ASpedia-R

October 30, 2023

`asr_converter` *Generate ASpedia-R input format from DAS analysis tools result.*

Description

result of DAS analysis tools convert to ASpedia-R input format.

Usage

```
asr_converter(
  das.analysis.result = "",
  program = "",
  pvalue.cutoff = 0.05,
  dpsi.cutoff = 0.1,
  gene.model = "Ensembl",
  genome.version = "GRCh38",
  as.type = "",
  gtf.file.name = "",
  ioe.file.name = ""
)
```

Arguments

`das.analysis.result`	name of DAS analysis tools result file.
`program`	name of DAS analysis tool. one of rMATS, SUPPA, or spliceR
`pvalue.cutoff`	value of pvalue cutoff. default value is 0.05
`dpsi.cutoff`	value of dPSI cutoff. default value is 0.1
`gene.model`	gene model of reference. One of Refseq, Ensembl, or GENCODE. (spliceR only)
`genome.version`	genome version of reference. One of hg18, GRCh19, or GRCh38. (spliceR only)
`as.type`	AS event type. One of A3SS, A5SS, SE, MXE, or RI. (rMATS only)
`gtf.file.name`	a GTF format file of reference. (SUPPA only)
`ioe.file.name`	name of ioe file generated by SUPPA generateEvents command. (SUPPA)

Value

converter.result

Examples

```
## rMATS
das.file.name <- system.file("extdata" "rMATS_test.txt", package="ASpediaR")
rmats.converter.result <- asr_converter(das.file.name, program="rMATS", as.type="SE")

##SUPPA
das.file.name <- system.file("extdata", "SUPPA_test.txt", package="ASpediaR")
gtf.file.name <- system.file("extdata", "test_gtf.gtf", package="ASpediaR")
ioe.file.name <- system.file("extdata", "SUPPA_test.ioe", package="ASpediaR")
suppa.converter.result <- asr_converter(das.file.name, program="SUPPA",
                                        gtf.file=gtf.file.name, ioe.file=ioe.file.name)

##spliceR
das.file.name <- system.file("extdata", "spliceR_test.txt", package="ASpediaR")
splilcer.converter.result <- asr_converter(das.file.name, program="spliceR",
                                           gene.model="Ensembl", genome.version="GRCh38")
```

`rMATS.converter` *Generate ASpedia-R input format from rMATS result.*

Description

rMSTS result convert to ASpedia-R input format.

Usage

```
rMATS.converter(rMATS.result, as.type, pvalue.cutoff, dpsi.cutoff)
```

Arguments

`rMATS.result`	name of rMATS result file.
`as.type`	AS event type. One of A3SS, A5SS, SE, MXE, or RI.
`pvalue.cutoff`	value of pvalue cutoff. default value is 0.05
`dpsi.cutoff`	value of dPSI cutoff. default value is 0.1

Value

converting.result

Examples

```
rmats.result.file.name <- system.file("extdata", "rMATS_test.txt", package="ASpediaR")
rmats.converter.result <- rMATS_converter(rmats.result.file.name, program="rMATS",
                                          as.type="SE")
```

SUPPA.converter *Generate ASpedia-R input format from SUPPA result.*

Description

SUPPA result convert to ASpedia-R input format.

Usage

```
SUPPA.converter(
  SUPPA.result,
  pvalue.cutoff,
  dpsi.cutoff,
  gtf.file.name,
  ioe.file.name
)
```

Arguments

SUPPA.result	name of SUPPA result file.
pvalue.cutoff	value of pvalue cutoff. default value is 0.05
dpsi.cutoff	value of dPSI cutoff. default value is 0.1
gtf.file.name	a GTF format file of reference.
ioe.file.name	name of ioe file generated by SUPPA generateEvents command.

Value

converting.result

Examples

```
suppa.result.file <- system.file("extdata", "SUPPA_test.txt", package="ASpediaR")
gtf.file.name <- system.file("extdata", "test_gtf.gtf", package="ASpediaR")
suppa.ioe.file.name <- system.file("extdata", "SUPPA_test.ioe", package="ASpediaR")
suppa.converter.result <- SUPPA_converter(suppa.result.file, gtf.file=gtf.file.name,
                                          ioe.file=suppa.ioe.file.name)
```

spliceR.converter *Generate ASpedia-R input format from spliceR result.*

Description

spliceR result convert to ASpedia-R input format.

Usage

```
spliceR.converter(spliceR.result, pvalue.cutoff, gene.model, genome.version)
```

Arguments

spliceR.result	name of spliceR result file.
pvalue.cutoff	value of pvalue cutoff. default value is 0.05
gene.model	gene model of reference. One of Refseq, Ensembl, or GENCODE.
genome.version	genome version of reference. One of hg18, GRCh19, or GRCh38.

Value

converting.result

Examples

```
spliceR.result.file <- system.file("extdata", "spliceR_test.txt", package="ASpediaR")
splilcer.converter.result <- spliceR_converter(spliceR.result.file, program="spliceR",
                                        gene.model="Ensembl", genome.version="GRCh38")
```

---

asr_annotation *converting result mapping to ASDB*

Description

DNA and RNA annotation related to aliternative splicing from ASpedia DB(ASDB) add to asr_converter (rMATS_converter, SUPPA_converter, or spliceR_converter) result. And gene enrichment test result are provided between annotation result gene list and knowledge-based database gene list

Usage

```
asr_annotation(
  converter.result,
  gene.model = "Ensembl",
  genome.version = "GRCh38",
  gsea.gene.list = "",
  result.dir = ""
)
```

Arguments

converter.result	asr_converter(rMATS_converter, SUPPA_converter, or spliceR_converter) result
gene.model	gene model of reference. One of Refseq, Ensembl, or GENCODE.
genome.version	genome version of reference. One of hg18, GRCh19, or GRCh38.
gsea.gene.list	optional. reference gene list for gene enrichment test with annotation result gene list and knowledge-based database gene list. If gene list is empty, use all genes in reference.
result.dir	directory where annotation result(.tsv file) is saved

Value

annotation result

Examples

```
annotation.result.dir <- system.file("extdata/annotation_result", package="ASpediaR")
annotation.result <- asr_annotation(rmats.converter.result, "Ensembl", "GRCh38",
                                    result.dir=annotation.result.dir)
```

`mining_gsea` *gene enrichment test between annotation gene list and knowledge-based database gene list.*

Description

Gene enrichment test result are provided to tsv and plot format between annotation gene list and knowledge-based database gene list.

Usage

```
mining_gsea(annotation.gene.list, gsea.gene.list, result.dir)
```

Arguments

`annotation.gene.list`	gene list from asr_annotation function result
`gsea.gene.list`	gene list from reference
`result.dir`	directory where GSEA result(.tsv and .png file) are saved

Examples

```
## reference gene list from GTF
library(rtracklayer)
gtf.file.name <- system.file("extdata", "test_gtf.gtf", package="ASpediaR")
gtf.data <- import(gtf.file.name)
gtf.gene.list <- unique(gtf.data$gene_name)
annotation.gene.list <- unique(annotation.result$gene_symbol)
gsea.result.dir <- system.file("extdata/gsea_result", package="ASpediaR")
mining_gsea(annotation.gene.list, gsea.gene.list=reference.gene.list,
              result.dir=gsea.result.dir)

## reference gene list from user input
test.gene.list.file.name <- system.file("extdata", "test_whole_gene.txt", package="ASpediaR")
test.gene.list <- read.table(test.gene.list.file.name, header=FALSE, stringsAsFactors=FALSE)
gsea.result.dir <- system.file("extdata/gsea_result", package="ASpediaR")
mining_gsea(annotation.gene.list, gsea.gene.list=test.gene.list, result.dir=gsea.result.dir)
```

asr_plot *visualization of annotation result*

Description

DNA and RNA features in result of asr_annotation function to be visualize.

Usage

```
asr_plot(
  annotation.result,
  gtf.file.name,
  gene.model = "Ensembl",
  genome.version = "hg38",
  gene.name = "",
  as.id = "",
  heights.list = "",
  plot.data.list = "",
  result.dir = ""
)
```

Arguments

annotation.result	asr_annotation function result.
gtf.file.name	a GTF format file of reference.
gene.model	gene model of reference. One of Refseq, Ensembl, or GENCODE.
genome.version	genome version of reference. One of hg18, GRCh19, or GRCh38.
gene.name	gene name to be visualization.
as.id	list of AS ID to be visualization.
heights.list	positive integer vectors for track heights include height of ideogram and gene track.
plot.data.list	list of DNA or RNA feature to be visualization. choose from “conservation”, “NMD”, “repeats”, “domain”, “PTM”, or “RBP”.
result.dir	directory where plots(.png files) are saved

Examples

```
gtf.file.name <- system.file(“extdata”, “test_gtf.gtf”, package=“ASpediaR”)
plot.result.dir <- system.file(“extdata/plot_result”, package=“ASpediaR”)

##using gene name
asr_plot(annotation.result, gtf.file.name, gene.model="Ensembl", genome.version="GRCh38",
          gene.name="FGFR2", result.dir=plot.result.dir)

##using AS ID
asr_plot(annotation.result, gtf.file.name, gene.model="Ensembl", genome.version="GRCh38",
          as.id="chr10:121520169:121519979:121518829:121518682:121517463:121517319",
          result.dir=plot.result.dir)
```

```
##track lists and track heights are change
asr_plot(annotation.result, gtf.file.name, gene.model="Ensembl", genome.version="GRCh38",
          gene.name="FGFR2", heights.list=c(1, 2, 2, 1, 1, 1, 2),
          list.of.plot=c("conservation", "domain", "PTM", "repeats", "RBP"),
          result.dir=plot.result.dir)
```

`asr_plot_ppi` *visualization of PPI network*

Description

If asr_annotation function result has protein protein interaction(PPI) information, we provide PPI network plot.

Usage

```
asr_plot_ppi(annotation.result, gene.name = "", as.id = "", result.dir = "")
```

Arguments

`annotation.result`	asr_annotation function result.
`gene.name`	gene name to be visualization.
`as.id`	list of AS ID to be visualization.
`result.dir`	directory where PPI plots(.png files) are saved

Examples

```
ppi.result.dir <- system.file("extdata/ppi_result", package="ASpediaR")

##using gene name
asr_plot_ppi(annotation.result, gene.name="FGFR2", result.dir=ppi.result.dir)

##using AS ID
asr_plot_ppi(annotation.result,
            as.id="chr10:121520169:121519979:121518829:121518682:121517463:121517319",
            result.dir=ppi.result.dir)
```

Index