

# Package ‘immcp’

November 14, 2020

**Title** Candidate Prescriptions Discovery Based on Pathway Fingerprint

**Version** 0.9.2

**Description** The pathway fingerprint is a method to indicate the profile of significant pathways being influenced by drugs, which may hint drug functions. Through the similarity of pathway fingerprints, the potential relationship between disease and prescription can be found. Ye (2012) <doi: 10.1007/s13238-012-2011-z>.

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1

**biocViews**

**Depends** R (>= 3.5.0)

**Imports** igraph, proxyC, pbapply, Matrix, magrittr, stats, utils,  
clusterProfiler, GSEABase, methods, ggplot2, visNetwork,  
RobustRankAggreg, reshape2, corrr, GSVa, Boruta, limma

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**BugReports** <https://github.com/YuanlongHu/immcp/issues>

**URL** <https://github.com/YuanlongHu/immcp>

**NeedsCompilation** no

**Author** Yuanlong Hu [aut, cre] (<<https://orcid.org/0000-0003-1933-7570>>)

**Maintainer** Yuanlong Hu <[huyuanlong1996@163.com](mailto:huyuanlong1996@163.com)>

**Repository** CRAN

**Date/Publication** 2020-11-14 13:00:05 UTC

**R topics documented:**

immcp-package . . . . .	2
as.data.frame . . . . .	3
BasicData-class . . . . .	3
CreateBasicData . . . . .	4
drugResult . . . . .	4
drugSample . . . . .	4
extrFP . . . . .	5
getDEG . . . . .	6
getF . . . . .	6
get_result . . . . .	7
head . . . . .	8
imm_centr . . . . .	8
overlap_pathway . . . . .	9
plot_density . . . . .	10
plot_network . . . . .	11
PrepareData . . . . .	13
read_gmt . . . . .	14
res_rank . . . . .	14
ScoreFP-class . . . . .	15
ScoreResult-class . . . . .	15
score_fp . . . . .	16
score_network . . . . .	17
simFP . . . . .	18
tail . . . . .	18
to_df . . . . .	19
to_list . . . . .	19
viewpathway . . . . .	20
write_gmt . . . . .	20
<b>Index</b>	<b>22</b>

immcp-package

*Candidate Prescriptions Discovery Based on Pathway Fingerprint***Description**

The pathway fingerprint is a method to indicate the profile of significant pathways being influenced by drugs, which may hint drug functions. Through the similarity of pathway fingerprints, the potential relationship between disease and prescription can be found. Ye (2012) <doi: 10.1007/s13238-012-2011-z>.

---

as.data.frame	<i>Coerce a ScoreResult object into a data frame</i>
---------------	--

---

**Description**

Coerce a ScoreResult object into a data frame

**Usage**

```
## S4 method for signature 'ScoreResult'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

**Arguments**

x	A ScoreResult object
row.names	NULL or a character vector giving the row names for the data frame. Missing values are not allowed.
optional	logical. If TRUE, setting row names and converting column names (to syntactic names: see make.names) is optional. Note that all of R's base package as.data.frame() methods use optional only for column names treatment, basically with the meaning of data.frame(*, check.names = !optional). See also the make.names argument of the matrix method.
...	other arguments

**Author(s)**

Yuanlong Hu

---

BasicData-class	<i>Class "BasicData" This class represents the basic input data.</i>
-----------------	--

---

**Description**

Class "BasicData" This class represents the basic input data.

**Slots**

BasicData	Alist containing basic data.
Key	Column name of basic data.
Relationship	Relationship.
CompoundAnno	Compound ID

**Author(s)**

Yuanlong Hu

CreateBasicData      *CreateBasicData*

---

**Description**

Prepare input data.

**Usage**

```
CreateBasicData(...)
```

**Arguments**

...      A data frame from PrepareData.

**Value**

a list

**Author(s)**

Yuanlong Hu

**Examples**

```
data("drugSample")
drug_herb <- PrepareData(drugSample$drug_herb, col1 = "drug", col2 = "herb")
herb_target <- PrepareData(drugSample$herb_target,
                           col1 = "herb", col2 = "target",
                           format = "basket", sep = ", ")
drug_target <- CreateBasicData(drug_herb, herb_target)
```

---

drugResult      *A demo dataset contains all result.*

---

**Description**

A demo dataset contains all result.

---

drugSample      *A demo dataset contains a sample of herbal prescription.*

---

**Description**

A demo dataset contains a sample of herbal prescription.

---

`extrFP`*extrFP*

---

**Description**

Calculate the pathway fingerprints

**Usage**

```
extrFP(drug_target, disease_biomarker, method = "enrich")

## S4 method for signature 'BasicData'
extrFP(drug_target, disease_biomarker, method = "enrich")

extrFP.BasicData(drug_target, disease_biomarker, method = "enrich")
```

**Arguments**

`drug_target`      A data frame or list of drug target.  
`disease_biomarker`  
                    A character of disease biomarkers or an order ranked geneList.  
`method`            one of "enrich" and "gsea"

**Value**

ScoreFP object  
ScoreFP object

**Author(s)**

Yuanlong Hu

**Examples**

```
data("drugSample")
drug_herb <- PrepareData(drugSample$drug_herb, col1 = "drug", col2 = "herb")
herb_target <- PrepareData(drugSample$herb_target,
                           col1 = "herb", col2 = "target",
                           format = "basket", sep = ", ")
drug_target <- CreateBasicData(drug_herb, herb_target)
FP <- extrFP(drug_target = drug_target,
             disease_biomarker = drugSample$disease_biomarker,
             method = "enrich")
```

---

 getDEG

*getDEG*


---

**Description**

Calculate differentially expressed genes using limma method.

**Usage**

```
getDEG(data, pdata, contrasts)
```

**Arguments**

data	A matrix of expression values where rows correspond to genes and columns correspond to samples.
pdata	A character vector of phenotype.
contrasts	character vector specifying contrasts

**Value**

A list

**Author(s)**

Yuanlong Hu

---

getF

*getF*


---

**Description**

Select features related to phenotype using Boruta

**Usage**

```
getF(expr, pdata, level = "gene", withTentative = TRUE, geneset)
```

**Arguments**

expr	A matrix of expression values where rows correspond to genes and columns correspond to samples.
pdata	A character of phenotype.
level	one of the gene or pathway
withTentative	If set to TRUE, Tentative attributes will be also returned.
geneset	A data frame of geneset containing two columns.

**Value**

A character of features.

**Author(s)**

Yuanlong Hu

---

<code>get_result</code>	<i>get_result</i>
-------------------------	-------------------

---

**Description**

Extract a table of the score result

**Usage**

```
get_result(result, pvalueCutoff = 0.05)
```

**Arguments**

`result` an object of class `ScoreResult`.  
`pvalueCutoff` p-value cutoff.

**Value**

a `data.frame`

**Author(s)**

Yuanlong Hu

**Examples**

```
data("drugResult")  
res <- drugResult$demoScoreFP  
res <- get_result(res)
```

head *Return the First Parts of a ScoreResult Object*

---

### Description

Return the First Parts of a ScoreResult Object

### Usage

```
## S4 method for signature 'ScoreResult'  
head(x, ...)
```

### Arguments

x	A ScoreResult object
...	other arguments

### Author(s)

Yuanlong Hu

---

imm\_centr *imm\_centr*

---

### Description

Computing the centrality of complex networks

### Usage

```
imm_centr(x, ...)  
  
## S4 method for signature 'data.frame'  
imm_centr(x)  
  
## S4 method for signature 'ScoreResultNet'  
imm_centr(x, drug, node = "target", net = "disease")  
  
imm_centr.data.frame(x, node, net)  
  
imm_centr.ScoreResultNet(x, drug, node, net)
```



**Arguments**

x	ScoreResultNet or data.frame object.
...	additional parameters
drug	drug name
node	Nodes that need to be evaluated. one of "disease" and "target."
net	Network. one of "disease" and "target."

**Value**

data.frame or ScoreResultNet object

**Author(s)**

Yuanlong Hu

Yuanlong Hu

Yuanlong Hu

---

overlap\_pathway      *overlap\_pathway*

---

**Description**

Performs set intersection on pathways fingerprints

**Usage**

```
overlap_pathway(FP, Drug)
```

**Arguments**

FP	A ScoreFP Object
Drug	The drug names

**Value**

a vector or data frame

**Author(s)**

Yuanlong Hu

---

plot_density	<i>plot_density</i>
--------------	---------------------

---

## Description

Plot smoothed density estimates for adjusted score

## Usage

```
plot_density(result, drug, fill = "#6495ED")
```

## Arguments

result	an object of class ScoreResult.
drug	a character of drug name.
fill	fill color.

## Value

a ggplot

## Author(s)

Yuanlong Hu

## Examples

```
## Not run:
data("drugSample")
FP <- extrFP(disease_biomarker = drugSample$disease_biomarker,
            drug_target = drugSample$herb_target,
            geneset = "ImmGenTop150")
res <- score_fp(FP, n=100)
plot_density(res, drug="BAN_XIA_XIE_XIN_TANG")

## End(Not run)
```

---

plot_network	<i>plot_network</i>
--------------	---------------------

---

## Description

Drug target or pathway network visualization

## Usage

```
plot_network(  
  x,  
  Drug,  
  node_color = c("lightblue", "orange", "red", "green"),  
  layout = "layout_nicely",  
  manipulation = FALSE,  
  ...  
)  
  
## S4 method for signature 'ScoreResultNet'  
plot_network(  
  x,  
  Drug,  
  node_color = c("lightblue", "orange", "red", "green"),  
  layout = "layout_nicely",  
  manipulation = FALSE,  
  node_type = "target",  
  background = "drug",  
  neighbor = FALSE  
)  
  
## S4 method for signature 'ScoreFP'  
plot_network(  
  x,  
  Drug,  
  node_type = "herb-compound-target",  
  node_color = c("lightblue", "orange", "red", "green"),  
  layout = "layout_nicely",  
  manipulation = FALSE,  
  highlight = NULL,  
  width = FALSE  
)  
  
## S4 method for signature 'ScoreResultFP'  
plot_network(  
  x,  
  Drug,  
  node_type = "herb-compound-target",
```

```

    node_color = c("lightblue", "orange", "red", "green"),
    layout = "layout_nicely",
    manipulation = FALSE,
    highlight = NULL,
    width = FALSE
)

plot_network.ScoreResultNet(
  x,
  Drug,
  node_color = c("lightblue", "orange", "red", "green"),
  layout = "layout_nicely",
  manipulation = FALSE,
  node_type = "target",
  background = "drug",
  neighbor = FALSE
)

plot_network.ScoreFP(
  x,
  Drug,
  node_type = "herb-compound-pathway",
  node_color = c("lightblue", "orange", "red", "green"),
  layout = "layout_nicely",
  manipulation = FALSE,
  highlight = NULL,
  width = FALSE
)

```

### Arguments

x	ScoreFP or ScoreResultNet object
Drug	The name of drug.
node_color	The node color.
layout	Character Name of network layout function to use. Default to "layout_nicely".
manipulation	Whether to edit the network.
...	additional parameters
node_type	network type. one of "herb-target", "herb-compound-target" and "target".
background	one of "drug" or "disease"
neighbor	logical.
highlight	A character vector of gene.
width	A logical. The number of overlapping genes between the two pathways is used as the width of the edges.

### Value

visNetwork object



---

 read\_gmt

*write\_gmt*


---

**Description**

parse gmt file to a data.frame

**Usage**

```
read_gmt(gmtfile, out_type = "data.frame")
```

**Arguments**

gmtfile	A GMT file name or URL containing gene sets.
out_type	A character vector of object name. one of "data.frame", "list", "GeneSetCollection"

**Value**

data.frame, list or GeneSetCollection

**Author(s)**

Yuanlong Hu

---

res\_rank

*res\_rank*


---

**Description**

Rank the results by rank aggregation methods

**Usage**

```
res_rank(..., method = "RRA")
```

**Arguments**

...	ScoreResult Object
method	rank aggregation method, by defaylt 'RRA', other options are 'min', 'geom.mean', 'mean', 'median' and 'stuart'

**Value**

a dataframe with two column

**Author(s)**

Yuanlong Hu

**References**

Kolde, R., Laur, S., Adler, P., & Vilo, J. (2012). Robust rank aggregation for gene list integration and meta-analysis. *Bioinformatics*, 28(4), 573-580.

---

ScoreFP-class

*Class "ScoreFP" This class represents the pathway fingerprint.*

---

**Description**

Class "ScoreFP" This class represents the pathway fingerprint.

**Slots**

Fingerprint pathway fingerprint

FPType pathway fingerprint type

Geneset Geneset name

**Author(s)**

Yuanlong Hu

---

ScoreResult-class

*Class "ScoreResult"*

---

**Description**

This class represents the result of score.

**Slots**

ScoreResult all score result.

adj distribution data

**Author(s)**

Yuanlong Hu

---

score_fp	<i>score_fp</i>
----------	-----------------

---

**Description**

Calculate the pathway fingerprint.similarity between disease and prescription.

**Usage**

```
score_fp(FP, n = 100, two_tailed = TRUE)
```

**Arguments**

FP	a ScoreFP object
n	The number of permutations.
two_tailed	whether returning a two-tailed p-value

**Value**

ScoreResult

**Author(s)**

Yuanlong Hu

**References**

Ye, H., Tang, K., Yang, L., Cao, Z., & Li, Y. (2012). Study of drug function based on similarity of pathway fingerprint. *Protein & cell*, 3(2), 132-139.

**Examples**

```
data("drugResult")
res <- score_fp(drugResult$demoFP, n=100)
res <- get_result(res)
```



---

score_network	<i>score_network</i>
---------------	----------------------

---

**Description**

Calculate the network score

**Usage**

```
score_network(Tar, DNet, n = 100, two_tailed = TRUE)
```

**Arguments**

Tar	A BasicData object containing drug target.
DNet	A data frame of disease network containing two columns.
n	The number of times random permutation sampling.
two_tailed	a logical: select a two-tailed p-value.

**Value**

ScoreResultNet object

**Author(s)**

Yuanlong Hu

**Examples**

```
data("drugSample")
drug_herb <- PrepareData(drugSample$drug_herb, col1 = "drug", col2 = "herb")
herb_target <- PrepareData(drugSample$herb_target,
                           col1 = "herb", col2 = "target",
                           format = "basket", sep = ", ")
drug_target <- CreateBasicData(drug_herb, herb_target)
res <- score_network(Tar = drug_target, DNet = drugSample$disease_network)
res <- get_result(res)
```

---

simFP	<i>simFP</i>
-------	--------------

---

**Description**

Calculate the similarity between Drug pathway Fingerprints

**Usage**

```
simFP(FP)
```

**Arguments**

FP                    A ScoreFP object

**Value**

a matrix

**Author(s)**

Yuanlong Hu

**Examples**

```
data("drugResult")
sim_mat <- simFP(drugResult$demoFP)
```

---

tail	<i>Return the last Parts of a ScoreResult Object</i>
------	--

---

**Description**

Return the last Parts of a ScoreResult Object

**Usage**

```
## S4 method for signature 'ScoreResult'
tail(x, ...)
```

**Arguments**

x                    A ScoreResult object  
...                   other arguments

**Author(s)**

Yuanlong Hu

---

*to\_df**to\_df*

---

**Description**

Convert list to data.frame

**Usage**`to_df(list)`**Arguments**`list` a list containing gene sets**Value**

data frame

**Author(s)**

Yuanlong Hu

---

*to\_list**to\_list*

---

**Description**

Create a new list from a data.frame of drug target and disease biomarker as input

**Usage**`to_list(dataframe, input = "single", sep = ", ")`**Arguments**`dataframe` a data frame of 2 column with term/drug and gene`input` one of the single or basket`sep` When 'input' is 'basket'.**Value**

list

**Author(s)**

Yuanlong Hu

---

viewpathway	<i>viewpathway</i>
-------------	--------------------

---

**Description**

View pathway information.

**Usage**

```
viewpathway(x, drug, view = "intersect")
```

**Arguments**

x	a ScoreFP object.
drug	character vector of drug name.
view	one of intersect, drug or disease.

**Value**

ScoreResult object

**Author(s)**

Yuanlong Hu

---

write_gmt	<i>write_gmt</i>
-----------	------------------

---

**Description**

prints data frame to a gmt file

**Usage**

```
write_gmt(geneset, gmt_file)
```

**Arguments**

geneset	A data.frame of 2 column with term/drug and gene
gmt_file	A character of gmt file name.

*write\_gmt*

21

**Value**

gmt file

**Author(s)**

Yuanlong Hu

# Index

- \* **datasets**
  - drugResult, 4
  - drugSample, 4
- as.data.frame, 3
- as.data.frame, ScoreResult-method (as.data.frame), 3
- BasicData-class, 3
- CreateBasicData, 4
- drugResult, 4
- drugSample, 4
- extrFP, 5
- extrFP, BasicData-method (extrFP), 5
- extrFP.BasicData (extrFP), 5
- get\_result, 7
- getDEG, 6
- getF, 6
- head, 8
- head, ScoreResult-method (head), 8
- imm\_centr, 8
- imm\_centr, data.frame-method (imm\_centr), 8
- imm\_centr, ScoreResultNet-method (imm\_centr), 8
- imm\_centr.data.frame (imm\_centr), 8
- imm\_centr.ScoreResultNet (imm\_centr), 8
- immcp-package, 2
- overlap\_pathway, 9
- plot\_density, 10
- plot\_network, 11
- plot\_network, ScoreFP-method (plot\_network), 11
- plot\_network, ScoreResultFP-method (plot\_network), 11
- plot\_network, ScoreResultNet-method (plot\_network), 11
- plot\_network.ScoreFP (plot\_network), 11
- plot\_network.ScoreResultNet (plot\_network), 11
- PrepareData, 13
- read\_gmt, 14
- res\_rank, 14
- score\_fp, 16
- score\_network, 17
- ScoreFP-class, 15
- ScoreResult-class, 15
- simFP, 18
- tail, 18
- tail, ScoreResult-method (tail), 18
- to\_df, 19
- to\_list, 19
- viewpathway, 20
- write\_gmt, 20