Package 'rUniChEMBL'

June 9, 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 . . . 2
get.bioactivity . . . 2
get.cmp.sim . . . 3
get.compound . . . 3
get.compound.substruct . . . 4
get.mapping.full . . . 4
get.moa . . . 5
get.sAll.InCHIKey . . . 5
get.sAll.sid . . . 6
get.scid.sid . . . 6
get.SrcAll.obs . . . 7
get.Src_cidAll.Src_id . . . 7
get.struc.all . . . 8
get.structure . . . 8
get.targets . . . 9
get.url.sid . . . 9
get.verbose.InCHIkey . . . 10
rUniChEMBL . . . 10

Index **11**

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	: ChEMBL target ID.

Details

get.appDrugs

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	: chemblid
type:	compound,target,assay. Default is compound.

Details

get.bioactivity.

Examples

```
get.bioactivity("CHEMBL12",type=compound)
get.bioactivity("CHEMBL240",type="target")
get.bioactivity("CHEMBL1217643",type=assay)
```

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

get.compound *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.compound(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. For the case of get.compound.list valid types are cansmi, substructure and similarity.

Details

get.compound

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

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	: source compound id
y	: source compound id

Details

get.mapping.full

Examples

```
get.mapping.full("3","1")
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	: chemblid

Details

get.moa

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	: InCHI Key

Details

get.sAll.InCHIKey

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	: chemblid
y	: source id

Details

get.sAll.sid

Examples

```
get.sAll.sid("CHEMBL12","1")
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	: Source compound id
y	: Source id

Details

get.scid.sid

Examples

```
get.scid.sid("CHEMBL12","1")
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:	source compound id
y:	to source id

Details

get.sAll.obs

get.Src_cidAll.Src_id *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	: InCHI Key

Details

get.src_id.InCHIKey

Examples

```
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	: chemblid
s	: source id (default is 1)

Details

get.struc.all

get.structure *Get structure*

Description

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

Usage

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

Arguments

X	: chemblid
s	: source id (default is 1)

Details

get.structure

Examples

```
get.structure("DB00321",2)
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`	: chemblid
`type:`	'chemblid' or 'uniprot'
`org:`	Species name like "Homo sapiens","Plasmodium falciparum" and etc.

Details

get.targets

Examples

```
get.targets("CHEMBL1862",type=chemblid)

get.targets("Q13936",type=uniprot)

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`	: source compound id
`y`	: source id
`z`	: to source id

Details

get.url.sid

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	: InCHI Key

Details

get.verbose.InCHIkey

rUniChEMBL *rUniChEMBL.*

Description

This package is built based on the web services of ChEMBL and Unichem

Index

get.appDrugs, 2
get.appDrugs-package *(get.appDrugs)*, 2
get.bioactivity, 2
get.bioactivity-package *(get.bioactivity)*, 2
get.cmp.sim, 3
get.cmp.sim-package *(get.cmp.sim)*, 3
get.cmp.substruct *(get.compound.substruct)*, 4
get.compound, 3
get.compound-package *(get.compound)*, 3
get.compound.substruct, 4
get.compound.substruct-package *(get.compound.substruct)*, 4
get.mapping.full, 4
get.mapping.full-package *(get.mapping.full)*, 4
get.moa, 5
get.moa-package *(get.moa)*, 5
get.sAll.InCHIKey, 5
get.sAll.InCHIKey-package *(get.sAll.InCHIKey)*, 5
get.sAll.obs *(get.SrcAll.obs)*, 7
get.sAll.sid, 6
get.sAll.sid-package *(get.sAll.sid)*, 6
get.scid.sid, 6
get.scid.sid-package *(get.scid.sid)*, 6
get.sid.InCHIKey *(get.Src_cidAll.Src_id)*, 7
get.Src_cidAll.Src_id, 7
get.Src_cidAll.Src_id-package *(get.Src_cidAll.Src_id)*, 7
get.SrcAll.obs, 7
get.SrcAll.obs-package *(get.SrcAll.obs)*, 7
get.struc.all, 8
get.struc.all-package *(get.struc.all)*, 8
get.structure, 8
get.structure-package *(get.structure)*, 8
get.targets, 9
get.targets-package *(get.targets)*, 9
get.url.sid, 9
get.url.sid-package *(get.url.sid)*, 9
get.verbose.InCHIkey, 10
get.verbose.InCHIkey-package *(get.verbose.InCHIkey)*, 10

rUniChEMBL, 10
rUniChEMBL-package *(rUniChEMBL)*, 10