# Package 'RCPA'

January 20, 2025

**Type** Package

**Title** Consensus Pathway Analysis

**Version** 0.2.5

**Description** Provides a set of functions to perform pathway analysis and meta-analysis from multiple gene expression datasets, as well as visualization of the results. This package wraps functionality from the following packages: Ritchie et al. (2015) <doi:10.1093/nar/gkv007>, Love et al. (2014) <doi:10.1186/s13059-014-0550-8>, Robinson et al. (2010) <doi:10.1093/bioinformatics/btp616>, Korotkevich et al. (2016) <arxiv:10.1101/060012>, Efron et al. (2015) <https://CRAN.R-project.org/package=GSA>, and Gu et al. (2012) <https://CRAN.R-project.org/package=CePa>.

**License** GPL-3

**Encoding** UTF-8

**Depends** R (>= 4.0), tidyr, dplyr, ggplot2, utils, stats

**biocViews** Biobase, DESeq2, GEOquery, edgeR, limma, RCyjs, fgsea, BrowserViz, SummarizedExperiment, AnnotationDbi, ROntoTools

**Imports** AnnotationDbi, SummarizedExperiment, BiocManager, Biobase, DESeq2, GEOquery, edgeR, limma, stringr(>= 1.5.0), ggnewscale, ggrepel, graph, httr, rlang, ggpattern, scales, RobustRankAggreg, methods, jsonlite, IRdisplay

**RoxygenNote** 7.3.2

**Suggests** knitr, rmarkdown, markdown, testthat (>= 3.0.0), ROntoTools, fgsea, GSA, CePa, meta, png, ggvenn, RCurl, XML, S4Vectors, hgu133plus2.db, org.Hs.eg.db, pd.hg.u133.plus.2, oligo

**Config/testthat/edition** 3

**Config/testthat/stop-on-test-error** false

**VignetteBuilder** knitr, rmarkdown

**BugReports** https://github.com/tinnlab/RCPA/issues

**NeedsCompilation** no

**Author** Ha Nguyen [aut, cre],
Phi Bya [aut],
Zeynab Maghsoudi [aut],
Tin Nguyen [fnd]

**Maintainer** Ha Nguyen <hvn0006@auburn.edu>

**Repository** CRAN

**Date/Publication** 2024-11-06 08:20:05 UTC

# Contents

downloadGEO . . . 2
getCePaPathwayCatalogue . . . 3
getCommonDEGenes . . . 4
getCommonPathways . . . 5
getEntrezAnnotation . . . 6
getGeneSets . . . 6
getSPIAKEGGNetwork . . . 7
getSupportedPlatforms . . . 8
loadData . . . 9
plotBarChart . . . 9
plotDEGeneHeatmap . . . 11
plotForest . . . 12
plotKEGGMap . . . 14
plotMA . . . 15
plotPathwayHeatmap . . . 16
plotPathwayNetwork . . . 17
plotVennDE . . . 20
plotVennPathway . . . 21
plotVolcanoDE . . . 22
plotVolcanoPathway . . . 23
processAffymetrix . . . 24
processAgilent . . . 25
runConsensusAnalysis . . . 26
runDEAnalysis . . . 27
runDEMetaAnalysis . . . 29
runGeneSetAnalysis . . . 30
runPathwayAnalysis . . . 32
runPathwayMetaAnalysis . . . 33

**Index** **35**

---

`downloadGEO` *Download GEO data*

---

## Description

This function download and process data from GEO for microarray and RNASeq data.

## Usage

```
downloadGEO(GEOID, protocol = c("affymetrix", "agilent"), platform, destDir)
```

### Arguments

| | |
|---|---|
| GEOID | The ID of the GEO dataset. |
| protocol | The protocol of selected GEO dataset. |
| platform | The platform of selected GEO dataset. |
| destDir | A path to save downloaded data. If the directory does not exist, it will be created. |

### Value

A vector of file paths to the downloaded files. The first element is the metadata file.

### Examples

```
library(RCPA)
# Affymetrix
downloadPath <- file.path(tempdir(), "GSE59761")
fileList <- RCPA::downloadGEO(GEOID = "GSE59761", protocol = "affymetrix",
                              platform ="GPL16311", destDir = downloadPath)
```

---

## getCePaPathwayCatalogue

*Get KEGG pathway catalogue (network) for CePa.ORA and CePa.GSA methods*

---

### Description

Get KEGG pathway catalogue for CePa.ORA and CePa.GSA methods

### Usage

```
getCePaPathwayCatalogue(org = "hsa", updateCache = FALSE)
```

### Arguments

| | |
|---|---|
| org | The organism abbreviation. E.g, hsa, mmu, dme, etc. To see the full list of supported organisms, visit https://www.genome.jp/kegg/catalog/org_list.html. |
| updateCache | A parameter to enable/disable cache update. |

### Value

A named list with three elements: network, names and sizes for CePa.ORA and CePa.GSA methods.

### Examples

```
cepaNetwork <- getCePaPathwayCatalogue("hsa")
```

`getCommonDEGenes` *Get common significant DE genes from multiple DE Analysis results*

## Description

Get a list of common significant DE genes from multiple DE Analysis results.

## Usage

```
getCommonDEGenes(
  DEResults,
  pThreshold = 0.05,
  useFDR = TRUE,
  stat = "logFC",
  statThreshold = 0
)
```

## Arguments

| | |
|---|---|
| `DEResults` | A list of data frames with the results of DE analysis. |
| `pThreshold` | The p-value threshold to determine if a gene is differentially expressed. |
| `useFDR` | Use the FDR adjusted p-value instead of the nominal p-value. |
| `stat` | The additional statistics column to use for filtering differentially expressed genes. |
| `statThreshold` | The absolute value of the statistic threshold to use for filtering differentially expressed genes. Default is 0, which means no filtering. |

## Value

A data frame wtih three columns: ID (Entrez IDs), Symbol and Description

## Examples

```
library(RCPA)
library(SummarizedExperiment)

affyDEExperiment <- loadData("affyDEExperiment")
agilDEExperiment <- loadData("agilDEExperiment")
RNASeqDEExperiment <- loadData("RNASeqDEExperiment")

DEResults <- list(
    "Affymetrix - GSE5281" = rowData(affyDEExperiment),
    "Agilent - GSE61196"   = rowData(agilDEExperiment),
    "RNASeq - GSE153873"   = rowData(RNASeqDEExperiment)
)

commonDEGenes <- RCPA::getCommonDEGenes(DEResults)

print(head(commonDEGenes))
```

`getCommonPathways` *Retrieve common significant pathways from multiple pathway analysis results*

## Description

Query a list of common significant pathways from multiple pathway analysis results.

## Usage

```
getCommonPathways(PAResults, pThreshold = 0.05, useFDR = TRUE)
```

## Arguments

| | |
|---|---|
| `PAResults` | A list of data frames with the results of pathway analysis. |
| `pThreshold` | The p-value threshold to determine if a pathway is enriched or significant. |
| `useFDR` | Use the FDR adjusted p-value instead of the nominal p-value. |

## Value

A data frame contains pathway ID and pathway names.

## Examples

```
library(RCPA)


affyFgseaResult <- loadData("affyFgseaResult")
agilFgseaResult <- loadData("agilFgseaResult")
RNASeqFgseaResult <- loadData("RNASeqFgseaResult")

PAResults <- list(
    "Affymetrix - GSE5281" = affyFgseaResult,
    "Agilent - GSE61196" = agilFgseaResult,
    "RNASeq - GSE153873" = RNASeqFgseaResult
)

commonPathways <- RCPA::getCommonPathways(PAResults)

print(head(commonPathways))
```

## getEntrezAnnotation *Get Entrez annotation*

### Description

This function retrieves Entrez annotation data from NCBI.

### Usage

```
getEntrezAnnotation(entrezIds)
```

### Arguments

| | |
|---|---|
| entrezIds | A vector of Entrez IDs. |

### Value

A data frame with Entrez annotation or NULL if retrieval fails. The columns are ID (Entrez ID), Symbol, Description, OtherDesignations, OtherAliases, and Chromosome.

### Examples

```
library(RCPA)
geneAnno <- getEntrezAnnotation(c("77267466", "77267467"))
```

## getGeneSets *Get gene sets*

### Description

This function retrieves gene sets for a given organism.

### Usage

```
getGeneSets(
  database = c("KEGG", "GO"),
  org = "hsa",
  taxid = 9606,
 namespace = c("biological_process", "molecular_function", "cellular_component"),
  minSize = 1,
  maxSize = 1000,
  useCache = FALSE
)
```

### Arguments

| | |
|---|---|
| database | The database of the gene sets. E.g, KEGG, GO. |
| org | The organism abbreviation. E.g, hsa, mmu, dme, etc. To see the full list of supported organisms, visit https://www.genome.jp/kegg/catalog/org_list.html. This parameter is only used when database is KEGG. |
| taxid | The NCBI taxonomy ID of the organism. This parameter is only used when database is GO. |
| namespace | The namespace of the GO terms. E.g, biological_process, molecular_function, cellular_component. |
| minSize | The minimum size of the gene sets. |
| maxSize | The maximum size of the gene sets. |
| useCache | A boolean parameter specifying if using pre-saved downloaded geneset database. It is FALSE by default. |

### Value

A named list with three elements: database, genesets and names.

### Examples

```
library(RCPA)

KEGGgenesets <- getGeneSets("KEGG", org = "hsa",
                            minSize = 10, maxSize = 1000, useCache = TRUE)

GOterms <- getGeneSets("GO", taxid = 9606,
                       namespace = "biological_process",
                       minSize = 10, maxSize = 1000, useCache = TRUE)
```

---

## getSPIAKEGGNetwork *Get KEGG pathway network for SPIA method*

---

### Description

Get KEGG pathway network for SPIA method

### Usage

```
getSPIAKEGGNetwork(org = "hsa", updateCache = FALSE)
```

### Arguments

| | |
|---|---|
| org | The organism abbreviation. E.g, hsa, mmu, dme, etc. To see the full list of supported organisms, visit https://www.genome.jp/kegg/catalog/org_list.html. |
| updateCache | A parameter to disable/enable cache update. |

### Value

A named list with three elements: network, names and sizes.

### Examples

```
spiaNetwork <- getSPIAKEGGNetwork("hsa")
```

---

## getSupportedPlatforms *Get supported platforms*

---

### Description

Get supported platforms

### Usage

```
getSupportedPlatforms()
```

### Value

A named list of supported platforms, where the names are the platform IDs and the values are the corresponding annotation packages from Bioconductor.

### Examples

```
library(RCPA)
supportedPlatforms <- getSupportedPlatforms()
```

## loadData *Load data from GitHub*

### Description

This function loads data from GitHub.

### Usage

```
loadData(name)
```

### Arguments

| | |
|---|---|
| name | The name of the data. |

### Value

Load the data with the specified name.

### Examples

```
library(RCPA)
RNASeqDataset <- loadData("RNASeqDataset")
```

## plotBarChart *Plot a bar chart of the pathway analysis results*

### Description

This function plots a bar chart of the pathway analysis results.

### Usage

```
plotBarChart(
  results,
  limit = Inf,
  label = "name",
  by = c("normalizedScore", "score", "pFDR", "p.value"),
  maxNegLog10PValue = 5,
  pThreshold = 0.05,
  useFDR = TRUE,
  selectedPathways = NULL
)
```

## Arguments

| | |
|---|---|
| results | A named list of data frame with pathway analysis results. The columns are ID, name, p.value, pFDR, size, nDE, score and normalizedScore. |
| limit | The maximum number of pathways to plot. The pathway will be sorted by the average absolute value of the -log10(p-value) or -log10(pFDR) if the 'by' parameter is 'p.value' or 'pFDR'. Otherwise, the pathway will be sorted by the average value of the 'by' parameter. |
| label | The column to use for the labels. |
| by | The column to use for the bar heights. |
| maxNegLog10PValue | The maximum -log10(p-value) to plot. |
| pThreshold | The p-value threshold to use for significance. |
| useFDR | If TRUE, use FDR adjusted p-values for the significance threshold. Otherwise, use raw p-values. This parameter is used to mark the color of the bars and is independent of the 'by' parameter. |
| selectedPathways | A vector of pathways ID, which is in the same format as ID column in the pathway analysis result, to be included in the plot. If it is NULL, all pathways will be included. |

## Value

A ggplot2 object.

## Examples

```
library(RCPA)

affyFgseaResult <- loadData("affyFgseaResult")
agilFgseaResult <- loadData("agilFgseaResult")
RNASeqFgseaResult <- loadData("RNASeqFgseaResult")
metaPAResult <- loadData("metaPAResult")

PAResults <- list(
    "Affymetrix - GSE5281" = affyFgseaResult,
    "Agilent - GSE61196" = agilFgseaResult,
    "RNASeq - GSE153873" = RNASeqFgseaResult,
    "Meta-analysis" = metaPAResult
)

selectedPathways <- c("path:hsa05010", "path:hsa05012", "path:hsa05014",
                      "path:hsa05016", "path:hsa05017", "path:hsa05020",
                      "path:hsa05022", "path:hsa04724", "path:hsa04727",
                      "path:hsa04725", "path:hsa04728", "path:hsa04726",
                     "path:hsa04720", "path:hsa04730", "path:hsa04723",
                      "path:hsa04721", "path:hsa04722")
resultsToPlot <- lapply(PAResults,
                    function(df) df[df$ID %in% selectedPathways,])
```

```
plotObj <- RCPA::plotBarChart(resultsToPlot) +
            ggplot2::ggtitle("FGSEA Analysis Results")
```

---

`plotDEGeneHeatmap` *Plot gene heatmap from a SummarizedExperiment object with DE analysis results*

---

## Description

Plot gene heatmap from a SummarizedExperiment object with DE analysis results. The heatmap contains p-values and log fold changes from the DE analysis.

## Usage

```
plotDEGeneHeatmap(
  DEResults,
  genes,
  useFDR = TRUE,
  labels = NULL,
  logFCLims = c(-5, 5),
  negLog10pValueLims = c(0, 5)
)
```

## Arguments

| | |
|---|---|
| `DEResults` | A named list of data frame of DE analysis results. |
| `genes` | A vector of gene id (e.g. Entrez IDs) to plot. The genes must be in the ID column of the data frame in DEResults. |
| `useFDR` | If TRUE, use FDR adjusted p-values. Otherwise, use raw p-values. |
| `labels` | A vector of labels for the genes. If not provided, the gene IDs will be used as labels. |
| `logFCLims` | A vector of length 2 specifying the minimum and maximum log fold change to plot. |
| `negLog10pValueLims` | A vector of length 2 specifying the minimum and maximum -log10(p-value) to plot. |

## Value

A heatmap of the genes from ggplot2.

## Examples

```
library(RCPA)
library(SummarizedExperiment)

affyDEExperiment <- loadData("affyDEExperiment")
agilDEExperiment <- loadData("agilDEExperiment")
RNASeqDEExperiment <- loadData("RNASeqDEExperiment")
metaDEResult <- loadData("metaDEResult")
genesets <- loadData("genesets")

DEResults <- list(
    "Affymetrix - GSE5281" = rowData(affyDEExperiment),
    "Agilent - GSE61196" = rowData(agilDEExperiment),
    "RNASeq - GSE153873" = rowData(RNASeqDEExperiment)
)

 metaDEResult <- metaDEResult[order(metaDEResult$pFDR),]

 alzheimerGenes <- genesets$genesets[["path:hsa05010"]]
 genesToPlot <- head(metaDEResult[metaDEResult$ID %in% alzheimerGenes, ], 50)$ID

 genesAnnotation <- RCPA::getEntrezAnnotation(genesToPlot)
 labels <- genesAnnotation[genesToPlot, "Description"]

 genesOrderByFC <- order(metaDEResult[match(genesToPlot, metaDEResult$ID), "logFC"])
 resultsToPlot <- c(DEResults, list(metaDEResult))
 names(resultsToPlot) <- c(names(DEResults), "Meta-analysis")

 plotObj <- RCPA::plotDEGeneHeatmap(
     resultsToPlot,
     genesToPlot[genesOrderByFC],
     labels = labels[genesOrderByFC],
     negLog10pValueLims = c(0, 5), logFCLims = c(-1, 1)
 )
```

---

`plotForest` *Plot pathway forest plot from pathway/geneset/meta analysis results*

---

## Description

pathways heatmap plot from pathway/geneset/meta analysis results.

## Usage

```
plotForest(
  resultsList,
```

```
  yAxis = c("ID", "name"),
  statLims = c(-2.5, 2.5),
  useFDR = TRUE,
  selectedPathways = NULL
)
```

## Arguments

| | |
|---|---|
| resultsList | A named list of dataframes from pathway analysis, geneset analysis, and/or meta analysis results. The columns are ID, name, description, p.value, pFDR, size, nDE, score and normalizedScore. |
| yAxis | The column to use for the y-axis. |
| statLims | A numeric vector of length 2 specifying the limits for score to use in the x-axis. |
| useFDR | Logical to indicate whether to use FDR or p-value. |
| selectedPathways | A vector of pathways ID, which is in the same format as ID column in the pathway analysis result, to be included in the plot. If it is NULL, all pathways will be included. |

## Value

A ggplot2 object for presenting the heatmap of the pathways.

## Examples

```
library(RCPA)
affyFgseaResult <- loadData("affyFgseaResult")
agilFgseaResult <- loadData("agilFgseaResult")
RNASeqFgseaResult <- loadData("RNASeqFgseaResult")
metaPAResult <- loadData("metaPAResult")

PAResults <- list(
    "Affymetrix - GSE5281" = affyFgseaResult,
    "Agilent - GSE61196" = agilFgseaResult,
    "RNASeq - GSE153873" = RNASeqFgseaResult,
    "Meta-analysis" = metaPAResult
)

selectedPathways <- c("path:hsa05010", "path:hsa05012", "path:hsa05014", "path:hsa05016",
                      "path:hsa05017", "path:hsa05020", "path:hsa05022", "path:hsa04724",
                      "path:hsa04727", "path:hsa04725", "path:hsa04728", "path:hsa04726",
                      "path:hsa04720", "path:hsa04730", "path:hsa04723", "path:hsa04721",
                       "path:hsa04722")
resultsToPlot <- lapply(PAResults, function(df) df[df$ID %in% selectedPathways,])

plotObj <- RCPA::plotForest(resultsToPlot, yAxis = "name", statLims = c(-3.5, 3.5))
```

plotKEGGMap *Plot KEGG map with DE genes*

## Description

This function plots KEGG map with DE genes.

## Usage

```
plotKEGGMap(
  DEResults,
  KEGGPathwayID,
  statistic = "logFC",
  useFDR = TRUE,
  pThreshold = 0.05,
  statLimit = 3
)
```

## Arguments

| | |
|---|---|
| DEResults | A named list of data frame of DE analysis results. The columns of each data frame should be at least ID, logFC, p.value and pFDR. |
| KEGGPathwayID | The KEGG pathway ID. |
| statistic | The column name of the statistic used to plot the DE genes. If statistic is p.value or pFDR, all genes are colored. Otherwise, only DE genes are colored. |
| useFDR | If TRUE, DE genes are selected based on pFDR, otherwise p.value. |
| pThreshold | The p-value threshold to select DE genes. Only used when statistic is not p.value or pFDR. |
| statLimit | The absolute value of the statistic to color the DE genes. If statistic is p.value or pFDR, this parameter is the limit of -log10(p-value). Otherwise, this parameter is the limit of the absolute value of the statistic. |

## Value

A list with the following elements:

- plot: A ggplot object of the KEGG map.
- width: The width of the KEGG map.
- height: The height of the KEGG map.

## Examples

```
library(RCPA)
library(SummarizedExperiment)

affyDEExperiment <- loadData("affyDEExperiment")
agilDEExperiment <- loadData("agilDEExperiment")
RNASeqDEExperiment <- loadData("RNASeqDEExperiment")

DEResults <- list(
    "Affymetrix - GSE5281" = rowData(affyDEExperiment),
    "Agilent - GSE61196" = rowData(agilDEExperiment),
    "RNASeq - GSE153873" = rowData(RNASeqDEExperiment)
)

plotObj <- RCPA::plotKEGGMap(DEResults, "hsa05010", stat = "logFC", pThreshold = 1, statLimit = 1)
```

---

`plotMA` *Plot MA plot from DE analysis results*

---

## Description

Plot MA plot from DE analysis results

## Usage

```
plotMA(
  DEResult,
  pThreshold = 0.05,
  useFDR = TRUE,
  logFCThreshold = 1,
  labels = NULL,
  fitMethod = "loess"
)
```

## Arguments

| | |
|---|---|
| `DEResult` | A data frame with DE analysis results. The columns are ID, p.value, pFDR, logFC, and aveExpr. |
| `pThreshold` | The p-value threshold to color significant points. |
| `useFDR` | Use FDR instead of p-value for significance. |
| `logFCThreshold` | The log2 fold change threshold to color significant points. |
| `labels` | named vector of labels to use for points, e.g., c("gene1" = "Gene 1", "gene2" = "Gene 2") |
| `fitMethod` | The method to use for fitting the loess line. If NULL then no line is drawn. |

## Value

A ggplot object.

## Examples

```
library(RCPA)
library(SummarizedExperiment)

RNASeqDEExperiment <- loadData("RNASeqDEExperiment")

plotObj <- RCPA::plotMA(rowData(RNASeqDEExperiment), logFCThreshold = 0.5) +
ggtitle("RNASeq - GSE153873")
```

---

`plotPathwayHeatmap` *Plot pathways heatmap plot from pathway/geneset/meta analysis results*

---

## Description

pathways heatmap plot from pathway/geneset/meta analysis results.

## Usage

```
plotPathwayHeatmap(
  resultsList,
  yAxis = c("ID", "name"),
  negLog10pValueLims = c(0, 5),
  useFDR = TRUE,
  selectedPathways = NULL
)
```

## Arguments

| | |
|---|---|
| `resultsList` | A named list of dataframes from pathway analysis, geneset analysis, and/or meta analysis results. The columns are ID, name, description, p.value, pFDR, size, nDE, score and normalizedScore. |
| `yAxis` | The column to use for the y-axis. |
| `negLog10pValueLims` | A vector of length 2 specifying the minimum and maximum -log10(p-value) to plot. |
| `useFDR` | Logical to indicate whether to use FDR or p-value. |
| `selectedPathways` | A vector of pathways ID, which is in the same format as ID column in the pathway analysis result, to be included in the plot. If it is NULL, all pathways will be included. |

## Value

A ggplot2 object for presenting the heatmap of the pathways.

## Examples

```
library(RCPA)
affyFgseaResult <- loadData("affyFgseaResult")
agilFgseaResult <- loadData("agilFgseaResult")
RNASeqFgseaResult <- loadData("RNASeqFgseaResult")
metaPAResult <- loadData("metaPAResult")

PAResults <- list(
    "Affymetrix - GSE5281" = affyFgseaResult,
    "Agilent - GSE61196" = agilFgseaResult,
    "RNASeq - GSE153873" = RNASeqFgseaResult,
    "Meta-analysis" = metaPAResult
)

selectedPathways <- c("path:hsa05010", "path:hsa05012", "path:hsa05014",
                      "path:hsa05016", "path:hsa05017", "path:hsa05020",
                      "path:hsa05022", "path:hsa04724", "path:hsa04727",
                      "path:hsa04725", "path:hsa04728", "path:hsa04726",
                      "path:hsa04720", "path:hsa04730", "path:hsa04723",
                      "path:hsa04721", "path:hsa04722")

resultsToPlot <- lapply(PAResults, function(df) df[df$ID %in% selectedPathways,])

plotObj <- RCPA::plotPathwayHeatmap(resultsToPlot, yAxis = "name")
```

---

# plotPathwayNetwork *Plot a pathway network*

---

## Description

This function plots a pathway network. This function needs an interactive environment with browser view support to work.

## Usage

```
plotPathwayNetwork(
  PAResults,
  genesets,
  selectedPathways = NULL,
  statistic = "pFDR",
  mode = c("continuous", "discrete"),
  pThreshold = 0.05,
  useFDR = TRUE,
```

```
  edgeThreshold = 0.5,
  statLimit = 4,
  discreteColors = NULL,
  continuousScaleFunc = NULL,
  NAColor = "#ffffff",
  borderColor = "#333333",
  nodeSizeFnc = function(id) length(genesets[[id]])^0.75,
  borderWidthFnc = function(id) 1,
  edgeWidthFnc = function(from, to) 1,
  styleFile = system.file(package = "RCPA", "extdata", "pieStyle.js"),
  file = tempfile(fileext = ".html")
)
```

## Arguments

| | |
|---|---|
| PAResults | A named list of data frame of Pathway analysis results. The columns of each data frame should be at least ID, name, p.value and pFDR. An optional column "color" can be used to specify the color of the nodes. If the column "color" is not specified, the color of the nodes will be determined by the mode and the statistic. |
| genesets | A genesets object that is obtained from getGeneSets function. |
| selectedPathways | A vector of pathway IDs to be included in the plot. If it is NULL, all pathways from genesets will be included. |
| statistic | A character value of the statistic to use for the network. The statistic should be one of the columns of the data frame in the results. |
| mode | A character value of the mode to use to color the nodes. The mode should be one of "discrete", "continuous". If the mode is "discrete", the color of the nodes are determined by whether the p-value is significant or not. If the mode is "continuous", the color of the nodes are proportional to the statistic. |
| pThreshold | A numeric value of p-value threshold. |
| useFDR | A logical value indicating whether to use FDR or not. This parameter is independent of the pThreshold. |
| edgeThreshold | A numeric from 0 to 1 indicating the threshold to draw edges. edgeThreshold of 0.1 means that edges are drawn if the number of genes in common is greater than 1% of the smaller gene set. |
| statLimit | A numeric value of the maximum absolute value of the statistic. If statistic is p.value or pFDR, this parameter is the limit of -log10(p-value). |
| discreteColors | A character vector of colors to use for the discrete mode. The length of the vector must be the same as the number of results. |
| continuousScaleFunc | A function that takes a numeric value from -1 to 1 and returns a color. |
| NAColor | A character value of the color to use for NA values. |
| borderColor | A character value of the color to use for the border of the nodes. |
| nodeSizeFnc | A function that takes a character value of the ID of the node and returns a numeric value of the size of the node. |

| | |
|---|---|
| borderWidthFnc | A function that takes a character value of the ID of the node and returns a numeric value of the width of the border of the node. |
| edgeWidthFnc | A function that takes a character value of the ID of the from node and a character value of the ID of the to node and returns a numeric value of the width of the edge. |
| styleFile | A character value of the path to the style file. If NULL, the default style file will be used, which is located at system.file(package="RCPA", "extdata", "pieStyle.js") |
| file | A character value of the path to the html file to be created. |

## Details

The function will plot a pathway network using the results of pathway analysis. The nodes of the network are the pathways and the edges are the pathways that have at least a certain number of genes in common defined by the edgeThreshold. The size of the nodes are proportional to the number of genes in the pathway. The color of the nodes are proportional to the statistic used if the mode is "continuous". If the mode is "discrete", the color of the nodes are determined by whether the p-value is significant or not. The width of the edges are proportional to the number of genes in common.

## Value

A character value of the html content of the plot.

## Examples

```
library(RCPA)

affyFgseaResult <- loadData("affyFgseaResult")
agilFgseaResult <- loadData("agilFgseaResult")
RNASeqFgseaResult <- loadData("RNASeqFgseaResult")
metaPAResult <- loadData("metaPAResult")
genesets <- loadData("genesets")

PAResults <- list(
    "Affymetrix - GSE5281" = affyFgseaResult,
    "Agilent - GSE61196" = agilFgseaResult,
    "RNASeq - GSE153873" = RNASeqFgseaResult,
    "Meta-analysis" = metaPAResult
)

genesetsToPlot <- metaPAResult$ID[order(metaPAResult$pFDR)][1:30]

pltHtml <- RCPA::plotPathwayNetwork(
    PAResults,
    genesets = genesets,
    selectedPathways = genesetsToPlot,
    edgeThreshold = 0.75,
    mode = "continuous",
    statistic = "normalizedScore"
```

```
)
```

---

plotVennDE *Plot Venn diagram from multiple DE Analysis results*

---

## Description

Plot a Venn diagram from multiple DE Analysis results.

## Usage

```
plotVennDE(
  DEResults,
  pThreshold = 0.05,
  useFDR = TRUE,
  stat = "logFC",
  statThreshold = 0,
  topToList = 10
)
```

## Arguments

| | |
|---|---|
| DEResults | A list of data frames with the results of DE analysis. |
| pThreshold | The p-value threshold to determine if a gene is differentially expressed. |
| useFDR | Use the FDR adjusted p-value instead of the raw p-value. |
| stat | The additional statistics column to use for filtering differentially expressed genes. |
| statThreshold | The absolute value of the statistic threshold to use for filtering differentially expressed genes. Default is 0, which means no filtering. |
| topToList | The number of common DE genes that are used to annotate the plot |

## Value

A ggplot2 object.

## Examples

```
library(RCPA)
library(SummarizedExperiment)

affyDEExperiment <- loadData("affyDEExperiment")
agilDEExperiment <- loadData("agilDEExperiment")
RNASeqDEExperiment <- loadData("RNASeqDEExperiment")

DEResults <- list(
```

```
    "Affymetrix - GSE5281" = rowData(affyDEExperiment),
    "Agilent - GSE61196"   = rowData(agilDEExperiment),
    "RNASeq - GSE153873"   = rowData(RNASeqDEExperiment)
)

DEResultUps <- lapply(DEResults, function(df) df[!is.na(df$logFC) & df$logFC > 0, ])

DEResultDowns <- lapply(DEResults, function(df) df[!is.na(df$logFC) & df$logFC < 0, ])

if (require("ggvenn", quietly = TRUE)){
p1 <- RCPA::plotVennDE(DEResults) +
        ggplot2::ggtitle("All DE Genes")
p2 <- RCPA::plotVennDE(DEResultUps) +
        ggplot2::ggtitle("Up-regulated DE Genes")
p3 <- RCPA::plotVennDE(DEResultDowns) +
        ggplot2::ggtitle("Down-regulated DE Genes")
}
```

---

plotVennPathway *Plot Venn diagram from multiple pathway analysis results*

---

## Description

Plot a Venn diagram from multiple pathway analysis results.

## Usage

```
plotVennPathway(PAResults, pThreshold = 0.05, useFDR = TRUE, topToList = 10)
```

## Arguments

| | |
|---|---|
| PAResults | A list of data frames with the results of pathway analysis. |
| pThreshold | The p-value threshold to determine if a pathway is enriched. |
| useFDR | Use the FDR adjusted p-value instead of the raw p-value. |
| topToList | The number of common signifcant pathways that are used to annotate the plot. |

## Value

A ggplot2 object.

## Examples

```
library(RCPA)


affyFgseaResult <- loadData("affyFgseaResult")
agilFgseaResult <- loadData("agilFgseaResult")
```

```
RNASeqFgseaResult <- loadData("RNASeqFgseaResult")
metaPAResult <- loadData("metaPAResult")

PAResults <- list(
    "Affymetrix - GSE5281" = affyFgseaResult,
    "Agilent - GSE61196" = agilFgseaResult,
    "RNASeq - GSE153873" = RNASeqFgseaResult,
    "Meta-analysis" = metaPAResult
)

PAResultUps <- lapply(PAResults, function(df) df[df$normalizedScore > 0,])

PAResultDowns <- lapply(PAResults, function(df) df[df$normalizedScore < 0,])

if (require("ggvenn", quietly = TRUE)){
p1 <- RCPA::plotVennPathway(PAResults, pThreshold = 0.05) +
    ggplot2::ggtitle("All Significant Pathways")
p2 <- RCPA::plotVennPathway(PAResultUps, pThreshold = 0.05) +
    ggplot2::ggtitle("Significantly Up-regulated Pathways")
p3 <- RCPA::plotVennPathway(PAResultDowns, pThreshold = 0.05) +
    ggplot2::ggtitle("Significantly Down-regulated Pathways")
}
```

---

`plotVolcanoDE` *Plot volcano plot from Pathway analysis results*

---

## Description

Plot volcano plot from Pathway analysis results

## Usage

```
plotVolcanoDE(DEResult, pThreshold = 0.05, useFDR = TRUE, logFCThreshold = 1)
```

## Arguments

| | |
|---|---|
| `DEResult` | A data frame with Pathway analysis results. The columns are ID, name, description, p.value, pFDR, size, nDE, score and normalizedScore. |
| `pThreshold` | The p-value threshold to use for the horizontal line. |
| `useFDR` | Whether to use the pFDR column instead of the p.value column. |
| `logFCThreshold` | The logFC threshold to use for the vertical line. |

## Value

A ggplot2 object.

## Examples

```
library(RCPA)
library(SummarizedExperiment)

affyDEExperiment <- loadData("affyDEExperiment")
agilDEExperiment <- loadData("agilDEExperiment")
RNASeqDEExperiment <- loadData("RNASeqDEExperiment")

p1 <- RCPA::plotVolcanoDE(rowData(affyDEExperiment), logFCThreshold = 0.5) +
    ggplot2::ggtitle("Affymetrix - GSE5281")
p2 <- RCPA::plotVolcanoDE(rowData(agilDEExperiment), logFCThreshold = 0.5) +
    ggplot2::ggtitle("Agilent - GSE61196")
p3 <- RCPA::plotVolcanoDE(rowData(RNASeqDEExperiment), logFCThreshold = 0.5) +
    ggplot2::ggtitle("RNASeq - GSE153873")
```

---

# plotVolcanoPathway *Plot volcano plot from Pathway analysis results*

---

## Description

Plot volcano plot from Pathway analysis results

## Usage

```
plotVolcanoPathway(
  PAResult,
  xAxis = c("normalizedScore", "score"),
  yAxis = c("-log10(pFDR)", "-log10(p.value)"),
  pThreshold = 0.05,
  label = "name",
  IDsToLabel = NULL,
  topToLabel = 10,
  sideToLabel = c("both", "left", "right")
)
```

## Arguments

| | |
|---|---|
| PAResult | A data frame with Pathway analysis results. The columns are ID, name, description, p.value, pFDR, pathwaySize, nDE, score and normalizedScore. |
| xAxis | The column to use for the x-axis. |
| yAxis | The column to use for the y-axis. |
| pThreshold | The p-value threshold to use for the horizontal line. |
| label | The column to use for the labels. Default is "name". |

| | |
|---|---|
| IDsToLabel | A vector of IDs to label. When NULL, the top pathways are labeled. Default is NULL. |
| topToLabel | The number of top pathways to label when IDsToLabels is NULL. |
| sideToLabel | The side of the plot to label. |

## Value

A ggplot2 object.

## Examples

```
library(RCPA)
affyFgseaResult <- loadData("affyFgseaResult")
agilFgseaResult <- loadData("agilFgseaResult")
RNASeqFgseaResult <- loadData("RNASeqFgseaResult")
metaPAResult <- loadData("metaPAResult")

p1 <- RCPA::plotVolcanoPathway(affyFgseaResult, sideToLabel = "left")
p2 <- RCPA::plotVolcanoPathway(agilFgseaResult, sideToLabel = "left")
p3 <- RCPA::plotVolcanoPathway(RNASeqFgseaResult, sideToLabel = "left")
p4 <- RCPA::plotVolcanoPathway(metaPAResult, sideToLabel = "left")
```

---

# processAffymetrix *Process and normalize affymetrix-based dataset*

---

## Description

This function process CEL files and normalize expression data

## Usage

```
processAffymetrix(dir, samples = NULL)
```

## Arguments

| | |
|---|---|
| dir | The path to the directory containing CEL files. |
| samples | A vector of samples IDs. If NULL, the function will automatically detect the samples in the directory. |

## Details

Read and normalize expression data for affymetrix using RMA method

## Value

A matrix of normalized expression data. Rows are probes and columns are samples.

## Examples

```
library(RCPA)
geoId <- "GSE59761"
downloadPath <- file.path(tempdir(), geoId)
fileList <- RCPA::downloadGEO(GEOID = geoId, protocol = "affymetrix",
                              platform ="GPL16311", destDir = downloadPath)
# process only 3 samples
expression <- RCPA::processAffymetrix(downloadPath,
                  samples = c("GSM1446171", "GSM1446172", "GSM1446173"))
```

---

`processAgilent` *Process and normalize agilent-based dataset*

---

## Description

This function process TXT files and normalize expression data

## Usage

```
processAgilent(dir, samples = NULL, greenOnly)
```

## Arguments

| | |
|---|---|
| `dir` | The path to the directory containing TXT files. |
| `samples` | A vector of samples IDs. If NULL, the function will automatically detect the samples in the directory. |
| `greenOnly` | Logical, for use with source, should the green (Cy3) channel only be read, or are both red and green required. |

## Details

Read and normalize expression data for agilent using limma normexp, loess, and quantile methods

## Value

A matrix of normalized expression data. Rows are probes and columns are samples.

## Examples

```
library(RCPA)
geoId <- "GSE28522"

downloadPath <- file.path(tempdir(), geoId)
fileList <- RCPA::downloadGEO(GEOID = geoId, protocol = "agilent",
                              platform ="GPL4133", destDir = downloadPath)

expression <- RCPA::processAgilent(downloadPath, greenOnly = FALSE)
```

`runConsensusAnalysis` *Perform Consensus Analysis*

## Description

This function performs consensus analysis using two methods. These methods are weighted.mean and RRA.

## Usage

```
runConsensusAnalysis(
  PAResults,
  method = c("weightedZMean", "RRA"),
  weightsList = NULL,
  useFDR = TRUE,
  rank.by = c("normalizedScore", "pFDR", "both"),
  backgroundSpace = NULL
)
```

## Arguments

| | |
|---|---|
| `PAResults` | A list of at least length two from enrichment analysis results. |
| `method` | The consensus analsyis method. This can be either weighted.mean or RRA. |
| `weightsList` | A vector of integer values. Each element shows the corresponding input result weight. When selected method is weighted.mean this parameter needs to be specified. If it is null all the weights are considered as equal. |
| `useFDR` | A logical parameter, indicating if adjusted p-values should be used. |
| `rank.by` | An string parameter which specifies how the input results should be ranked. This parameter is used when the selected method is RRA. |
| `backgroundSpace` | A list of lists with the same length as PAResults. Each list contains underlying space (set of pathways) for each input dataframe in PAResults. This parameter is optional. NULL means all input dataframes share a common space. So the union pathways of all input dataframes is taken into account. |

## Value

A dataframe of consensus analysis result, which contains the following columns:

- ID: The ID of pathway
- p.value: The p-value of pathway
- pFDR: The adjusted p-value using Benjamini-Hochberg method
- name: The name of pathway
- pathwaySize: The size of pathway

## Examples

```
library(RCPA)

affyFgseaResult <- loadData("affyFgseaResult")
agilFgseaResult <- loadData("agilFgseaResult")
RNASeqFgseaResult <- loadData("RNASeqFgseaResult")

consensusPAResult <- RCPA::runConsensusAnalysis(
    list(affyFgseaResult, agilFgseaResult, RNASeqFgseaResult),
    method = "weightedZMean"
)

print(head(consensusPAResult))
```

---

`runDEAnalysis` *Differential expression analysis*

---

## Description

This function performs differential expression analysis using either limma, DESeq2 or edgeR.

## Usage

```
runDEAnalysis(
  summarizedExperiment,
  method = c("limma", "DESeq2", "edgeR"),
  design,
  contrast,
  annotation = NULL
)
```

## Arguments

| | |
|---|---|
| `summarizedExperiment` | SummarizedExperiment object |
| `method` | Method to use for differential expression analysis. Can be "limma", "DESeq2" or "edgeR". |
| `design` | A design model output by model.matrix. |
| `contrast` | A contrast matrix. See limma::makeContrasts. |
| `annotation` | A data frame mapping between probe IDs and entrez gene IDs. If not provided, the function will try to get the mapping from the platform annotation in the SummarizedExperiment object. If the annotation is not available, the function will return the probe IDs. Regardless of the type of annotation, it must contains two columns: FROM and TO, where FROM is the probe ID and TO is the entrez gene ID. |

## Value

A SummarizedExperiment object with DE analysis results appended to the rowData slot with the following columns:

- ID: gene ID. If annotation is provided, this will be the entrez gene ID. Otherwise, it will be the probe ID.
- logFC: log2 fold change
- p.value: p-value from the DE analysis using the specified method
- pFDR: p-value adjusted for multiple testing using Benjamini-Hochberg method
- statistic: statistic from the DE analysis using the specified method. For limma, this is the t-statistic. For DESeq2, this is the Wald statistic. For edgeR, this is the log fold change.
- avgExpr: For limma, it is the average expression. For DESeq2, it is the log base mean. For edgeR, it is the log CPM.
- logFCSE: standard error of the log fold change.
- sampleSize: sample size used for DE analysis.

The assay slot will contain the input expression/count matrix, and the rownames will be mapped to the gene IDs if annotation is found in the input SummarizedExperiment object or in the annotation parameter. Other slots will be the same as in the input SummarizedExperiment object.

## Examples

```
library(RCPA)
library(SummarizedExperiment)

# GSE5281
affyDataset <- loadData("affyDataset")
affyDesign <- model.matrix(~0 + condition + region + condition:region,
                           data = colData(affyDataset))
colnames(affyDesign) <- make.names(colnames(affyDesign))
affyContrast <- limma::makeContrasts(conditionalzheimer-conditionnormal,
                                     levels=affyDesign)

if (require("hgu133plus2.db", quietly = TRUE)){
affyDEExperiment <- RCPA::runDEAnalysis(affyDataset, method = "limma",
                                        design = affyDesign,
                                        contrast = affyContrast,
                                        annotation = "GPL570")

# check the DE analysis results

print(head(rowData(affyDEExperiment)))
}



# GSE61196
agilDataset <- loadData("agilDataset")
agilDesign <- model.matrix(~0 + condition,
```

```
                            data = colData(agilDataset))
agilContrast <- limma::makeContrasts(conditionalzheimer-conditionnormal,
                                     levels=agilDesign)

# Create Probe mapping
options(timeout = 3600)
GPL4133Anno <- GEOquery::dataTable(GEOquery::getGEO("GPL4133"))@table
GPL4133GeneMapping <- data.frame(FROM = GPL4133Anno$SPOT_ID,
                                 TO = as.character(GPL4133Anno$GENE),
                                 stringsAsFactors = FALSE)
GPL4133GeneMapping <- GPL4133GeneMapping[!is.na(GPL4133GeneMapping$TO), ]

agilDEExperiment <- RCPA::runDEAnalysis(agilDataset, method = "limma",
                                        design = agilDesign,
                                        contrast = agilContrast,
                                        annotation = GPL4133GeneMapping)
print(head(rowData(agilDEExperiment)))

# GSE153873
RNASeqDataset <- loadData("RNASeqDataset")
RNASeqDesign <- model.matrix(~0 + condition, data = colData(RNASeqDataset))
RNASeqContrast <- limma::makeContrasts(conditionalzheimer-conditionnormal,
                                       levels=RNASeqDesign)


if (require("org.Hs.eg.db", quietly = TRUE)){
    GeneSymbolMapping <- AnnotationDbi::select(org.Hs.eg.db,
                                               keys = rownames(RNASeqDataset),
                                               columns = c("SYMBOL", "ENTREZID"),
                                               keytype = "SYMBOL")
    colnames(GeneSymbolMapping) <- c("FROM", "TO")

    RNASeqDEExperiment <- RCPA::runDEAnalysis(RNASeqDataset,
                            method = "DESeq2",
                            design = RNASeqDesign,
                            contrast = RNASeqContrast,
                            annotation = GeneSymbolMapping)
    print(head(rowData(RNASeqDEExperiment)))
}
```

---

`runDEMetaAnalysis` *Combine DE analysis results*

---

## Description

This function performs mata analysis on multiple DE analysis results.

### Usage

```
runDEMetaAnalysis(
  DEResults,
  method = c("stouffer", "fisher", "addCLT", "geoMean", "minP", "REML")
)
```

### Arguments

| | |
|---|---|
| DEResults | A list of dataframes containing DE analysis results. Each dataframe must have ID, p.value, logFC and logFCSE columns. |
| method | The method to combine p-values. It can be one of "fisher", "stouffer", "geoMean", "addCLT", "minP", or "REML". |

### Value

A dataframe containing combined DE analysis results. The dataframe has ID, p.value, pDFR, logFC, and logFCSE columns.

### Examples

```
library(RCPA)
library(SummarizedExperiment)
affyDEExperiment <- loadData("affyDEExperiment")
agilDEExperiment <- loadData("agilDEExperiment")
RNASeqDEExperiment <- loadData("RNASeqDEExperiment")

metaDEResult <- RCPA::runDEMetaAnalysis(list(
    rowData(affyDEExperiment)[1:1000,],
    rowData(agilDEExperiment)[1:1000,],
    rowData(RNASeqDEExperiment)[1:1000,]
), method = "stouffer")
```

---

## runGeneSetAnalysis *Gene Set Enrichment Analysis*

---

### Description

This function performs gene set enrichment analysis using either ORA, fgsea, GSA, ks, or wilcox approaches.

### Usage

```
runGeneSetAnalysis(
  summarizedExperiment,
  genesets,
  method = c("ora", "fgsea", "gsa", "ks", "wilcox"),
```

```
  ORAArgs = list(pThreshold = 0.05),
 FgseaArgs = list(sampleSize = 101, minSize = 1, maxSize = Inf, eps = 1e-50, scoreType =
   "std", nproc = 0, gseaParam = 1, BPPARAM = NULL, nPermSimple = 1000, absEps = NULL),
 GSAArgs = list(method = "maxmean", random.seed = NULL, knn.neighbors = 10, s0 = NULL,
    s0.perc = NULL, minsize = 15, maxsize = 500, restand = TRUE, restand.basis =
   "catalog", nperms = 200, xl.mode = "regular", xl.time = NULL, xl.prevfit = NULL)
)
```

## Arguments

| | |
|---|---|
| summarizedExperiment | The generated SummarizedExpriment object from DE analysis result. |
| genesets | The gene sets definition, ex. KEGG genesets from getGeneSets function. |
| method | The gene set enrichment analsyis method, including ORA, fgsea, GSA, ks, and wilcox. |
| ORAArgs | A list of other passed arguments to ORA. pThreshold is used as p.value cutoff to pick DE genes. |
| FgseaArgs | A list of other passed arguments to fgsea. See fgsea function. |
| GSAArgs | A list of other passed arguments to GSA. See GSA function. |

## Value

A dataframe of gene set enrichment analysis result, which contains the following columns:

- ID: The ID of the gene set
- p.value: The p-value of the gene set
- pFDR: The adjusted p-value of the gene set using the Benjamini-Hochberg method
- score: The enrichment score of the gene set
- normalizedScore: The normalized enrichment score of the gene set
- sampleSize: The total number of samples in the study
- name: The name of the gene set
- pathwaySize: The size of the gene set

The returned data frame is sorted based on the pathways' nominal p-values.

## Examples

```
library(RCPA)

RNASeqDEExperiment <- loadData("RNASeqDEExperiment")
genesets <- loadData("genesets")


oraResult <- runGeneSetAnalysis(RNASeqDEExperiment, genesets,
                                method = "ora",
                                ORAArgs = list(pThreshold = 0.05))
```

```
print(head(oraResult))

fgseaResult <- runGeneSetAnalysis(RNASeqDEExperiment, genesets,
                                  method = "fgsea",
                                  FgseaArgs = list(minSize = 10, maxSize = Inf))
print(head(fgseaResult))
```

---

runPathwayAnalysis *Topology-based Pathway Analysis*

---

## Description

This function performs patwhay analysis using SPIA, CePaORA, and CePaGSA methods.

## Usage

```
runPathwayAnalysis(
  summarizedExperiment,
  network,
  method = c("spia", "cepaORA", "cepaGSA"),
 SPIAArgs = list(all = NULL, nB = 2000, verbose = TRUE, beta = NULL, combine = "fisher",
    pThreshold = 0.05),
 CePaORAArgs = list(bk = NULL, cen = c("equal.weight", "in.degree", "out.degree",
  "betweenness", "in.reach", "out.reach"), cen.name = c("equal.weight", "in.degree",
  "out.degree", "betweenness", "in.reach", "out.reach"), iter = 1000, pThreshold =
    0.05),
 CePaGSAArgs = list(cen = c("equal.weight", "in.degree", "out.degree", "betweenness",
  "in.reach", "out.reach"), cen.name = c("equal.weight", "in.degree", "out.degree",
  "betweenness", "in.reach", "out.reach"), nlevel = "tvalue_abs", plevel = "mean", iter
    = 1000)
)
```

## Arguments

| | |
|---|---|
| summarizedExperiment | The generated SummarizedExpriment object from DE analysis result. |
| network | The pathways network object. |
| method | The pathway analsyis method, including SPIA, cepaORA, and cepaGSA. |
| SPIAArgs | A list of other passed arguments to spia. See spia function. |
| CePaORAArgs | A list of other passed arguments to CePaORA. See CePa function. |
| CePaGSAArgs | A list of other passed arguments to CePaGSA. See CePa function. |

## Value

A dataframe of pathway analysis result, which contains the following columns

- ID: The ID of the gene set
- p.value: The p-value of the gene set
- pFDR: The adjusted p-value of the gene set using the Benjamini-Hochberg method
- score: The enrichment score of the gene set
- normalizedScore: The normalized enrichment score of the gene set
- sampleSize: The total number of samples in the study
- name: The name of the gene set
- pathwaySize: The size of the gene set

## Examples

```
library(RCPA)
RNASeqDEExperiment <- loadData("RNASeqDEExperiment")
spiaNetwork <- loadData("spiaNetwork")
cepaNetwork <- loadData("cepaNetwork")

spiaResult <- runPathwayAnalysis(RNASeqDEExperiment, spiaNetwork, method = "spia")
cepaORAResult <- runPathwayAnalysis(RNASeqDEExperiment, cepaNetwork, method = "cepaORA")
```

---

# runPathwayMetaAnalysis

*Perform Meta Analysis*

---

## Description

This function performs meta analysis on multiple pathway analysis results.

## Usage

```
runPathwayMetaAnalysis(
  PAResults,
  method = c("stouffer", "fisher", "addCLT", "geoMean", "minP", "REML")
)
```

## Arguments

| | |
|---|---|
| PAResults | A list of at least size two of data frames obtained from pathway analysis |
| method | A method used to combine pathway analysis results, which can be "stouffer", "fisher", "addCLT", "geoMean", "minP", or "REML" |

## Details

This function performs meta-analysis on multiple pathway analysis results.

## Value

A dataframe of meta analysis results including the following columns:

- ID: The ID of pathway
- name: The name of pathway
- p.value: The meta p-value of pathway
- pFDR: The adjusted meta p-value of pathway using Benjamini-Hochberg method
- score: The combined score of pathway
- normalizedScore: The combined normalized score of pathway
- pathwaySize: The size of pathway

## Examples

```
library(RCPA)
affyFgseaResult <- loadData("affyFgseaResult")
agilFgseaResult <- loadData("agilFgseaResult")
RNASeqFgseaResult <- loadData("RNASeqFgseaResult")

metaPAResult <- RCPA::runPathwayMetaAnalysis(
    list(affyFgseaResult, agilFgseaResult, RNASeqFgseaResult),
    method = "stouffer"
)
```

# Index

downloadGEO, 2

getCePaPathwayCatalogue, 3
getCommonDEGenes, 4
getCommonPathways, 5
getEntrezAnnotation, 6
getGeneSets, 6
getSPIAKEGGNetwork, 7
getSupportedPlatforms, 8

loadData, 9

plotBarChart, 9
plotDEGeneHeatmap, 11
plotForest, 12
plotKEGGMap, 14
plotMA, 15
plotPathwayHeatmap, 16
plotPathwayNetwork, 17
plotVennDE, 20
plotVennPathway, 21
plotVolcanoDE, 22
plotVolcanoPathway, 23
processAffymetrix, 24
processAgilent, 25

runConsensusAnalysis, 26
runDEAnalysis, 27
runDEMetaAnalysis, 29
runGeneSetAnalysis, 30
runPathwayAnalysis, 32
runPathwayMetaAnalysis, 33