

# Package ‘cosmosR’

November 12, 2021

**Type** Package

**Title** COSMOS (Causal Oriented Search of Multi-Omic Space)

**Version** 1.3.0

**Description** COSMOS (Causal Oriented Search of Multi-Omic Space) is a method that integrates phosphoproteomics, transcriptomics, and metabolomics data sets based on prior knowledge of signaling, metabolic, and gene regulatory networks. It estimated the activities of transcription factors and kinases and finds a network-level causal reasoning. Thereby, COSMOS provides mechanistic hypotheses for experimental observations across multi-omics datasets.

**URL** <https://github.com/saezlab/COSMOSR>

**BugReports** <https://github.com/saezlab/COSMOSR/issues>

**Depends** R (>= 4.1)

**License** GPL-3

**Encoding** UTF-8

**LazyData** false

**RoxygenNote** 7.1.2

**VignetteBuilder** knitr

**Imports** AnnotationDbi, biomaRt, CARNIVAL, dorothea, dplyr, ggplot2, GSEABase, igraph, magrittr, org.Hs.eg.db, plyr, purrr, readr, rlang, scales, stringr, tibble, utils, visNetwork

**Suggests** testthat, knitr, rmarkdown, piano

**biocViews** CellBiology, Pathways, Network, Proteomics, Metabolomics, Transcriptomics, GeneSignaling

**git\_url** <https://git.bioconductor.org/packages/cosmosR>

**git\_branch** master

**git\_last\_commit** 0360169

**git\_last\_commit\_date** 2021-10-26

**Date/Publication** 2021-11-12

**Author** Aurélien Dugourd [aut] (<<https://orcid.org/0000-0002-0714-028X>>),  
 Attila Gabor [aut] (<<https://orcid.org/0000-0002-0776-1182>>),  
 Katharina Zirngibl [cre, aut] (<<https://orcid.org/0000-0002-7518-0339>>)

**Maintainer** Katharina Zirngibl <katharina.zirngibl@uni-heidelberg.de>

## R topics documented:

convert_ensembl_to_entrezid . . . . .	2
convert_genesymbols_to_entrezid . . . . .	3
cosmos_data . . . . .	4
default_CARNIVAL_options . . . . .	5
display_node_neighborhood . . . . .	5
extract_nodes_for_ORA . . . . .	7
format_COSMOS_res . . . . .	8
gmt_to_dataframe . . . . .	9
load_tf_regulon_dorothea . . . . .	9
load_tf_regulon_dorothea_omnipath . . . . .	10
metabolite_to_pubchem . . . . .	11
meta_network . . . . .	11
omnipath_ptm . . . . .	12
prepare_metabolomics_data . . . . .	13
preprocess_COSMOS_metabolism_to_signaling . . . . .	14
preprocess_COSMOS_signaling_to_metabolism . . . . .	16
print.cosmos_data . . . . .	18
run_COSMOS_metabolism_to_signaling . . . . .	19
run_COSMOS_signaling_to_metabolism . . . . .	20
toy_metabolic_input . . . . .	21
toy_network . . . . .	22
toy_RNA . . . . .	23
toy_signaling_input . . . . .	23
<b>Index</b>	<b>25</b>

---

convert\_ensembl\_to\_entrezid  
*convert gene ensembl to entrez id*

---

### Description

convert gene ensembl to entrez id

### Usage

convert\_ensembl\_to\_entrezid(ensembl)

**Arguments**

ensembl            vector of genes with ensembl id

**Value**

named vector, where names are the old ensemblIDs and values are the entrezIDs

**See Also**

[convert\\_genesymbols\\_to\\_entrezid](#)

**Examples**

```
ensembl <- c("ENSG00000100601", "ENSG00000178826", "ENSG00000138231")
entrez_map <- convert_ensembl_to_entrezid(ensembl)
```

---

*convert\_genesymbols\_to\_entrezid*  
*convert gene symbols to entrez id*

---

**Description**

convert gene symbols to entrez id

**Usage**

```
convert_genesymbols_to_entrezid(symbols)
```

**Arguments**

symbols            vector of genesymbols

**Value**

data.frame with human gene ENTREZID and SYMBOL mapping

**See Also**

[convert\\_ensembl\\_to\\_entrezid](#)

**Examples**

```
symbols <- c("MDH1", "PARP1", "IL6")
symbol_entrez_map <- convert_genesymbols_to_entrezid(symbols)
```

---

`cosmos_data`*Create Cosmos Data*

---

## Description

An S3 class that combines the required data into a comprehensive list. Use the [preprocess\\_COSMOS\\_signaling\\_to\\_metabolism](#) or [preprocess\\_COSMOS\\_metabolism\\_to\\_signaling](#) to create an instance.

## Usage

```
cosmos_data(  
  meta_network,  
  tf_regulon = NULL,  
  signaling_data,  
  metabolic_data,  
  expression_data,  
  verbose = TRUE  
)
```

## Arguments

<code>meta_network</code>	Prior knowledge network (PKN). By default COSMOS use a PKN derived from Omnipath, STITCHdb and Recon3D. See details on the data <a href="#">meta_network</a> .
<code>tf_regulon</code>	Collection of transcription factor - target interactions. A default collection from dorothea can be obtained by the <a href="#">load_tf_regulon_dorothea</a> function.
<code>signaling_data</code>	Numerical vector, where names are signaling nodes in the PKN and values are from {1, 0, -1}. Continuous data will be discretized using the <a href="#">sign</a> function.
<code>metabolic_data</code>	Numerical vector, where names are metabolic nodes in the PKN and values are continuous values that represents log <sub>2</sub> fold change or t-values from a differential analysis. These values are compared to the simulation results (simulated nodes can take value -1, 0 or 1).
<code>expression_data</code>	Numerical vector that represents the results of a differential gene expression analysis. Names are gene names using EntrezID starting with an X and values are log fold change or t-values. Genes with NA values are considered none expressed and they will be removed from the TF-gene expression interactions.
<code>verbose</code>	(default: TRUE) Reports details about the <a href="#">cosmos_data</a> object.

## Value

cosmos data class instance.

---

`default_CARNIVAL_options`*Setting Default CARNIVAL Options*

---

**Description**

Returns the default CARNIVAL options as a list. You can modify the elements of the list and then use it as an argument in “run\_COSMOS()”.

**Usage**`default_CARNIVAL_options()`**Value**

returns a list with all possible options implemented in CARNIVAL. see the documentation on [runCARNIVAL](#).

**Examples**

```
# load and change default options:
my_options = default_CARNIVAL_options()

my_options$solverPath = "/Applications/CPLEX_Studio128/cplex/bin/x86-64_osx/cplex"
my_options$threads = 2
my_options$timelimit = 3600*15
```

---

`display_node_neighborhood`*display\_node\_neighborhood*

---

**Description**

display input and measurements within n steps of a given set of nodes

**Usage**`display_node_neighborhood(central_node, sif, att, n = 100)`

**Arguments**

central_node	character or character vector; node ID(s) around which a network will be branched out until measurements and input are reached
sif	df; COSMOS network solution in sif format like the first list element returned by the format_cosmos_res function
att	df; attributes of the nodes of the COSMOS network solution like the second list element returned by the format_cosmos_res function
n	numeric; maximum number of steps in the network to look for inputs and measurements

**Value**

a visnetwork object

**Examples**

```

CARNIVAL_options <- cosmosR::default_CARNIVAL_options()
CARNIVAL_options$solver <- "lpSolve"
data(toy_network)
data(toy_signaling_input)
data(toy_metabolic_input)
data(toy_RNA)
test_for <- preprocess_COSMOS_signaling_to_metabolism(meta_network = toy_network,
signaling_data = toy_signaling_input,
metabolic_data = toy_metabolic_input,
diff_expression_data = toy_RNA,
maximum_network_depth = 15,
remove_unexpressed_nodes = TRUE,
CARNIVAL_options = CARNIVAL_options
)
test_result_for <- run_COSMOS_signaling_to_metabolism(data = test_for,
CARNIVAL_options = CARNIVAL_options)
data(metabolite_to_pubchem)
data(omnipath_ptm)
test_result_for <- format_COSMOS_res(test_result_for,
metab_mapping = metabolite_to_pubchem,
measured_nodes = unique(c(names(toy_metabolic_input),
names(toy_signaling_input))),
omnipath_ptm = omnipath_ptm)
network_plot <- display_node_neighborhood(central_node = 'NFKB1',
sif = test_result_for[[1]],
att = test_result_for[[2]],
n = 7)
network_plot

```

---

extract\_nodes\_for\_ORA *Extract COSMOS nodes for ORA analysis*

---

## Description

Function to extract the nodes that appear in the COSMOS output network and the background genes (all genes present in the prior knowledge network)

## Usage

```
extract_nodes_for_ORA(sif, att)
```

## Arguments

sif	df; COSMOS network solution in sif format like the first list element returned by the format_cosmos_res function
att	df; attributes of the nodes of the COMSOS network solution like the second list element returned by the format_cosmos_res function

## Value

List with 2 objects: the success and the background genes

## Examples

```
CARNIVAL_options <- cosmosR::default_CARNIVAL_options()
CARNIVAL_options$solver <- "lpSolve"
data(toy_network)
data(toy_signaling_input)
data(toy_metabolic_input)
data(toy_RNA)
test_for <- preprocess_COSMOS_signaling_to_metabolism(meta_network = toy_network,
signaling_data = toy_signaling_input,
metabolic_data = toy_metabolic_input,
diff_expression_data = toy_RNA,
maximum_network_depth = 15,
remove_unexpressed_nodes = TRUE,
CARNIVAL_options = CARNIVAL_options
)
test_result_for <- run_COSMOS_signaling_to_metabolism(data = test_for,
CARNIVAL_options = CARNIVAL_options)
data(metabolite_to_pubchem)
data(omnipath_ptm)
test_result_for <- format_COSMOS_res(test_result_for,
metab_mapping = metabolite_to_pubchem,
measured_nodes = unique(c(names(toy_metabolic_input),
names(toy_signaling_input))),
omnipath_ptm = omnipath_ptm)
extrected_nodes <- extract_nodes_for_ORA(
```

```
sif = test_result_for[[1]],
att = test_result_for[[2]]
)
```

---

```
format_COSMOS_res      format_COSMOS_res
```

---

## Description

formats the network with readable names

## Usage

```
format_COSMOS_res(
  cosmos_res,
  metab_mapping,
  gene_mapping = "org.Hs.eg.db",
  measured_nodes,
  omnipath_ptm
)
```

## Arguments

cosmos_res	results of CARNIVAL run
metab_mapping	a named vector with pubchem cid as names and desired metabolite names as values.
gene_mapping	by default, use the 'org.Hs.eg.db' to map gene names. Can also be a named vector with entrez gene id as names and desired gene names as values.
measured_nodes	vector of nodes that are measured or inputs
omnipath_ptm	ptms database from OmnipathR

## Value

list with network and attribute tables.

## Examples

```
CARNIVAL_options <- cosmosR::default_CARNIVAL_options()
CARNIVAL_options$solver <- "lpSolve"
data(toy_network)
data(toy_signaling_input)
data(toy_metabolic_input)
data(toy_RNA)
test_for <- preprocess_COSMOS_signaling_to_metabolism(meta_network = toy_network,
signaling_data = toy_signaling_input,
metabolic_data = toy_metabolic_input,
diff_expression_data = toy_RNA,
```



```
maximum_network_depth = 15,  
remove_unexpressed_nodes = TRUE,  
CARNIVAL_options = CARNIVAL_options  
)  
test_result_for <- run_COSMOS_signaling_to_metabolism(data = test_for,  
CARNIVAL_options = CARNIVAL_options)  
data(metabolite_to_pubchem)  
data(omnipath_ptm)  
test_result_for <- format_COSMOS_res(test_result_for,  
metab_mapping = metabolite_to_pubchem,  
measured_nodes = unique(c(names(toy_metabolic_input),  
names(toy_signaling_input))),  
omnipath_ptm = omnipath_ptm)
```

---

gmt\_to\_dataframe      *Convert gmt file to data frame*

---

### Description

This function is designed to convert a gmt file (gene set file from MSigDB) into a two column data frame where the first column corresponds to omic features (genes) and the second column to associated terms (pathway the gene belongs to). One gene can belong to several pathways.

### Usage

```
gmt_to_dataframe(gmtfile)
```

### Arguments

gmtfile              a full path name of the gmt file to be converted

### Value

a two column data frame where the first column corresponds to omic features and the second column to associated terms (pathways).

---

load\_tf\_regulon\_dorothea  
*load transcription factor regulon*

---

### Description

load the transcription factors from DOROTHEA package and converts gene symbols to EntrezID using org.Hs.eg.db

**Usage**

```
load_tf_regulon_dorothea(toEntrez = TRUE, confidence = c("A", "B", "C"))
```

**Arguments**

`toEntrez` if TRUE (default), converts gene symbols to EntrezID

`confidence` strong vector (by default: `c("A","B","C")`). Subset of {A, B, C, D, E}. See the ‘dorothea’ for the meaning of confidence levels. `package` for further details.

**Value**

returns a PKN of a form of a data table. Each row is an interaction. Columns names are:

- ‘tf’ transcription factor
- ‘confidence’ class of confidence
- ‘target’ target gene
- ‘sign’ indicates if interaction is up (1) or down-regulation (-1).

**Examples**

```
load_tf_regulon_dorothea()
```

---

```
load_tf_regulon_dorothea_omnipath  
load_tf_regulon_dorothea_omnipath
```

---

**Description**

downloads the TF-regulons from Omnipaht. Different from DOROTHEA regulon because sign is handled differently: if both stimulation and inhibition was reported then it is removed.

**Usage**

```
load_tf_regulon_dorothea_omnipath()
```

**Value**

TF - TF-target interactions in tibble format.

**Examples**

```
load_tf_regulon_dorothea_omnipath()
```

---

metabolite\_to\_pubchem *Metabolite-PubChem CID Mapping*

---

**Description**

Mapping between metabolite names and PubChem CIDs obtained from the recon3D metabolic model. Combined table version from the recon3D matlab object.

**Usage**

```
data(metabolite_to_pubchem)
```

**Format**

An object of class “data.frame” with 1131 rows (metabolites) and two variables:

name Metabolite name synonym

pubchem Pubchem CID

**Source**

<https://www.vmh.life/#downloadview>, downloaded on Feb 19th, 2018.

**References**

Brunk, E. et al. (2018) *Nature Biotechnology*. **36**(3), 272–281.

**Examples**

```
data(metabolite_to_pubchem)
```

---

meta\_network *Meta Prior Knowledge Network*

---

**Description**

Comprehensive Prior Knowledge Network (PKN), which combines signaling and metabolic interaction networks. The network was constructed using the Recon3D and STITCH metabolic networks as well as the signaling network from OmniPath.

**Usage**

```
data(meta_network)
```

**Format**

An object of class “tibble” with 117065 rows (interactions) and three variables:

source Source node, either metabolite or protein

interaction Type of interaction, 1 = Activation, -1 = Inhibition

target Target node, either metabolite or protein A detailed description of the identifier formatting can be found under [https://metapkn.omnipathdb.org/00\\_\\_README.txt](https://metapkn.omnipathdb.org/00__README.txt).

**Source**

The network is available in Omnipath: [https://metapkn.omnipathdb.org/metapkn\\_\\_20200122.txt](https://metapkn.omnipathdb.org/metapkn__20200122.txt), the scripts used for the build of the network are available under [https://github.com/saezlab/Meta\\_PKN](https://github.com/saezlab/Meta_PKN).

**References**

Dugourd, A., Kuppe, C. and Sciacovelli, M. et. al. (2021) *Molecular Systems Biology*. **17**, e9730.

**Examples**

```
data(meta_network)
```

---

omnipath\_ptm

*OmniPath PTMs*

---

**Description**

Collection of annotated enzyme-substrate post translational modifications obtained from OmniPath.

**Usage**

```
data(omnipath_ptm)
```

**Format**

An object of class “data.frame” with 39201 rows (PTMs) and 12 variables:

enzyme

substrate

enzyme\_genesymbol

substrate\_genesymbol

residue\_type

residue\_offset

modification

sources  
references  
curation\_effort  
n\_references  
n\_resources

### Source

Default resource collection of OmniPath: <http://omnipathdb.org/ptms?fields=sources,references&genesymbols=1>, version of Feb 5th, 2020.

### References

Turei, D., Korcsmaros, T. and Saez-Rodriguez, J. (2016) *Nature Methods*. **13**(12), 966–967.

### Examples

```
data(omnipath_ptm)
```

---

```
prepare_metabolomics_data  
    format COSMOS metabolic input
```

---

### Description

This function prepares the metabolic data to be used in the COSMOS optimization steps. It takes as input a vector with the metabolic data (e.g. limma t values) named with PUBCHEM IDs and expand it to the multi-compartment COSMOS format. It also messages the number of final inputs in the meta network.

### Usage

```
prepare_metabolomics_data(metabolic_data, meta_network)
```

### Arguments

`metabolic_data` A named numeric vector, containing the values to be used for the metabolic layer in COSMOS. The names of the vector should be PUBCHEM IDs.  
`meta_network` Prior knowledge network created with `data(meta_network)`.

### Value

A new vector ready to be used as COSMOS input.

**Examples**

```
# generate random t-values:
t_values <- rnorm(10)
# assign to metabolites with pubchem names
data(metabolite_to_pubchem)
names(t_values) <- names(metabolite_to_pubchem)[1:10]

data(meta_network)
prepare_metabolomics_data(t_values, meta_network)
```

---

```
preprocess_COSMOS_metabolism_to_signaling
```

*Preprocess COSMOS Inputs For Metabolism to Signaling*

---

**Description**

Runs checks on the input data and simplifies the prior knowledge network. Simplification includes the removal of (1) nodes that are not reachable from signaling nodes and (2) interactions between transcription factors and target genes if the target gene does not respond or the response is contradictory with the change in the transcription factor activity. Optionally, further TF activities are estimated via network optimization via CARNIVAL and the interactions between TF and genes are filtered again.

**Usage**

```
preprocess_COSMOS_metabolism_to_signaling(
  meta_network = meta_network,
  tf_regulon = load_tf_regulon_dorothea(),
  signaling_data,
  metabolic_data,
  diff_expression_data,
  diff_exp_threshold = 1,
  maximum_network_depth = 8,
  expressed_genes = names(diff_expression_data)[!is.na(diff_expression_data)],
  remove_unexpressed_nodes = TRUE,
  filter_tf_gene_interaction_by_optimization = TRUE,
  CARNIVAL_options = default_CARNIVAL_options()
)
```

**Arguments**

meta_network	prior knowledge network (PKN). A PKN released with COSMOS and derived from Omnipath, STITCHdb and Recon3D can be used. See details on the data <a href="#">meta_network</a> .
tf_regulon	collection of transcription factor - target interactions. A default collection from dorothea can be obtained by the <a href="#">load_tf_regulon_dorothea</a> function.

signaling_data	numerical vector, where names are signaling nodes in the PKN and values are from {1, 0, -1}. Continuous data will be discretized using the <a href="#">sign</a> function.
metabolic_data	numerical vector, where names are metabolic nodes in the PKN and values are continuous values that represents log2 fold change or t-values from a differential analysis. These values are compared to the simulation results (simulated nodes can take value -1, 0 or 1)
diff_expression_data	(optional) numerical vector that represents the results of a differential gene expression analysis. Names are gene names using EntrezID starting with an X and values are log fold change or t-values. <a href="#">convert_genesymbols_to_entrezid</a> can be used for conversion. We use the “diff_exp_threshold” parameter to decide which genes changed significantly. Genes with NA values are considered none expressed and they will be removed from the TF-gene expression interactions.
diff_exp_threshold	threshold parameter (default 1) used to binarize the values of “diff_expression_data”.
maximum_network_depth	integer > 0 (default: 8). Nodes that are further than “maximum_network_depth” steps from the signaling nodes on the directed graph of the PKN are considered non-reachable and are removed.
expressed_genes	character vector. Names of nodes that are expressed. By default we consider all the nodes that appear in diff_expression_data with a numeric value (i.e. nodes with NA are removed)
remove_unexpressed_nodes	if TRUE (default) removes nodes from the PKN that are not expressed, see input “expressed_genes”.
filter_tf_gene_interaction_by_optimization	(default:TRUE), if TRUE then runs a network optimization that estimates TF activity not included in the inputs and checks the consistency between the estimated activity and change in gene expression. Removes interactions where TF and gene expression are inconsistent
CARNIVAL_options	list that controls the options of CARNIVAL. See details in <a href="#">default_CARNIVAL_options</a> .

## Value

cosmos\_data object with the following fields:

- meta\_network Filtered PKN
- tf\_regulon TF - target regulatory network
- signaling\_data\_bin Binarised signaling data
- metabolic\_data Metabolomics data
- diff\_expression\_data\_bin Binarized gene expression data
- optimized\_network Initial optimized network if filter\_tf\_gene\_interaction\_by\_optimization is TRUE

**See Also**

[meta\\_network](#) for meta PKN, [load\\_tf\\_regulon\\_dorothea](#) for tf regulon, [convert\\_genesymbols\\_to\\_entrezid](#) for gene conversion, [runCARNIVAL](#).

**Examples**

```
CARNIVAL_options <- cosmosR::default_CARNIVAL_options()
CARNIVAL_options$solver <- "lpSolve"
data(toy_network)
data(toy_signaling_input)
data(toy_metabolic_input)
data(toy_RNA)
test_back <- preprocess_COSMOS_metabolism_to_signaling(meta_network = toy_network,
signaling_data = toy_signaling_input,
metabolic_data = toy_metabolic_input,
diff_expression_data = toy_RNA,
maximum_network_depth = 15,
remove_unexpressed_nodes = TRUE,
CARNIVAL_options = CARNIVAL_options
)
```

---

```
preprocess_COSMOS_signaling_to_metabolism
```

*Preprocess COSMOS Inputs For Signaling to Metabolism*

---

**Description**

Runs checks on the input data and simplifies the prior knowledge network. Simplification includes the removal of (1) nodes that are not reachable from signaling nodes and (2) interactions between transcription factors and target genes if the target gene does not respond or the response is contradictory with the change in the transcription factor activity. Optionally, further TF activities are estimated via network optimization via CARNIVAL and the interactions between TF and genes are filtered again.

**Usage**

```
preprocess_COSMOS_signaling_to_metabolism(
  meta_network = meta_network,
  tf_regulon = load_tf_regulon_dorothea(),
  signaling_data,
  metabolic_data,
  diff_expression_data,
  diff_exp_threshold = 1,
  maximum_network_depth = 8,
  expressed_genes = names(diff_expression_data)[!is.na(diff_expression_data)],
  remove_unexpressed_nodes = TRUE,
  filter_tf_gene_interaction_by_optimization = TRUE,
  CARNIVAL_options = default_CARNIVAL_options()
)
```



**Arguments**

<code>meta_network</code>	prior knowledge network (PKN). A PKN released with COSMOS and derived from Omnipath, STITCHdb and Recon3D can be used. See details on the data <a href="#">meta_network</a> .
<code>tf_regulon</code>	collection of transcription factor - target interactions. A default collection from dorothea can be obtained by the <code>load_tf_regulon_dorothea</code> function.
<code>signaling_data</code>	numerical vector, where names are signaling nodes in the PKN and values are from {1, 0, -1}. Continuous data will be discretized using the <code>sign</code> function.
<code>metabolic_data</code>	numerical vector, where names are metabolic nodes in the PKN and values are continuous values that represents log2 fold change or t-values from a differential analysis. These values are compared to the simulation results (simulated nodes can take value -1, 0 or 1)
<code>diff_expression_data</code>	(optional) numerical vector that represents the results of a differential gene expression analysis. Names are gene names using EntrezID starting with an X and values are log fold change or t-values. <code>convert_genesymbols_to_entrezid</code> can be used for conversion. We use the “diff_exp_threshold” parameter to decide which genes changed significantly. Genes with NA values are considered none expressed and they will be removed from the TF-gene expression interactions.
<code>diff_exp_threshold</code>	threshold parameter (default 1) used to binarize the values of “diff_expression_data”.
<code>maximum_network_depth</code>	integer > 0 (default: 8). Nodes that are further than “maximum_network_depth” steps from the signaling nodes on the directed graph of the PKN are considered non-reachable and are removed.
<code>expressed_genes</code>	character vector. Names of nodes that are expressed. By default we consider all the nodes that appear in <code>diff_expression_data</code> with a numeric value (i.e. nodes with NA are removed)
<code>remove_unexpressed_nodes</code>	if TRUE (default) removes nodes from the PKN that are not expressed, see input “expressed_genes”.
<code>filter_tf_gene_interaction_by_optimization</code>	(default:TRUE), if TRUE then runs a network optimization that estimates TF activity not included in the inputs and checks the consistency between the estimated activity and change in gene expression. Removes interactions where TF and gene expression are inconsistent
<code>CARNIVAL_options</code>	list that controls the options of CARNIVAL. See details in <a href="#">default_CARNIVAL_options</a> .

**Value**

`cosmos_data` object with the following fields:

`meta_network` Filtered PKN

tf\_regulon TF - target regulatory network  
 signaling\_data\_bin Binarised signaling data  
 metabolic\_data Metabolomics data  
 diff\_expression\_data\_bin Binarized gene expression data  
 optimized\_network Initial optimized network if filter\_tf\_gene\_interaction\_by\_optimization is TRUE

### See Also

[meta\\_network](#) for meta PKN, [load\\_tf\\_regulon\\_dorothea](#) for tf regulon, [convert\\_genesymbols\\_to\\_entrezid](#) for gene conversion, [runCARNIVAL](#).

### Examples

```
CARNIVAL_options <- cosmosR::default_CARNIVAL_options()
CARNIVAL_options$solver <- "lpSolve"
data(toy_network)
data(toy_signaling_input)
data(toy_metabolic_input)
data(toy_RNA)
test_for <- preprocess_COSMOS_signaling_to_metabolism(meta_network = toy_network,
signaling_data = toy_signaling_input,
metabolic_data = toy_metabolic_input,
diff_expression_data = toy_RNA,
maximum_network_depth = 15,
remove_unexpressed_nodes = TRUE,
CARNIVAL_options = CARNIVAL_options
)
```

---

print.cosmos\_data      *Print Cosmos Data Summary Print a summary of cosmos data.*

---

### Description

Print Cosmos Data Summary Print a summary of cosmos data.

### Usage

```
## S3 method for class 'cosmos_data'
print(x, ...)
```

### Arguments

x                    [cosmos\\_data](#) object. Use the [preprocess\\_COSMOS\\_signaling\\_to\\_metabolism](#) or [preprocess\\_COSMOS\\_metabolism\\_to\\_signaling](#) functions to create one.  
 ...                  Further print arguments passed to or from other methods.

**Value**

input (invisible)

**See Also**

[print](#), [cosmos\\_data](#)

---

```
run_COSMOS_metabolism_to_signaling
    run COSMOS metabolism to signaling
```

---

**Description**

Runs COSMOS from metabolism to signaling. This function uses CARNIVAL to find a subset of the prior knowledge network based on optimization that (1) includes the most measured and input nodes and (2) which is in agreement with the data. Use [preprocess\\_COSMOS\\_metabolism\\_to\\_signaling](#) to prepare the the inputs, measurements and the prior knowledge network.

**Usage**

```
run_COSMOS_metabolism_to_signaling(
    data,
    CARNIVAL_options = default_CARNIVAL_options()
)
```

**Arguments**

`data` [cosmos\\_data](#) object. Use the [preprocess\\_COSMOS\\_metabolism\\_to\\_signaling](#) function to create an instance.

`CARNIVAL_options` List that controls the options of CARNIVAL. See details in [default\\_CARNIVAL\\_options](#).

**Value**

List with the following elements:

`weightedSIF` The averaged networks found by optimization in a format of a Simple Interaction network, i.e. each row codes an edge

`N_networks` Number of solutions found by the optimization

`nodesAttributes` Estimated node properties

`individual_networks` List of optimal networks found

`individual_networks_node_attributes` Node activity in each network

**See Also**

[preprocess\\_COSMOS\\_metabolism\\_to\\_signaling](#), [runCARNIVAL](#), [cosmos\\_data](#)

**Examples**

```

CARNIVAL_options <- cosmosR::default_CARNIVAL_options()
CARNIVAL_options$solver <- "lpSolve"
data(toy_network)
data(toy_signaling_input)
data(toy_metabolic_input)
data(toy_RNA)
test_back <- preprocess_COSMOS_metabolism_to_signaling(meta_network = toy_network,
signaling_data = toy_signaling_input,
metabolic_data = toy_metabolic_input,
diff_expression_data = toy_RNA,
maximum_network_depth = 15,
remove_unexpressed_nodes = TRUE,
CARNIVAL_options = CARNIVAL_options
)
test_result_back <- run_COSMOS_metabolism_to_signaling(data = test_back,
CARNIVAL_options = CARNIVAL_options)

```

---

```

run_COSMOS_signaling_to_metabolism
      run COSMOS signaling to metabolism

```

---

**Description**

Runs COSMOS from signaling to metabolism. This function uses CARNIVAL to find a subset of the prior knowledge network based on optimisation that (1) includes the most measured and input nodes and (2) which is in agreement with the data. Use [preprocess\\_COSMOS\\_signaling\\_to\\_metabolism](#) to prepare inputs, measurements and prior knowledge network.

**Usage**

```

run_COSMOS_signaling_to_metabolism(
  data,
  CARNIVAL_options = default_CARNIVAL_options()
)

```

**Arguments**

**data** [cosmos\\_data](#) object. Use the [preprocess\\_COSMOS\\_signaling\\_to\\_metabolism](#) function to create an instance.

**CARNIVAL\_options** List that controls the options of CARNIVAL. See details in [default\\_CARNIVAL\\_options](#).

**Value**

List with the following elements:

**weightedSIF** The averaged networks found by optimization in a format of a Simple Interaction network, i.e. each row codes an edge

N\_networks Number of solutions found by the optimization  
nodesAttributes Estimated node properties  
individual\_networks List of optimal networks found  
individual\_networks\_node\_attributes Node activity in each network

### See Also

[preprocess\\_COSMOS\\_metabolism\\_to\\_signaling](#), [runCARNIVAL](#), [cosmos\\_data](#)

### Examples

```
CARNIVAL_options <- cosmosR::default_CARNIVAL_options()
CARNIVAL_options$solver <- "lpSolve"
data(toy_network)
data(toy_signaling_input)
data(toy_metabolic_input)
data(toy_RNA)
test_for <- preprocess_COSMOS_signaling_to_metabolism(meta_network = toy_network,
signaling_data = toy_signaling_input,
metabolic_data = toy_metabolic_input,
diff_expression_data = toy_RNA,
maximum_network_depth = 15,
remove_unexpressed_nodes = TRUE,
CARNIVAL_options = CARNIVAL_options
)
test_result_for <- run_COSMOS_signaling_to_metabolism(data = test_for,
CARNIVAL_options = CARNIVAL_options)
```

---

toy\_metabolic\_input    *Toy Metabolic Input Data*

---

### Description

This metabolic data are a subset from the metabolic measurements used as an input in the case study of the COSMOS paper. The subset contains a random selection of metabolites present in the toy network.

### Usage

```
data(toy_metabolic_input)
```

### Format

An object of class “numeric” containing the t-values of 3 metabolites, which are named with metabolite PubChem CIDs matching the toy network.

### Source

Subset of: [https://github.com/saezlab/COSMOS\\_MSB/blob/main/data/metab\\_input\\_COSMOS.csv](https://github.com/saezlab/COSMOS_MSB/blob/main/data/metab_input_COSMOS.csv)

### References

Dugourd, A., Kuppe, C. and Sciacovelli, M. et. al. (2021) *Molecular Systems Biology*. **17**, e9730.

### Examples

```
data(toy_metabolic_input)
```

---

toy\_network

*Toy Input Network*

---

### Description

This signaling network is the reduced COSMOS network solution obtained in the case study of the COSMOS paper. Here, this network solution is reused as an exemplary input prior knowledge network (PKN).

### Usage

```
data(toy_network)
```

### Format

An object of class “data.frame” with 19 rows (interactions) and three variables:

source Source node, either metabolite or protein

interaction Type of interaction, 1 = Activation, -1 = Inhibition

target Target node, either metabolite or protein A detailed description of the identifier formatting can be found under [https://metapkn.omnipathdb.org/00\\_\\_README.txt](https://metapkn.omnipathdb.org/00__README.txt).

### Source

The network data are available on github: [https://github.com/saezlab/COSMOS\\_MSB/tree/main/results/COSMOS\\_result/COSMOS\\_res\\_session.RData](https://github.com/saezlab/COSMOS_MSB/tree/main/results/COSMOS_result/COSMOS_res_session.RData). The toy\_network is the combined network of the COSMOS network solutions CARNIVAL\_Result2 and CARNIVAL\_Result\_rerun subsequently reduced to 19 exemplary nodes.

### References

Dugourd, A., Kuppe, C. and Sciacovelli, M. et. al. (2021) *Molecular Systems Biology*. **17**, e9730.

**Examples**

```
data(toy_network)
```

---

toy\_RNA

*Toy Input Transcription Data Set*

---

**Description**

This exemplary transcription data are the differential expression results analysed in the case study of the COSMOS paper.

**Usage**

```
data(toy_RNA)
```

**Format**

An object of class “numeric” containing the t-values of 15919 genes, which are named with gene Entrez IDs matching the toy network.

**Source**

[https://github.com/saezlab/COSMOS\\_MSB/blob/main/data/RNA\\_ttop\\_tumorvshealthy.csv](https://github.com/saezlab/COSMOS_MSB/blob/main/data/RNA_ttop_tumorvshealthy.csv)

**References**

Dugourd, A., Kuppe, C. and Sciacovelli, M. et. al. (2021) *Molecular Systems Biology*. **17**, e9730.

**Examples**

```
data(toy_RNA)
```

---

toy\_signaling\_input

*Toy Signaling Input*

---

**Description**

This signaling data are a subset of the footprint-based signaling activity estimates of transcription factors, kinases and phosphatases used as an input in the case study of the COSMOS paper. The subset contains a random selection of signaling proteins present in the toy network.

**Usage**

```
data(toy_signaling_input)
```

**Format**

An object of class “data.frame” containing the normalised enrichment scores (NES) of 2 signaling proteins, which are named with their respective gene Entrez ID matching the toy network.

**Source**

Subset of: [https://github.com/saezlab/COSMOS\\_MSB/blob/main/data/signaling\\_input\\_COSMOS.csv](https://github.com/saezlab/COSMOS_MSB/blob/main/data/signaling_input_COSMOS.csv)

**References**

Dugourd, A., Kuppe, C. and Sciacovelli, M. et. al. (2021) *Molecular Systems Biology*. **17**, e9730.

**Examples**

```
data(toy_signaling_input)
```



# Index

## \* datasets

- meta\_network, 11
  - metabolite\_to\_pubchem, 11
  - omnipath\_ptm, 12
  - toy\_metabolic\_input, 21
  - toy\_network, 22
  - toy\_RNA, 23
  - toy\_signaling\_input, 23
- convert\_ensembl\_to\_entrezid, 2, 3
- convert\_genesymbols\_to\_entrezid, 3, 3,  
15–18
- cosmos\_data, 4, 4, 18–21
- default\_CARNIVAL\_options, 5, 15, 17, 19,  
20
- display\_node\_neighborhood, 5
- extract\_nodes\_for\_ORA, 7
- format\_COSMOS\_res, 8
- gmt\_to\_dataframe, 9
- load\_tf\_regulon\_dorothea, 4, 9, 14, 16–18
- load\_tf\_regulon\_dorothea\_omnipath, 10
- meta\_network, 4, 11, 14, 16–18
- metabolite\_to\_pubchem, 11
- omnipath\_ptm, 12
- prepare\_metabolomics\_data, 13
- preprocess\_COSMOS\_metabolism\_to\_signaling,  
4, 14, 18, 19, 21
- preprocess\_COSMOS\_signaling\_to\_metabolism,  
4, 16, 18, 20
- print, 19
- print.cosmos\_data, 18
- run\_COSMOS\_metabolism\_to\_signaling, 19
- run\_COSMOS\_signaling\_to\_metabolism, 20
- runCARNIVAL, 5, 16, 18, 19, 21
- sign, 4, 15, 17
- toy\_metabolic\_input, 21
- toy\_network, 22
- toy\_RNA, 23
- toy\_signaling\_input, 23