Package ‘OmnipathR’

March 30, 2021

Type Package

Title OmniPath web service client

Version 2.0.0

Description A client for the OmniPath web service (https://www.omnipathdb.org). It also includes functions to transform and pretty print some of the downloaded data.

License MIT + file LICENSE

URL `https://saezlab.github.io/OmnipathR/`

BugReports `https://github.com/saezlab/OmnipathR/issues`

biocViews GraphAndNetwork, Network, Pathways, Software, ThirdPartyClient, DataImport, DataRepresentation, GeneSignaling, GeneRegulation, SystemsBiology

Encoding UTF-8

VignetteBuilder knitr

Depends R(>= 4.0), igraph, graphics, methods, utils, jsonlite

Imports dplyr, stats, rlang, tidyr

Suggests dnet, gprofiler2, BiocStyle, testthat, knitr, rmarkdown, ggplot2, ggraph

RoxygenNote 7.1.1

git_url https://git.bioconductor.org/packages/OmnipathR

git_branch RELEASE_3_12

git_last_commit 2c98139

git_last_commit_date 2020-10-27

Date/Publication 2021-03-29

Author Alberto Valdeolivas [aut] (<https://orcid.org/0000-0001-5482-9023>),
Denes Turei [cre, aut] (<https://orcid.org/0000-0002-7249-9379>),
Attila Gabor [aut]

Maintainer Denes Turei <`turei.denes@gmail.com`>

R topics documented:

`.omnipath_options_defaults`

Default values for the package options

Description

These options describe the default settings for OmnipathR so you do not need to pass these parameters at each function call. Currently the only option useful for the public web service at omnipathdb.org is “omnipath.license“. If you are a for-profit user set it to “'commercial'“ to make sure all the data you download from OmniPath is legally allowed for commercial use. Otherwise just leave it as it is: “'academic'“. If you don’t use omnipathdb.org but within your organization you deployed your own pypath server and want to share data whith a limited availability to outside users, you may want to use a password. For this you can use the “omnipath.password“ option. Also if you want the R package to work from another pypath server instead of omnipathdb.org, you can change the option “omnipath.url“.

Usage

```
.omnipath_options_defaults
```

Format

An object of class `list` of length 4.

`get_annotation_resources` *Retrieves a list of available resources in the annotations database of OmniPath*

Description

Get the names of the resources from `https://omnipath.org/annotations`.

Usage

```
get_annotation_resources(dataset = NULL, ...)

get_annotation_databases(...)
```

Arguments

`dataset`	ignored for this query type
`...`	Passed to `get_annotation_resources`.

Value

character vector with the names of the annotation resources

See Also

`get_resources`, `import_omnipath_annotations`

Examples

```
get_annotation_resources()
```

`get_complex_genes` *Get all the molecular complexes for a given gene(s)*

Description

This function returns all the molecular complexes where an input set of genes participate. User can choose to retrieve every complex where any of the input genes participate or just retrieve these complexes where all the genes in input set participate together.

Usage

```
get_complex_genes(
  complexes = import_Omnipath_complexes(),
  select_genes,
  total_match = FALSE
)
```

Arguments

`complexes`	complexes data frame (obtained using `import_omnipath_complexes`)
`select_genes`	vector containing the genes for whom complexes will be retrieved (hgnc format).
`total_match`	[default=FALSE] logical indicating if the user wants to get all the complexes where any of the input genes participate (FALSE) or to get only the complexes where all the input genes participate together (TRUE)

Value

data.frame of complexes

See Also

`import_omnipath_complexes`)

Examples

```
complexes <- import_omnipath_complexes(
    filter_databases = c("CORUM", "hu.MAP")
)
query_genes = c("LMNA","BANF1")
complexes_query_genes = get_complex_genes(complexes,query_genes)
```

`get_complex_resources` *Retrieve a list of complex resources available in Omnipath*

Description

get the names of the resources from `https://omnipath.org/complexes`

Usage

```
get_complex_resources(dataset = NULL)

get_complexes_databases(...)
```

Arguments

`dataset`	ignored for this query type
`...`	Passed to `import_omnipath_enzsub`.

Value

character vector with the names of the databases

See Also

`get_resources`,`import_omnipath_complexes`

Examples

```
get_complex_resources()
```

`get_enzsub_resources` *Retrieves a list of enzyme-substrate resources available in OmniPath*

Description

Get the names of the enzyme-substrate relationship resources available in `https://omnipath.org/enzsub`

Usage

```
get_enzsub_resources(dataset = NULL)

get_ptms_databases(...)
```

Arguments

`dataset`	ignored for this query type
`...`	Passed to `get_enzsub_resources`.

Value

character vector with the names of the enzyme-substrate resources

See Also

`get_resources`, `import_omnipath_enzsub`

Examples

```
get_enzsub_resources()
```

---

get_interaction_resources

Retrieve a list of interaction resources available in Omnipath

Description

Gets the names of the resources from `https://omnipath.org/interactions`.

Usage

```
get_interaction_resources(dataset = NULL)

get_interaction_databases(...)
```

Arguments

`dataset`	a dataset within the interactions query type. Currently available datasets are ‘omnipath‘, ‘kinaseextra‘, ‘pathwayextra‘, ‘ligrecextra‘, ‘dorothea‘, ‘tf_target‘, ‘tf_mirna‘, ‘mirnatarget‘ and ‘lncrna_mrna‘
`...`	Passed to `get_interaction_resources`.

Value

character vector with the names of the interaction databases

See Also

`get_resources`, `import_all_interactions`, `import_omnipath_interactions`, `import_pathwayextra_interacti`

Examples

```
get_interaction_resources()
```

`get_intercell_categories`

Retrieves a list of categories from the intercell database of OmniPath

---

Description

Retrieves a list of categories from `https://omnipath.org/intercell`.

Usage

```
get_intercell_categories()
```

Value

character vector with the different intercell categories

See Also

`import_omnipath_intercell`,`get_intercell_classes`

Examples

```
get_intercell_categories()
```

`get_intercell_generic_categories`

Retrieves a list of the generic categories in the intercell database of OmniPath

---

Description

Retrieves a list of the generic categories from `https://omnipath.org/intercell`.

Usage

```
get_intercell_generic_categories()

get_intercell_classes(...)
```

Arguments

`...`	Passed to `get_intercell_generic_categories`.

Value

character vector with the different intercell main classes

See Also

`import_omnipath_intercell`,`get_intercell_categories`

Examples

```
get_intercell_generic_categories()
```

`get_intercell_resources` *Retrieves a list of intercellular communication resources available in OmniPath*

Description

Retrieves a list of the databases from `https://omnipath.org/intercell`.

Usage

```
get_intercell_resources(dataset = NULL)
```

Arguments

`dataset`	ignored at this query type

Value

character vector with the names of the databases

See Also

`get_resources`,`import_omnipath_intercell`

Examples

```
get_intercell_resources()
```

`get_resources` *Retrieve the available resources for a given query type*

Description

Collects the names of the resources available in OmniPath for a certain query type and optionally for a dataset within that.

Usage

```
get_resources(query_type, datasets = NULL, generic_categories = NULL)
```

Arguments

`query_type`	one of the query types ‘interactions‘, ‘enz_sub‘, ‘complexes‘, ‘annotations‘ or ‘intercell‘
`datasets`	currently within the ‘interactions‘ query type only, multiple datasets are available: ‘omnipath‘, ‘kinaseextra‘, ‘pathwayextra‘, ‘ligrecextra‘, ‘dorothea‘, ‘tf_target‘, ‘tf_mirna‘, ‘mirnatarget‘ and ‘lncrna_mrna‘
`generic_categories`	for the ‘intercell‘ query type, restrict the search for some generic categories e.g. ‘ligand‘ or ‘receptor‘

Value

a character vector with resource names

Examples

```
get_resources(query_type = 'interactions')
```

`get_signed_ptms` *get signs for ptms interactions*

Description

ptms data does not contain sign (activation/inhibition), we generate this information based on the interaction network

Usage

```
get_signed_ptms(
  ptms = import_omnipath_enzsub(),
  interactions = import_omnipath_interactions()
)
```

Arguments

`ptms`	ptms data frame generated by `import_omnipath_enzsub`
`interactions`	interaction data frame generated by `import_omnipath_interactions`

Value

data.frame of ptms with is_inhibition and is_stimulation columns

See Also

`import_omnipath_enzsub` `import_omnipath_interactions`

Examples

```
ptms = import_omnipath_enzsub(resources=c("PhosphoSite", "SIGNOR"))
interactions = import_omnipath_interactions()
ptms = get_signed_ptms(ptms,interactions)
```

`import_all_interactions` *Imports all interaction datasets available in OmniPath*

Description

The interaction datasets currently available in OmniPath:

Usage

```
import_all_interactions(
  cache_file = NULL,
  resources = NULL,
  organism = 9606,
  dorothea_levels = c("A", "B"),
  exclude = NULL,
  fields = NULL,
  default_fields = TRUE,
  references_by_resource = TRUE,
  ...
)

import_AllInteractions(...)
```

Arguments

`cache_file`	path to an earlier data file
`resources`	interactions not reported in these databases are removed. See `get_interaction_resources` for more information.
`organism`	Interactions are available for human, mouse and rat. Choose among: 9606 human (default), 10116 rat and 10090 Mouse
`dorothea_levels`	The confidence levels of the dorothea interactions (TF-target) which range from A to D. Set to A and B by default.
`exclude`	datasets to exclude
`fields`	The user can define here the fields to be added. If used, set the next argument, ‘default_fields‘, to FALSE.
`default_fields`	whether to include the default fields (columns) for the query type. If FALSE, only the fields defined by the user in the ‘fields‘ argument will be added.
`references_by_resource`	if FALSE, removes the resource name prefixes from the references (PubMed IDs); this way the information which reference comes from which resource will be lost and the PubMed IDs will be unique.
`...`	Passed to `import_all_interactions`.

Details

omnipath: the OmniPath data as defined in the paper, an arbitrary optimum between coverage and quality pathwayextra: activity flow interactions without literature reference kinaseextra: enzyme-substrate interactions without literature reference ligrecextra: ligand-receptor interactions without literature reference dorothea: transcription factor (TF)-target interactions from DoRothEA tf_target: transcription factor (TF)-target interactions from other resources mirnatarget: miRNA-mRNA interactions tf_mirna: TF-miRNA interactions lncrna_mrna: lncRNA-mRNA interactions

Value

A dataframe containing all the datasets in the interactions query

See Also

`get_interaction_resources`

Examples

```
interactions <- import_all_interactions(
    resources = c('HPRD', 'BioGRID'),
    organism = 9606
)
```

`import_dorothea_interactions` *From the OmniPath webservice imports interactions from the DoRothEA dataset*

Description

Imports the dataset from: `https://omnipathdb.org/interactions?datasets=dorothea` which contains transcription factor (TF)-target interactions from DoRothEA `https://github.com/saezlab/DoRothEA`

Usage

```
import_dorothea_interactions(
  cache_file = NULL,
  resources = NULL,
  organism = 9606,
  dorothea_levels = c("A", "B"),
  fields = NULL,
  default_fields = TRUE,
  references_by_resource = TRUE,
  ...
)

import_TFregulons_Interactions(...)

import_tfregulons_interactions(...)
```

Arguments

`cache_file`	path to an earlier data file
`resources`	interactions not reported in these databases are removed. See `get_interaction_resources` for more information.
`organism`	Interactions are available for human, mouse and rat. Choose among: 9606 human (default), 10116 rat and 10090 Mouse
`dorothea_levels`	Vector detailing the confidence levels of the interactions to be downloaded. In dorothea, every TF-target interaction has a confidence score ranging from A to E, being A the most reliable interactions. By default we take A and B level interactions (`c(A,B)`). It is to note that E interactions are not available in OmnipathR.
`fields`	The user can define here the fields to be added. If used, set the next argument, 'default_fields', to FALSE.
`default_fields`	whether to include the default fields (columns) for the query type. If FALSE, only the fields defined by the user in the 'fields' argument will be added.
`references_by_resource`	if FALSE, removes the resource name prefixes from the references (PubMed IDs); this way the information which reference comes from which resource will be lost and the PubMed IDs will be unique.
`...`	Passed to `import_dorothea_interactions`.

Value

A dataframe containing TF-target interactions from DoRothEA

See Also

`get_interaction_resources`, `import_all_interactions`

Examples

```
interactions <- import_dorothea_interactions(
    resources = c('DoRothEA_A', 'ARACNe-GTEx_DoRothEA'),
    organism = 9606
)
```

`import_intercell_network` *Imports an intercellular network combining annotations and interactions*

Description

Imports an intercellular network by mapping intercellular annotations and protein interactions. First imports a network of protein-protein interactions. Then, it retrieves annotations about the proteins intercellular communication roles, once for the transmitter (delivering information from the expressing cell) and second, the receiver (receiving signal and relaying it towards the expressing cell) side. These 3 queries can be customized by providing parameters in lists which will be passed to the respective methods (`import_omnipath_interactions` for the network and `import_omnipath_intercell` for the annotations). Finally the 3 data frames combined in a way that the source proteins in each interaction annotated by the transmitter, and the target proteins by the receiver categories. If undirected interactions present (these are disabled by default) they will be duplicated, i.e. both partners can be both receiver and transmitter. If a cache file provided, its content will be returned without any further filtering.

Usage

```
import_intercell_network(
  cache_file = NULL,
  interactions_param = list(),
  transmitter_param = list(),
  receiver_param = list()
)
```

Arguments

`cache_file`	path to an earlier data file; if exists, will be loaded as it is, the further arguments have no effect; if does not exists, the result will be dumped into this file.
`interactions_param`	a list with arguments for an interactions query: `import_omnipath_interactions`, `import_pathway`
`transmitter_param`	a list with arguments for `import_omnipath_intercell`, to define the transmitter side of intercellular connections
`receiver_param`	a list with arguments for `import_omnipath_intercell`, to define the receiver side of intercellular connections

Value

A dataframe containing information about protein-protein interactions and the inter-cellular roles of the protiens involved in those interactions.

See Also

`get_intercell_categories`, `get_intercell_generic_categories`, `import_omnipath_intercell`, `import_omnip`

Examples

```
intercellNetwork <- import_intercell_network(
   interactions_param = list(datasets = 'ligrecextra'),
   receiver_param = list(categories = c('receptor', 'transporter')),
   transmitter_param = list(categories = c('ligand', 'secreted_enzyme')))
```

`import_kinaseextra_interactions` *Imports interactions from the ‘kinase extra‘ dataset of OmniPath*

Description

Imports the dataset from: `https://omnipathdb.org/interactions?datasets=kinaseextra`, which contains enzyme-substrate interactions without literature reference. The enzyme-substrate interactions supported by literature references are part of the ‘omnipath‘ dataset.

Usage

```
import_kinaseextra_interactions(
  cache_file = NULL,
  resources = NULL,
  organism = 9606,
  fields = NULL,
  default_fields = TRUE,
  references_by_resource = TRUE,
  ...
)

import_KinaseExtra_Interactions(...)
```

Arguments

`cache_file`	path to an earlier data file
`resources`	interactions not reported in these databases are removed. See `get_interaction_resources` for more information.
`organism`	Interactions are available for human, mouse and rat. Choose among: 9606 human (default), 10116 rat and 10090 Mouse
`fields`	The user can define here the fields to be added. If used, set the next argument, ‘default_fields‘, to FALSE.
`default_fields`	whether to include the default fields (columns) for the query type. If FALSE, only the fields defined by the user in the ‘fields‘ argument will be added.
`references_by_resource`	if FALSE, removes the resource name prefixes from the references (PubMed IDs); this way the information which reference comes from which resource will be lost and the PubMed IDs will be unique.
`...`	Passed to `import_kinaseextra_interactions`.

Value

A dataframe containing enzyme-substrate interactions without literature reference

See Also

`get_interaction_resources`, `import_all_interactions`

Examples

```
interactions <-
   import_kinaseextra_interactions(
       resources = c('PhosphoPoint', 'PhosphoSite'),
       organism = 9606
   )
```

---

import_ligrecextra_interactions

Imports interactions from the 'ligrec extra' dataset of OmniPath

Description

Imports the dataset from: https://omnipathdb.org/interactions?datasets=ligrecextra, which contains ligand-receptor interactions without literature reference. The ligand-receptor interactions supported by literature references are part of the 'omnipath' dataset.

Usage

```
import_ligrecextra_interactions(
  cache_file = NULL,
  resources = NULL,
  organism = 9606,
  fields = NULL,
  default_fields = TRUE,
  references_by_resource = TRUE,
  ...
)

import_LigrecExtra_Interactions(...)
```

Arguments

`cache_file`	path to an earlier data file
`resources`	interactions not reported in these databases are removed. See `get_interaction_resources` for more information.
`organism`	Interactions are available for human, mouse and rat. Choose among: 9606 human (default), 10116 rat and 10090 Mouse
`fields`	The user can define here the fields to be added. If used, set the next argument, 'default_fields', to FALSE.
`default_fields`	whether to include the default fields (columns) for the query type. If FALSE, only the fields defined by the user in the 'fields' argument will be added.
`references_by_resource`	if FALSE, removes the resource name prefixes from the references (PubMed IDs); this way the information which reference comes from which resource will be lost and the PubMed IDs will be unique.
`...`	Passed to `import_ligrecextra_interactions`.

Value

A dataframe containing ligand-receptor interactions including the ones without literature references

See Also

`get_interaction_resources`, `import_all_interactions`

Examples

```
interactions <- import_ligrecextra_interactions(
    resources = c('HPRD', 'Guide2Pharma'),
    organism = 9606
)
```

`import_lncrna_mrna_interactions`

Imports interactions from the lncRNA-mRNA dataset of OmniPath

Description

Imports the dataset from: `https://omnipathdb.org/interactions?datasets=lncrna_mrna`, which contains lncRNA-mRNA interactions

Usage

```
import_lncrna_mrna_interactions(
  cache_file = NULL,
  resources = NULL,
  organism = 9606,
  fields = NULL,
  default_fields = TRUE,
  references_by_resource = TRUE,
  ...
)
```

Arguments

`cache_file`	path to an earlier data file
`resources`	interactions not reported in these databases are removed. See `get_interaction_resources` for more information.
`organism`	Interactions are available for human, mouse and rat. Choose among: 9606 human (default), 10116 rat and 10090 Mouse
`fields`	The user can define here the fields to be added. If used, set the next argument, ‘default_fields‘, to FALSE.
`default_fields`	whether to include the default fields (columns) for the query type. If FALSE, only the fields defined by the user in the ‘fields‘ argument will be added.
`references_by_resource`	if FALSE, removes the resource name prefixes from the references (PubMed IDs); this way the information which reference comes from which resource will be lost and the PubMed IDs will be unique.
`...`	optional additional arguments

Value

A dataframe containing lncRNA-mRNA interactions

See Also

`get_interaction_resources`, `import_all_interactions`

Examples

```
interactions <-
    import_lncrna_mrna_interactions(
        resources = c('ncRDeathDB')
    )
```

`import_mirnatarget_interactions`

Imports interactions from the miRNA-target dataset of OmniPath

---

Description

Imports the dataset from: `https://omnipathdb.org/interactions?datasets=mirnatarget`, which contains miRNA-mRNA interactions.

Usage

```
import_mirnatarget_interactions(
  cache_file = NULL,
  resources = NULL,
  organism = 9606,
  fields = NULL,
  default_fields = TRUE,
  references_by_resource = TRUE,
  ...
)

import_miRNAtarget_Interactions(...)
```

Arguments

`cache_file`	path to an earlier data file
`resources`	interactions not reported in these databases are removed. See `get_interaction_resources` for more information.
`organism`	Interactions are available for human, mouse and rat. Choose among: 9606 human (default), 10116 rat and 10090 Mouse
`fields`	The user can define here the fields to be added. If used, set the next argument, ‘default_fields‘, to FALSE.
`default_fields`	whether to include the default fields (columns) for the query type. If FALSE, only the fields defined by the user in the ‘fields‘ argument will be added.

`references_by_resource`	if FALSE, removes the resource name prefixes from the references (PubMed IDs); this way the information which reference comes from which resource will be lost and the PubMed IDs will be unique.
`...`	Passed to `import_mirnatarget_interactions`.

Value

A dataframe containing miRNA-mRNA interactions

See Also

`get_interaction_resources`,`import_all_interactions`

Examples

```
interactions <-
    import_mirnatarget_interactions(
        resources = c('miRTarBase', 'miRecords')
    )
```

`import_omnipath_annotations`

Imports annotations from OmniPath

Description

Imports protein annotations about function, localization, expression, structure and other properties of proteins from OmniPath `https://omnipathdb.org/annotations`. Note: there might be also a few miRNAs annotated; a vast majority of protein complex annotations are inferred from the annotations of the members: if all members carry the same annotation the complex inherits.

Usage

```
import_omnipath_annotations(
  cache_file = NULL,
  proteins = NULL,
  resources = NULL,
  force_full_download = FALSE,
  wide = FALSE,
  ...
)

import_Omnipath_annotations(...)

import_OmniPath_annotations(...)
```

Arguments

`cache_file`	Path to an earlier data file
`proteins`	Vector containing the genes or proteins for whom annotations will be retrieved (UniProt IDs or HGNC Gene Symbols or miRBase IDs). It is also possible to donwload annotations for protein complexes. To do so, write 'COMPLEX:' right before the genesymbols of the genes integrating the complex. Check the vignette for examples.
`resources`	Load the annotations only from these databases. See `get_annotation_resources` for possible values.
`force_full_download`	Force the download of the entire annotations dataset. This is disabled by default because the size of this data is around 1GB. We recommend to retrieve the annotations for a set of proteins or only from a few resources, depending on your interest.
`wide`	Convert the annotation table to wide format, which corresponds more or less to the original resource. If the data comes from more than one resource a list of wide tables will be returned.
`...`	Passed to `import_omnipath_annotations`.

Value

A data.frame containing different gene/complex annotations

See Also

`get_annotation_databases`

Examples

```
annotations = import_omnipath_annotations(
    proteins = c('TP53', 'LMNA'),
    resources = c('HPA_subcellular')
)
```

`import_omnipath_complexes`

Imports protein complexes from OmniPath

Description

Imports the complexes stored in Omnipath database from `https://omnipathdb.org/complexes`.

Usage

```
import_omnipath_complexes(cache_file = NULL, resources = NULL, ...)

import_Omnipath_complexes(...)

import_OmniPath_complexes(...)
```

Arguments

`cache_file`	path to an earlier data file
`resources`	complexes not reported in these databases are removed. See `get_complexes_databases` for more information.
`...`	Passed to `import_omnipath_complexes`.

Value

A dataframe containing information about complexes

See Also

`get_complexes_databases`

Examples

```
complexes = import_omnipath_complexes(
    resources = c('CORUM', 'hu.MAP')
)
```

`import_omnipath_enzsub`

Imports enzyme-substrate relationships from OmniPath

Description

Imports the enzyme-substrate (more exactly, enzyme-PTM) relationship database from `https://omnipathdb.org/enzsub`

Usage

```
import_omnipath_enzsub(
  cache_file = NULL,
  resources = NULL,
  organism = 9606,
  fields = NULL,
  default_fields = TRUE,
  references_by_resource = TRUE,
  ...
)

import_Omnipath_PTMS(...)

import_OmniPath_PTMS(...)
```

Arguments

`cache_file`	path to an earlier data file
`resources`	PTMs not reported in these databases are removed. See `get_ptms_databases` for more information
`organism`	PTMs are available for human, mouse and rat. Choose among: 9606 human (default), 10116 rat and 10090 Mouse
`fields`	The user can define here the fields to be added. If used, set the next argument, ‘default_fields‘, to FALSE.
`default_fields`	whether to include the default fields (columns) for the query type. If FALSE, only the fields defined by the user in the ‘fields‘ argument will be added.
`references_by_resource`	if FALSE, removes the resource name prefixes from the references (PubMed IDs); this way the information which reference comes from which resource will be lost and the PubMed IDs will be unique.
`...`	Passed to `import_omnipath_enzsub`.

Value

A data frame containing the information about ptms

See Also

`get_ptms_databases`,`import_omnipath_interactions`

Examples

```
ptms = import_omnipath_enzsub(
    resources = c('PhosphoSite', 'SIGNOR'),
    organism = 9606
)
```

`import_omnipath_interactions`

Imports interactions from the ‘omnipath‘ dataset of Omnipath

Description

Imports the database from `https://omnipathdb.org/interactions`, which contains only interactions supported by literature references. This part of the interaction database compiled a similar way as it has been presented in the first paper describing OmniPath (Turei et al. 2016).

Usage

```
import_omnipath_interactions(
  cache_file = NULL,
  resources = NULL,
  organism = 9606,
  datasets = "omnipath",
  fields = NULL,
```

```
  default_fields = TRUE,
  references_by_resource = TRUE,
  ...
)

import_Omnipath_Interactions(...)

import_OmniPath_Interactions(...)
```

Arguments

`cache_file`	path to an earlier data file
`resources`	interactions not reported in these databases are removed. See `get_interaction_resources` for more information.
`organism`	Interactions are available for human, mouse and rat. Choose among: 9606 human (default), 10116 rat and 10090 Mouse
`datasets`	Names of the interaction datasts to download: omnipath (by default). Other possiblites are: pathwayextra, kinaseextra, ligrecextra, dorothea,tf_target, mirnatarget, tf_mirna, lncrna_mrna. The user can select multiple datasets as for example: c('omnipath', 'pathwayextra', 'kinaseextra')
`fields`	The user can define here the fields to be added. If used, set the next argument, ‘default_fields‘, to FALSE.
`default_fields`	whether to include the default fields (columns) for the query type. If FALSE, only the fields defined by the user in the ‘fields‘ argument will be added.
`references_by_resource`	if FALSE, removes the resource name prefixes from the references (PubMed IDs); this way the information which reference comes from which resource will be lost and the PubMed IDs will be unique.
`...`	Passed to `import_omnipath_interactions`.

Value

A dataframe of protein-protein interactions

See Also

`get_interaction_resources`,`import_all_interactions`

Examples

```
interactions = import_omnipath_interactions(
    resources = c('SignaLink3'),
    organism = 9606
)
```

`import_omnipath_intercell`

Imports OmniPath intercell annotations

Description

Imports the OmniPath intercellular communication role annotation database from `https://omnipathdb.org/intercell`. It provides information on the roles in inter-cellular signaling. E.g. if a protein is a ligand, a receptor, an extracellular matrix (ECM) component, etc.

Usage

```
import_omnipath_intercell(
  cache_file = NULL,
  categories = NULL,
  resources = NULL,
  parent = NULL,
  scope = NULL,
  aspect = NULL,
  source = NULL,
  transmitter = NULL,
  receiver = NULL,
  secreted = NULL,
  plasma_membrane_peripheral = NULL,
  plasma_membrane_transmembrane = NULL,
  proteins = NULL,
  topology = NULL,
  causality = NULL,
  ...
)

import_Omnipath_intercell(...)

import_OmniPath_intercell(...)
```

Arguments

`cache_file`	path to an earlier data file
`categories`	vector containing the categories to be retrieved. All the genes belonging to those categories will be returned. For further information about the categories see `get_intercell_categories`
`resources`	limit the query to certain resources; see the available resources by `get_intercell_resources`
`parent`	vector containing the parent classes to be retrieved. All the genes belonging to those classes will be returned. For furter information about the main classes see `get_intercell_categories`
`scope`	either ‘specific‘ or ‘generic‘
`aspect`	either ‘locational‘ or ‘functional‘
`source`	either ‘resource_specific‘ or ‘composite‘

`transmitter`	logical, include only transmitters i.e. proteins delivering signal from a cell to its environment
`receiver`	logical, include only receivers i.e. proteins delivering signal to the cell from its environment
`secreted`	logical, include only secreted proteins
`plasma_membrane_peripheral`	logical, include only plasma membrane peripheral membrane proteins
`plasma_membrane_transmembrane`	logical, include only plasma membrane transmembrane proteins
`proteins`	limit the query to certain proteins
`topology`	topology categories: one or more of ‘secreted‘ (sec), ‘plasma_membrane_peripheral‘ (pmp), ‘plasma_membrane_transmembrane‘ (pmtm) (both short or long notation can be used)
`causality`	‘transmitter‘ (trans), ‘receiver‘ (rec) or ‘both‘ (both short or long notation can be used)
`...`	Passed to `import_omnipath_intercell`.

Value

A dataframe cotaining information about roles in intercellular signaling.

See Also

`get_intercell_categories`,`get_intercell_generic_categories`,`import_intercell_network`

Examples

```
intercell = import_omnipath_intercell(categories = c('ecm'))
```

`import_pathwayextra_interactions`

Imports interactions from the ‘pathway extra‘ dataset of Omnipath

Description

Imports the dataset from: `https://omnipathdb.org/interactions?datasets=pathwayextra`, which contains activity flow interactions without literature reference. The activity flow interactions supported by literature references are part of the ‘omnipath‘ dataset.

Usage

```
import_pathwayextra_interactions(
  cache_file = NULL,
  resources = NULL,
  organism = 9606,
  fields = NULL,
  default_fields = TRUE,
  references_by_resource = TRUE,
```

```
    ...
)

import_PathwayExtra_Interactions(...)
```

Arguments

`cache_file`	path to an earlier data file
`resources`	interactions not reported in these databases are removed. See `get_interaction_resources` for more information.
`organism`	Interactions are available for human, mouse and rat. Choose one of those: 9606 human (default), 10116 rat or 10090 Mouse.
`fields`	The user can define here the fields to be added. If used, set the next argument, 'default_fields', to FALSE.
`default_fields`	whether to include the default fields (columns) for the query type. If FALSE, only the fields defined by the user in the 'fields' argument will be added.
`references_by_resource`	if FALSE, removes the resource name prefixes from the references (PubMed IDs); this way the information which reference comes from which resource will be lost and the PubMed IDs will be unique.
`...`	Passed to `import_pathwayextra_interactions`.

Value

A dataframe containing activity flow interactions between proteins without literature reference

See Also

`get_interaction_resources`, `import_all_interactions`

Examples

```
interactions <-
    import_pathwayextra_interactions(
        resources = c('BioGRID', 'IntAct'),
        organism = 9606
    )
```

`import_post_translational_interactions`

Imports all post-translational interactions from OmniPath

Description

Imports the dataset from all post-translational datasets of OmniPath.

Usage

```
import_post_translational_interactions(
  resources = NULL,
  organism = 9606,
  exclude = NULL,
  references_by_resource = TRUE,
  ...
)
```

Arguments

`resources`	interactions not reported in these databases are removed. See `get_interaction_resources` for more information.
`organism`	Interactions are available for human, mouse and rat. Choose among: 9606 human (default), 10116 rat and 10090 Mouse
`exclude`	datasets to exclude
`references_by_resource`	if FALSE, removes the resource name prefixes from the references (PubMed IDs); this way the information which reference comes from which resource will be lost and the PubMed IDs will be unique.
`...`	optional additional arguments

Value

A dataframe containing post-translational interactions

See Also

`get_interaction_resources`, `import_all_interactions`

Examples

```
interactions <-
    import_post_translational_interactions(
        resources = c('BioGRID')
    )
```

`import_tf_mirna_interactions`

Imports interactions from the TF-miRNA dataset of OmniPath

Description

Imports the dataset from: `https://omnipathdb.org/interactions?datasets=tf_mirna`, which contains transcription factor-miRNA gene interactions

Usage

```
import_tf_mirna_interactions(
  cache_file = NULL,
  resources = NULL,
  organism = 9606,
  fields = NULL,
  default_fields = TRUE,
  references_by_resource = TRUE,
  ...
)
```

Arguments

`cache_file`	path to an earlier data file
`resources`	interactions not reported in these databases are removed. See `get_interaction_resources` for more information.
`organism`	Interactions are available for human, mouse and rat. Choose among: 9606 human (default), 10116 rat and 10090 Mouse
`fields`	The user can define here the fields to be added. If used, set the next argument, ‘default_fields‘, to FALSE.
`default_fields`	whether to include the default fields (columns) for the query type. If FALSE, only the fields defined by the user in the ‘fields‘ argument will be added.
`references_by_resource`	if FALSE, removes the resource name prefixes from the references (PubMed IDs); this way the information which reference comes from which resource will be lost and the PubMed IDs will be unique.
`...`	optional additional arguments

Value

A dataframe containing TF-miRNA interactions

See Also

`get_interaction_resources`, `import_all_interactions`

Examples

```
interactions <-
    import_tf_mirna_interactions(
        resources = c('TransmiR')
    )
```

`import_tf_target_interactions`

Imports interactions from the TF-target dataset of OmniPath

Description

Imports the dataset from: https://omnipathdb.org/interactions?datasets=tf_target, which contains transcription factor-target protein coding gene interactions. Note: this is not the only TF-target dataset in OmniPath, ‘dorothea‘ is the other one and the ‘tf_mirna‘ dataset provides TF-miRNA gene interactions.

Usage

```
import_tf_target_interactions(
  cache_file = NULL,
  resources = NULL,
  organism = 9606,
  fields = NULL,
  default_fields = TRUE,
  references_by_resource = TRUE,
  ...
)
```

Arguments

`cache_file`	path to an earlier data file
`resources`	interactions not reported in these databases are removed. See `get_interaction_resources` for more information.
`organism`	Interactions are available for human, mouse and rat. Choose among: 9606 human (default), 10116 rat and 10090 Mouse
`fields`	The user can define here the fields to be added. If used, set the next argument, ‘default_fields‘, to FALSE.
`default_fields`	whether to include the default fields (columns) for the query type. If FALSE, only the fields defined by the user in the ‘fields‘ argument will be added.
`references_by_resource`	if FALSE, removes the resource name prefixes from the references (PubMed IDs); this way the information which reference comes from which resource will be lost and the PubMed IDs will be unique.
`...`	optional additional arguments

Value

A dataframe containing TF-target interactions

See Also

`get_interaction_resources`, `import_all_interactions`

Examples

```
interactions <-
    import_tf_target_interactions(
        resources = c('DoRothEA_A', 'SIGNOR')
    )
```

`import_transcriptional_interactions`

Imports all TF-target interactions from OmniPath

Description

Imports the dataset from: `https://omnipathdb.org/interactions?datasets=tf_target,dorothea`, which contains transcription factor-target protein coding gene interactions.

Usage

```
import_transcriptional_interactions(
  resources = NULL,
  organism = 9606,
  dorothea_levels = c("A", "B"),
  references_by_resource = TRUE,
  ...
)
```

Arguments

`resources`	interactions not reported in these databases are removed. See `get_interaction_resources` for more information.
`organism`	Interactions are available for human, mouse and rat. Choose among: 9606 human (default), 10116 rat and 10090 Mouse
`dorothea_levels`	Vector detailing the confidence levels of the interactions to be downloaded. In dorothea, every TF-target interaction has a confidence score ranging from A to E, being A the most reliable interactions. By default we take A and B level interactions (`c(A,B)`). It is to note that E interactions are not available in OmnipathR.
`references_by_resource`	if FALSE, removes the resource name prefixes from the references (PubMed IDs); this way the information which reference comes from which resource will be lost and the PubMed IDs will be unique.
`...`	optional additional arguments

Value

A dataframe containing TF-target interactions

See Also

`get_interaction_resources`, `import_all_interactions`

Examples

```
interactions <-
    import_transcriptional_interactions(
        resources = c('PAZAR', 'ORegAnno', 'DoRothEA_A')
    )
```

`interaction_graph` *Build Omnipath interaction graph*

Description

transforms the interactions data.frame to an igraph object

Usage

```
interaction_graph(interactions = interactions)
```

Arguments

`interactions`	data.frame created by `import_omnipath_interactions`, `import_pathwayextra_interactions`, `import_kinaseextra_interactions`, `import_ligrecextra_interactions`, `import_dorothea_interactions`, `import_mirnatarget_interactions` or `import_all_interac`

Value

An igraph object

See Also

`import_omnipath_interactions`, `import_pathwayextra_interactions`, `import_kinaseextra_interactions`, `import_ligrecextra_interactions`, `import_dorothea_interactions`, `import_mirnatarget_interactions` or `import_all_interactions`

Examples

```
interactions = import_omnipath_interactions(resources=c("SignaLink3"))
OPI_g = interaction_graph(interactions)
```

`OmnipathR` *The OmnipathR package*

Description

OmnipathR is an R package built to provide easy access to the data stored in the OmniPath web service:

https://omnipathdb.org/

The web service implements a very simple REST style API. This package make requests by the HTTP protocol to retreive the data. Hence, fast Internet access is required for a propser use of OmnipathR.

The package also provides some utility functions to filter, analyse and visualize the data.

Author(s)

Alberto Valdeolivas <<`alvaldeolivas@gmail`>> and Denes Turei <<`turei.denes@gmail.com`>> and Attila Gabor <<`gaborattila87@gmail.com`>>

Examples

```
# Download post-translational modifications:
ptms = import_omnipath_enzsub(resources=c("PhosphoSite", "SIGNOR"))

# Download protein-protein interactions
interactions = import_omnipath_interactions(resources=c("SignaLink3"))

# Convert to igraph objects:
ptms_g = ptms_graph(ptms = ptms )
OPI_g = interaction_graph(interactions = interactions )

# Print some interactions:
print_interactions(head(ptms))

# interactions with references:
print_interactions(tail(ptms),writeRefs=TRUE)

# find interactions between kinase and substrate:
print_interactions(dplyr::filter(ptms,enzyme_genesymbol=="MAP2K1",
   substrate_genesymbol=="MAPK3"))

# find shortest paths on the directed network between proteins
print_path_es(shortest_paths(OPI_g,from = "TYRO3",to = "STAT3",
   output = 'epath')$epath[[1]],OPI_g)

# find all shortest paths between proteins
print_path_vs(
    all_shortest_paths(
        ptms_g,
        from = "SRC",
        to = "STAT1"
    )$res,
    ptms_g
)
```

`pivot_annotations` *Converts annotation tables to a wide format*

Description

Use this method to reconstitute the annotation tables into the format of the original resources. With the ‘wide=TRUE‘ option `import_omnipath_annotations` applies this function to the downloaded data.

Usage

```
pivot_annotations(annotations)
```

Arguments

`annotations`	A data frame of annotations downloaded from the OmniPath web service.

Value

A wide format tibble if the provided data contains annotations from one resource, otherwise a list of wide format tibbles.

`print_interactions` *print interactions*

Description

prints the interactions/ptms in a nice format

Usage

```
print_interactions(interDF, writeRefs = FALSE)
```

Arguments

`interDF`	data.frame with the interactions generated by any of the following functions: `import_omnipath_enzsub`, `import_omnipath_interactions`, `import_pathwayextra_interacti` `import_kinaseextra_interactions`, `import_ligrecextra_interactions`, `import_dorothea_interactions`, `import_mirnatarget_interactions` or `import_all_interac`
`writeRefs`	[FALSE] writes also the PubMed IDs if available

Value

Interactions displayed in a nice format

Examples

```
ptms = import_omnipath_enzsub()
print_interactions(head(ptms))
print_interactions(tail(ptms),writeRefs=TRUE)
print_interactions(dplyr::filter(ptms,enzyme_genesymbol=="MAP2K1",
  substrate_genesymbol=="MAPK3"))
```

`print_path_es` *print network paths given by edge sequence*

Description

Prints the interactions in the path in a nice format.

Usage

```
print_path_es(edgeSeq, G)

printPath_es(...)
```

Arguments

`edgeSeq`	edge sequence
`G`	igraph object (from ptms or any interaction dataset)
`...`	Passed to `print_path_es`.

Value

Interactions displayed in a nice format

See Also

`print_path_vs`

Examples

```
interactions = import_omnipath_interactions(resources=c("SignaLink3"))
OPI_g = interaction_graph(interactions = interactions )
print_path_es(shortest_paths(OPI_g,from = "TYRO3",to = "STAT3",
    output = 'epath')$epath[[1]],OPI_g)
```

---

print_path_vs *print networks paths given by node sequence*

Description

Prints the interactions in the path in a nice format.

Usage

```
print_path_vs(nodeSeq, G)

printPath_vs(...)
```

Arguments

`nodeSeq`	node sequence
`G`	igraph object (from ptms or interactions)
`...`	Passed to `print_path_vs`.

Value

Interactions displayed in a nice format

See Also

`print_path_es`

Examples

```
interactions = import_omnipath_interactions(resources=c("SignaLink3"))
OPI_g = interaction_graph(interactions = interactions )
print_path_vs(
    all_shortest_paths(
        OPI_g,
        from = "TYRO3",
        to = "STAT3"
    )$vpath,
    OPI_g
)
ptms = import_omnipath_enzsub(resources=c("PhosphoSite", "SIGNOR"))
ptms_g = ptms_graph(ptms)
print_path_vs(
    all_shortest_paths(
        ptms_g,
        from = "SRC",
        to = "STAT1"
    )$res,
    ptms_g
)
```

`ptms_graph` *Post-translational modifications (PTMs) graph*

Description

transforms the ptms interactions data.frame to igraph object

Usage

```
ptms_graph(ptms)
```

Arguments

`ptms`	data.frame created by `import_omnipath_enzsub`

Value

An igraph object

See Also

`import_omnipath_enzsub`

Examples

```
ptms = import_omnipath_enzsub(resources=c("PhosphoSite", "SIGNOR"))
ptms_g = ptms_graph(ptms = ptms )
```

Index