

# Package ‘DysPIA’

January 20, 2025

**Type** Package

**Title** Dysregulated Pathway Identification Analysis

**Version** 1.3

**Date** 2020-06-26

**Maintainer** Limei Wang <lemon619@gmail.com>

**Description** It is used to identify dysregulated pathways based on a pre-ranked gene pair list. A fast algorithm is used to make the computation really fast. The data in package 'DysPIAData' is needed.

**License** GPL (>= 2)

**Depends** R (>= 3.5.0), DysPIAData

**Imports** Rcpp (>= 1.0.4), BiocParallel, fastmatch, data.table, stats, parmigene

**LinkingTo** Rcpp

**RoxygenNote** 7.1.0

**Encoding** UTF-8

**LazyData** true

**NeedsCompilation** yes

**Author** Limei Wang [aut, cre],  
Jin Li [aut, ctb]

**Repository** CRAN

**Date/Publication** 2020-07-10 05:10:03 UTC

## Contents

calcDyspiaStat . . . . .	2
calcDyspiaStatCumulative . . . . .	3
calcDyspiaStatCumulativeBatch . . . . .	3
calEdgeCorScore_ESEA . . . . .	4
class.labels_p53 . . . . .	5
DysGPS . . . . .	5
DysGPS_p53 . . . . .	6

DysPIA . . . . .	6
DyspiaRes_p53 . . . . .	8
DyspiaSig . . . . .	8
DyspiaSimpleImpl . . . . .	9
gene_expression_p53 . . . . .	10
sample_background . . . . .	10
setUpBPPARAM . . . . .	11

## Index 12

---

calcDyspiaStat	<i>calcDyspiaStat: Calculates DysPIA statistics</i>
----------------	---

---

### Description

Calculates DysPIA statistics for a given query gene pair set.

### Usage

```
calcDyspiaStat(
  stats,
  selectedStats,
  DyspiaParam = 1,
  returnAllExtremes = FALSE,
  returnLeadingEdge = FALSE
)
```

### Arguments

stats	Named numeric vector with gene pair-level statistics sorted in decreasing order (order is not checked).
selectedStats	Indexes of selected gene pairs in the 'stats' array.
DyspiaParam	DysPIA weight parameter (0 is unweighted, suggested value is 1).
returnAllExtremes	If TRUE return not only the most extreme point, but all of them. Can be used for enrichment plot.
returnLeadingEdge	If TRUE return also leading edge gene pairs.

### Value

Value of DysPIA statistic if both returnAllExtremes and returnLeadingEdge are FALSE. Otherwise returns list with the following elements:

- res – value of DysPIA statistic
- tops – vector of top peak values of cumulative enrichment statistic for each gene pair;
- bottoms – vector of bottom peak values of cumulative enrichment statistic for each gene pair;
- leadingEdge – vector with indexes of leading edge gene pairs that drive the enrichment.

---

`calcDyspiaStatCumulative`*Calculates DysPIA statistic values for all the prefixes of a gene pair set*

---

**Description**

Calculates DysPIA statistic values for all the prefixes of a gene pair set

**Usage**

```
calcDyspiaStatCumulative(stats, selectedStats, DyspiaParam)
```

**Arguments**

<code>stats</code>	Named numeric vector with gene pair-level statistics sorted in decreasing order (order is not checked)
<code>selectedStats</code>	indexes of selected gene pairs in a 'stats' array
<code>DyspiaParam</code>	DysPIA weight parameter (0 is unweighted, suggested value is 1)

**Value**

Numeric vector of DysPIA statistics for all prefixes of selectedStats.

---

`calcDyspiaStatCumulativeBatch`*Calculates DysPIA statistic values for the gene pair sets*

---

**Description**

Calculates DysPIA statistic values for the gene pair sets

**Usage**

```
calcDyspiaStatCumulativeBatch(  
  stats,  
  DyspiaParam,  
  pathwayScores,  
  pathwaySizes,  
  iterations,  
  seed  
)
```

**Arguments**

<code>stats</code>	Named numeric vector with gene pair-level statistics sorted in decreasing order (order is not checked).
<code>DyspiaParam</code>	DysPIA weight parameter (0 is unweighted, suggested value is 1).
<code>pathwayScores</code>	Vector with enrichment scores for the pathways in the database.
<code>pathwaysSizes</code>	Vector of pathway sizes.
<code>iterations</code>	Number of iterations.
<code>seed</code>	Seed vector

**Value**

List of DysPIA statistics for gene pair sets.

---

`calEdgeCorScore_ESEA`    *calEdgeCorScore\_ESE*

---

**Description**

Calculates differential Mutual information.

**Usage**

```
calEdgeCorScore_ESEA(
  dataset,
  class.labels,
  controlcharacter,
  casecharacter,
  background
)
```

**Arguments**

<code>dataset</code>	Matrix of gene expression values (rownames are genes, columnnames are samples).
<code>class.labels</code>	Vector of binary labels.
<code>controlcharacter</code>	Character of control in the class labels.
<code>casecharacter</code>	Character of case in the class labels.
<code>background</code>	Matrix of the edges' background.

**Value**

A vector of the aberrant correlation in phenotype P based on mutual information (MI) for each edge.

**Examples**

```
data(gene_expression_p53, class.labels_p53, sample_background)
ESEAscore_p53<-calEdgeCorScore_ESEA(gene_expression_p53, class.labels_p53,
  "WT", "MUT", sample_background)
```

---

class.labels_p53	<i>Example vector of category labels.</i>
------------------	---

---

**Description**

The labels for the 50 cell lines in p53 data. Control group's label is 'WT', case group's label is 'MUT'.

**Usage**

```
data(class.labels_p53)
```

---

DysGPS	<i>DysGPS: Calculates Dysregulated gene pair score (DysGPS) for each gene pair</i>
--------	--

---

**Description**

Calculates Dysregulated gene pair score (DysGPS) for each gene pair. Two-sample Welch's T test of gene pairs between case and control samples. The package 'DysPIAData' including the background data is needed to be loaded.

**Usage**

```
DysGPS(  
  dataset,  
  class.labels,  
  controlcharacter,  
  casecharacter,  
  background = combined_background  
)
```

**Arguments**

dataset	Matrix of gene expression values (rownames are genes, columnnames are samples).
class.labels	Vector of category labels.
controlcharacter	Charactor of control group in the class labels.
casecharacter	Charactor of case group in the class labels.
background	Matrix of the gene pairs' background. The default is 'combined_background', which includes real pathway gene pairs and randomly produced gene pairs. The 'combined_background' was include in 'DysPIAData'.

**Value**

A vector of DysGPS for each gene pair.

**Examples**

```
data(gene_expression_p53, class.labels_p53, sample_background)
DysGPS_sample<-DysGPS(gene_expression_p53, class.labels_p53,
  "WT", "MUT", sample_background)
```

---

DysGPS\_p53

*Example vector of DysGPS in p53 data.*


---

**Description**

The score vector of 164923 gene pairs from p53 dataset. It can be loaded from the example datasets of R-package 'DysPIA', and also can be obtained by running DysGPS(), details see DysGPS.R

**Usage**

```
data(DysGPS_p53)
```

---

DysPIA

*DysPIA: Dysregulated Pathway Identification Analysis*


---

**Description**

Runs Dysregulated Pathway Identification Analysis (DysPIA).The package 'DysPIAData' including the background data is needed to be loaded.

**Usage**

```
DysPIA(
  pathwayDB = "kegg",
  stats,
  nperm = 10000,
  minSize = 15,
  maxSize = 1000,
  nproc = 0,
  DyspiaParam = 1,
  BPPARAM = NULL
)
```

**Arguments**

pathwayDB	Name of the pathway database (8 databases:reactome,kegg,biocarta,panther,pathbank,nci,smpdb,pharmgk). The default value is "kegg".
stats	Named vector of CILP scores for each gene pair. Names should be the same as in pathways.
nperm	Number of permutations to do. Minimal possible nominal p-value is about 1/nperm. The default value is 10000.
minSize	Minimal size of a gene pair set to test. All pathways below the threshold are excluded. The default value is 15.
maxSize	Maximal size of a gene pair set to test. All pathways above the threshold are excluded. The default value is 1000.
nproc	If not equal to zero sets BPPARAM to use nproc workers (default = 0).
DyspiaParam	DysPIA parameter value, all gene pair-level status are raised to the power of 'DyspiaParam' before calculation of DysPIA enrichment scores.
BPPARAM	Parallelization parameter used in bplapply. Can be used to specify cluster to run. If not initialized explicitly or by setting 'nproc' default value 'bpparam()' is used.

**Value**

A table with DysPIA results. Each row corresponds to a tested pathway. The columns are the following:

- pathway – name of the pathway as in 'names(pathway)';
- pval – an enrichment p-value;
- padj – a BH-adjusted p-value;
- DysPS – enrichment score, same as in Broad DysPIA implementation;
- NDysPS – enrichment score normalized to mean enrichment of random samples of the same size;
- nMoreExtreme – a number of times a random gene pair set had a more extreme enrichment score value;
- size – size of the pathway after removing gene pairs not present in 'names(stats)';
- leadingEdge – vector with indexes of leading edge gene pairs that drive the enrichment.

**Examples**

```
data(pathway_list,package="DysPIAData")
data(DysGPS_p53)
DyspiaRes_p53 <- DysPIA("kegg", DysGPS_p53, nperm = 100, minSize = 20, maxSize = 100)
```

---

DyspiaRes_p53	<i>Example list of DysPIA result in p53 data.</i>
---------------	---

---

**Description**

The list includes 81 pathway results from 'DisPIA.R' as an example used in 'DyspiaSig.R'.

**Usage**

```
data(DyspiaRes_p53)
```

---

DyspiaSig	<i>DyspiaSig</i>
-----------	------------------

---

**Description**

Returns the significant summary of DysPIA results.

**Usage**

```
DyspiaSig(DyspiaRes, fdr)
```

**Arguments**

DyspiaRes	Table with results of running DysPIA().
fdr	Significant threshold of 'padj' (a BH-adjusted p-value).

**Value**

A list of significant DysPIA results, including correlation gain and correlation loss.

**Examples**

```
data(pathway_list,package="DysPIAData")
data(DyspiaRes_p53)
summary_p53 <- DyspiaSig(DyspiaRes_p53, 0.05) # filter with padj<0.05
```



---

DyspiaSimpleImpl	<i>DyspiaSimpleImpl</i>
------------------	-------------------------

---

## Description

Runs dysregulated pathway identification analysis for preprocessed input data.

## Usage

```
DyspiaSimpleImpl(
  pathwayScores,
  pathwaysSizes,
  pathwaysFiltered,
  leadingEdges,
  permPerProc,
  seeds,
  toKeepLength,
  stats,
  BPPARAM
)
```

## Arguments

pathwayScores	Vector with enrichment scores for the pathways in the database.
pathwaysSizes	Vector of pathway sizes.
pathwaysFiltered	Filtered pathways.
leadingEdges	Leading edge gene pairs.
permPerProc	Parallelization parameter for permutations.
seeds	Seed vector
toKeepLength	Number of ‘pathways’ that meet the condition for ‘minSize’ and ‘maxSize’.
stats	Named vector of gene pair-level scores. Names should be the same as in pathways of ‘pathwayDB’.
BPPARAM	Parallelization parameter used in bplapply. Can be used to specify cluster to run. If not initialized explicitly or by setting ‘nproc’ default value ‘bpparam()’ is used.

## Value

A table with DysPIA results. Each row corresponds to a tested pathway. The columns are the following:

- pathway – name of the pathway as in ‘names(pathway)’;
- pval – an enrichment p-value;

- padj – a BH-adjusted p-value;
- DysPS – enrichment score, same as in Broad DysPIA implementation;
- NDysPS – enrichment score normalized to mean enrichment of random samples of the same size;
- nMoreExtreme<sup>4</sup> – a number of times a random gene pair set had a more extreme enrichment score value;
- size – size of the pathway after removing gene pairs not present in 'names(stats)';
- leadingEdge – vector with indexes of leading edge gene pairs that drive the enrichment.

---

gene\_expression\_p53     *Example matrix of gene expression value.*

---

### Description

A dataset of transcriptional profiles from p53+ and p53 mutant cancer cell lines. It includes the normalized gene expression for 6385 genes in 50 samples. Rownames are genes, columnnames are samples.

### Usage

```
data(gene_expression_p53)
```

---

sample\_background     *Example list of gene pair background.*

---

### Description

The list of background was used in "DysGPS.R" and 'calEdgeCorScore\_ESEA.R' which is a part of the 'combined\_background' in 'DysPIAData'.

### Usage

```
data(sample_background)
```

---

<code>setUpBPPARAM</code>	<i>setUpBPPARAM</i>
---------------------------	---------------------

---

**Description**

Sets up parameter BPPARAM value.

**Usage**

```
setUpBPPARAM(nproc = 0, BPPARAM = NULL)
```

**Arguments**

<code>nproc</code>	If not equal to zero sets BPPARAM to use nproc workers (default = 0).
<code>BPPARAM</code>	Parallelization parameter used in <code>bplapply</code> . Can be used to specify cluster to run. If not initialized explicitly or by setting ‘nproc’ default value ‘ <code>bpparam()</code> ’ is used.

**Value**

parameter BPPARAM value

# Index

`calcDyspiaStat`, [2](#)  
`calcDyspiaStatCumulative`, [3](#)  
`calcDyspiaStatCumulativeBatch`, [3](#)  
`calEdgeCorScore_ESEA`, [4](#)  
`class.labels_p53`, [5](#)

`DysGPS`, [5](#)  
`DysGPS_p53`, [6](#)  
`DysPIA`, [6](#)  
`DyspiaRes_p53`, [8](#)  
`DyspiaSig`, [8](#)  
`DyspiaSimpleImpl`, [9](#)

`gene_expression_p53`, [10](#)

`sample_background`, [10](#)  
`setUpBPPARAM`, [11](#)