$sample.id,
                          cell.group.column="cell.type",
                          cell.group.order=c("iPSC", "Endoderm"),
                          min.cells=5,
                          features=gene_ids,
                          point.size=2
                          )

# Check outputs
head(marvel.demo$PCA$Exp$Results$ind$coord)
marvel.demo$PCA$Exp$Plot
```

RunPCA.PSI *Principle component analysis for splicing data*

Description

Performs principle component analysis using PSI values.

Usage

```
RunPCA.PSI(
  MarvelObject,
  sample.ids = NULL,
  cell.group.column,
  cell.group.order,
  cell.group.colors = NULL,
  features,
  min.cells = 25,
  point.size = 0.5,
  point.alpha = 0.75,
  point.stroke = 0.1,
  seed = 1,
  method.impute = "random",
  cell.group.column.impute = NULL
)
```

Arguments

MarvelObject	Marvel object. S3 object generated from TransformExpValues function.
sample.ids	Character strings. Specific cells to plot.
cell.group.column	Character string. The name of the sample metadata column in which the variables will be used to label the cell groups on the PCA.
cell.group.order	Character string. The order of the variables under the sample metadata column specified in cell.group.column to appear in the PCA cell group legend.
cell.group.colors	Character string. Vector of colors for the cell groups specified for PCA analysis using cell.type.columns and cell.group.order. If not specified, default ggplot2 colors will be used.
features	Character string. Vector of tran_id for analysis. Should match tran_id column of MarvelObject$ValidatedSpliceFeature.
min.cells	Numeric value. The minimum no. of cells expressing the splicing event to be included for analysis.
point.size	Numeric value. Size of data points on reduced dimension space.
point.alpha	Numeric value. Transparency of the data points on reduced dimension space. Take any values between 0 to 1. The smaller the value, the more transparent the data points will be.
point.stroke	Numeric value. The thickness of the outline of the data points. The larger the value, the thicker the outline of the data points.
seed	Numeric value. Ensures imputed values for NA PSIs are reproducible.
method.impute	Character string. Indicate the method for imputing missing PSI values (low coverage). "random" method randomly assigns any values between 0-1. "population.mean" method uses the mean PSI value for each cell population. Default option is "population.mean".

`cell.group.column.impute`
: Character string. Only applicable when `method.impute` set to `"population.mean"`. The name of the sample metadata column in which the variables will be used to impute missing values.

Value

An object of class S3 containing with new slots `MarvelObject$PCA$PSI$Results` and `MarvelObject$PCA$PSI$Plot`

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

# Define splicing events for analysis
df <- do.call(rbind.data.frame, marvel.demo$PSI)
tran_ids <- df$tran_id

# PCA
marvel.demo <- RunPCA.PSI(MarvelObject=marvel.demo,
                          sample.ids=marvel.demo$SplicePheno$sample.id,
                          cell.group.column="cell.type",
                          cell.group.order=c("iPSC", "Endoderm"),
                          cell.group.colors=NULL,
                          min.cells=5,
                          features=tran_ids,
                          point.size=2
                          )

# Check outputs
head(marvel.demo$PCA$PSI$Results$ind$coord)
marvel.demo$PCA$PSI$Plot
```

SubsetCrypticA3SS *Differential gene expression analysis for differentially spliced genes*

Description

Performs differential gene expression analysis between 2 groups of cells only on differentially spliced genes.

Usage

```
SubsetCrypticA3SS(MarvelObject, method, distance.to.ss = c(1, 100))
```

Arguments

`MarvelObject`	Marvel object. S3 object generated from `TransformExpValues` function.
`method`	Vector of character string(s). To include splicing events from these method(s) for differential splicing analysis.
`distance.to.ss`	Character string. Range of distances between A3SS and canonical splice site to consider A3SS to be cryptic. Default value `c(1, 100)`.

Value

An object of class S3 updated slot `MarvelObject$DE$PSI$Table` and new slot `MarvelObject$DE$PSI$A3SS.dist.to.ss`.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

marvel.demo <- SubsetCrypticA3SS(MarvelObject=marvel.demo,
                                 method="ad",
                                 distance.to.ss=c(1,100)
                                 )

# Check output
head(marvel.demo$DE$PSI$Table[["ad"]])
```

`SubsetSamples` *Subset samples (cells)*

Description

Subsets specific samples (cells) from sample metadata.

Usage

```
SubsetSamples(MarvelObject, sample.ids)
```

Arguments

`MarvelObject`	Marvel object. S3 object generated from `CreateMarvelObject` function.
`sample.ids`	Vector of character strings. Sample IDs to subset.

Value

An object of class S3 with updated slot `MarvelObject$SplicePheno`.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

sample.ids <- sample(marvel.demo$SplicePheno$sample.id, size=10)

marvel.demo <- SubsetSamples(MarvelObject=marvel.demo,
                             sample.ids=sample.ids
                             )
```

`TransformExpValues` *Transform gene expression Values*

Description

Transforms gene expression values and censor lowly-expressing genes.

Usage

```
TransformExpValues(
  MarvelObject,
  offset = 1,
  transformation = "log2",
  threshold.lower = 1
)
```

Arguments

`MarvelObject`	Marvel object. S3 object generated from `CheckAlignment` function.
`offset`	Numeric value. To indicate the value to add to the expression values before log transformation. The only option for this argument is 1.
`transformation`	Character string. To indicate the type of transformation to use on the expression values after offsetting the values. The only option for this argument is `log2`.
`threshold.lower`	Numeric value. To indicate the value below which the expression values will be censored, i.e. re-coded as 0, after offsetting and transforming the values. The only option for this argument is 1.

Value

An object of class S3 with updated slot `MarvelObject$Exp`.

Examples

```
marvel.demo <- readRDS(system.file("extdata/data", "marvel.demo.rds", package="MARVEL"))

marvel.demo <- TransformExpValues(MarvelObject=marvel.demo,
                                  offset=1,
                                  transformation="log2",
                                  threshold.lower=1
                                  )
```

ValidateSJ.10x *Validate splice junctions*

Description

Retains splice junctions whose start and end belong to the same gene.

Usage

```
ValidateSJ.10x(MarvelObject, keep.novel.sj = FALSE)
```

Arguments

MarvelObject	Marvel object. S3 object generated from AnnotateSJ.10x function.
keep.novel.sj	Logical value. If set to TRUE, novel splice junctions will be retained for downstream analysis. Novel splice junctions are defined as splice junctions with one end reported in GTF while the other was not reported in GTF. Default value is FALSE.

Value

An object of class S3 containing the updated slots MarvelObject$sj.metadata and MarvelObject$sj.count.matrix.

Examples

```
# Load un-processed MARVEL object
marvel.demo.10x.raw <- readRDS(system.file("extdata/data",
                               "marvel.demo.10x.raw.rds",
                               package="MARVEL")
                               )

# Annotate gene metadata
marvel.demo.10x <- AnnotateGenes.10x(MarvelObject=marvel.demo.10x.raw)

# Annotate junction metadata
marvel.demo.10x <- AnnotateSJ.10x(MarvelObject=marvel.demo.10x)

# Validate junctions
marvel.demo.10x <- ValidateSJ.10x(MarvelObject=marvel.demo.10x)
```

Index