Package 'rUniChEMBL'

June 11, 2014

Title Accessing the ChEMBL and Unichem data through web services

Description What the package does (paragraph)

Version 1.0

Author Abhik Seal <abseal@indiana.edu>

Maintainer Abhik Seal <abseal@indiana.edu>

Depends R (>= 3.0.3),RCurl,jsonlite

License MIT

imports RCurl,jsonlite

LazyData true

R topics documented:

get.appDrugs *Get approved drugs for target.*

Description

This function retrieves a dataframe of all approved drug compounds from ChEMBL database given a string of ChEMBL target ID.

Usage

```
get.appDrugs(x)
```

Arguments

x	: string ChEMBL target ID.

Details

get.appDrugs

Examples

```
#get chembl ids of approved drugs
get.appDrugs("CHEMBL1824")
```

get.bioactivity *Get Bioactivity Information for Compounds, Targets or Assays.*

Description

This method retrieves bioactivity information for a compound across multiple targets/assays or else for a target across multiple compounds. The function can also be used to retrieve all activities within a given assay. In all cases, ChEMBL identifiers must be used.

Usage

```
get.bioactivity(x, type = "compound")
```

Arguments

x	: Input string chemblid
type	: Input string compound,target,assay. Default is compound.

Details

get.bioactivity.

Examples

```
# get bioactivities of compounds
get.bioactivity("CHEMBL12",type=compound)

# get compound bioactivities for targets
get.bioactivity("CHEMBL240",type="target")

# get bioactivities by assay
get.bioactivity("CHEMBL1217643",type=assay)
```

`get.cmp.inf` *Get compound information from ChEMBL*

Description

These functions allow one to retrieve compounds information from ChEMBL compounds are identified either by a ChEMBL ID or by a standard InChI key.

Usage

```
get.cmp.inf(x, type = "chemblid")
```

Arguments

`x`	: String representing chemblid or standard InCHI key for the molecule.
`type`	: For `get.compound`, one of `chemblid` or `stdinchi` to indicate the nature of the molecule id.

Details

get.cmp.inf

Examples

```
#get information for chembl compound id
get.compound("CHEMBL12")

#get information for standard inchi
get.compound("QFFGVLORLPOAEC-SNVBAGLBSA-N",type=stdinchi)
```

get.cmp.sim *Retrive similar compounds from ChEMBL database.*

Description

This function retrieves a dataframe of similar compounds from ChEMBL database given a smiles string as query and also given a similarity score above 70.

Usage

```
get.cmp.sim(mol, sim = 70)
```

Arguments

mol	: String representing smiles of the moelcule
sim	: Integer representing for percentage of similarity for the query compound and the database molecules. Values ranges from 70 to 100.

Details

get.cmp.sim

Examples

```
# Get compound similarity with 70 percent or more
get.cmp.sim("c1ccc(cc1)CN(c2cc(ccc2[N+](=O)[O-])c3c(nc(nc3CC)N)N)C",sim=70)
# Get compound similarity with 80 percent or more
get.cmp.sim("c1ccc(cc1)CN(c2cc(ccc2[N+](=O)[O-])c3c(nc(nc3CC)N)N)C",sim=80)
```

get.compound.substruct *Get compound information from substructure query smiles.*

Description

This function retrieves a dataframe of all compounds from ChEMBL database containing the substructure represented by the given Canonical SMILES and their chemical properties.

Usage

```
get.cmp.substruct(mol)
```

Arguments

mol	: String representing smiles of the moelcule

Details

get.cmp.substruct

Examples

```
#get compounds by substructure
get.cmp.subsruct("CN(CCCN)c1cccc2ccccc12")
```

`get.mapping.full` *Get full mapping between two sources.*

Description

Obtain a full mapping between two sources. Uses only currently assigned src_compound_ids from both sources.

Usage

```
get.mapping.full(x, y)
```

Arguments

x	: Input integer source id
y	: Input integer source id

Details

get.mapping.full

Examples

```
# Get full mapping of PDBe and ChEMBL
get.mapping.full(3,1)
# Get full mapping of ZINC and ChEMBL
get.mapping.full(9,1)
```

`get.moa` *Get mechanism of action*

Description

This function retrieves a data frame of compounds and its mode of action for a compound (where compound is a drug) and drug targets.

Usage

```
get.moa(x)
```

Arguments

x	: Input string chemblid

Details

get.moa

Examples

```
# get moa of drug
get.moa("CHEMBL1642")
```

`get.sAll.InCHIKey` *Get all src_compound_ids.*

Description

Get a list of all src_compound_ids (from all sources) which have current AND obsolete assignments to a query InChIKey

Usage

```
get.sAll.InCHIKey(x)
```

Arguments

x	: Input string InCHI Key

Details

get.sAll.InCHIKey

Examples

```
# Get all the IDs using InCHIKey
 get.sAll.InCHIKey("AAOVKJBEBIDNHE-UHFFFAOYSA-N")
```

`get.sAll.sid` *Get the all source compound ids from another source compound id*

Description

Obtain a list of all src_compound_ids from all sources (including BOTH current AND obsolete assignments) to the same structure as a currently assigned query src_compound_id.

Usage

```
get.sAll.sid(x, y)
```

Arguments

x	: Input chemblid
y	: Input integer source id

Details

get.sAll.sid

Examples

```
# Get all source ids using ChEMBL id and source
get.sAll.sid("CHEMBL12",1)
# Using drugbank id and source
get.sAll.sid("DB00789",2)
```

---

get.scid.sid *Get the source compound ids from another source compound id*

Description

a list of all src_compound_ids from all sources which are CURRENTLY assigned to the same structure as a currently assigned query src_compound_id. The output will include query src_compound_id if it is a valid src_compound_id with a current assignment.

Usage

```
get.scid.sid(x, y)
```

Arguments

x	: Input string Source compound id
y	: Input integer Source id

Details

get.scid.sid

Examples

```
# Get source compound ids and source information
# Using ChEMBL ID and source
get.scid.sid("CHEMBL12",1)
# Using drugbank id and source
get.scid.sid("DB00789",2)
```

get.SrcAll.obs *Get source compound id from obsolete source compound id*

Description

Get a list of all src_compound_ids from all sources with BOTH current AND obsolete to the same structure with an obsolete assignment to the #' query src_compound_id.

Usage

```
get.sAll.obs(x, y)
```

Arguments

x	: Input string source compound id
y	: Input integer to source id

Details

get.sAll.obs

Examples

```
#get for drugbank compound and source
get.sAll.obs("DB07699",2)
#get for chembl compound and source
get.sAll.obs("CHEMBL12",1)
```

get.src_id.InCHIKey *Get source compound ids*

Description

Obtain a list of src_compound_ids (from all sources) which are CURRENTLY assigned to a query InChI Key.

Usage

```
get.sid.InCHIKey(x)
```

Arguments

x	: Input string InCHI Key

Details

get.src_id.InCHIKey

Examples

```
# Get source compound ids from InCHIKey
get.sid.InCHIKey("AAOVKJBEBIDNHE-UHFFFAOYSA-N")
get.sid.InCHIKey("BSYNRYMUTXBXSQ-UHFFFAOYSA-N")
```

---

get.struc.all *Get Structures for source compound id*

Description

Get the standard InCHI and standard InCHI Key for the source compound id

Usage

```
get.struc.all(x, s = 1)
```

Arguments

x	: Input string chemblid
s	: Input integer source id (default is 1)

Details

get.struc.all

Examples

```
# Get all the structure information using ChEMBL id and source.
get.struc.all("CHEMBL1231",1)
#using drugbank id and source
get.structure("DB00321",2)
```

---

get.structure *Get structure*

Description

Get structure(s) currently assigned to a query src_compound_id.

Usage

```
get.structure(x, s = 1)
```

Arguments

x	: Input string chemblid
s	: Input integer source id (default is 1)

Details

get.structure

Examples

```
# Get Standard inhci and InCHIKey from drugbank compound and source
get.structure("DB00321",2)
# Using ChEMBL compound and source id
get.structure("CHEMBL1231",1)
```

`get.targets` *Get target information.*

Description

This function retrieves the target information by chembl id and uniprot id and also retrieves all the names of targets by organisms. When `org="Homo sapiens"` subsets the data frame by organism homo sapiens and retrieves all the Homo sapiens taregts

Usage

```
get.targets(x, type = "chemblid", org = NULL)
```

Arguments

`x`	: Input string chemblid
`type`	: Input string 'chemblid' or 'uniprot'
`org`	: Input string species name like "Homo sapiens","Plasmodium falciparum" and etc.

Details

get.targets

Examples

```
#get target information by chembl ids
get.targets("CHEMBL1862",type=chemblid)

#get target information by uniprot ids
get.targets("Q13936",type=uniprot)

#get all the target information using organism name
get.targets(org="Homo Sapiens")
```

get.url.sid *Get url for the query compound*

Description

Get a list of URLs for all src_compound_ids, from a specified source .

Usage

```
get.url.sid(x, y, z)
```

Arguments

x	: Input string source compound id
y	: Input integer source id
z	: Input integer to source id

Details

get.url.sid

Examples

```
# get urls of compounds using source compound id, source id
# get drugbank url from ChEMBL source id and ChEMBL source
get.url.sid("ChEMBL490",1,2)

# get chembl url from drugbank id and source
get.url.sid("DB00715",2,1)
```

get.verbose.InCHIkey *Get all src_compound_ids to a query InChIKey*

Description

Returns a dataframe containing src_id (the src_id for this source), src_url (the main home page of the source), name (the unique name for the source in UniChem, always lower case), name_long (the full name of the source, as defined by the source),name_label (A name for the source suitable for use as a 'label' for the source within a web-page. Correct case setting for source, and always less than 30 characters), description (a description of the content of the source), base_id_url_available (an flag indicating whether this source provides a valid base_id_url for creating cpd-specific links [1=yes, 0=no]),base_id_url (the base url for constructing hyperlinks to this source [append an identifier from this source to the end of this url to create a valid url to a specific page for this cpd], unless aux_for_url=1), aux_for_url (A flag to indicate whether the aux_src field should be used to create hyperlinks instead of the src_compound_id [1=yes, 0=no] , src_compound_id (a list of src_compound_ids from this source which are currently assigned to the query InChIKey, aux_src (a list of src-compound_id keys mapping to corresponding auxiliary data (url_id:value), for creating links if aux_for_url=1. Only shown if aux_for_url=1).

Usage

```
get.verbose.InCHIkey(x)
```

Arguments

x	: Input string InCHI Key

Details

get.verbose.InCHIkey

Examples

```
# get for InCHIkey
get.verbose.InCHIkey("GZUITABIAKMVPG-UHFFFAOYSA-N")

get.verbose.InCHIkey("AAOVKJBEBIDNHE-UHFFFAOYSA-N")
```

Index