Package 'rpubchem'

December 27, 2016

Version 1.5.10

Date 2016-12-24

Title An Interface to the PubChem Collection

Author Rajarshi Guha [aut, cre],
John Buonagurio [ctb]

Maintainer Rajarshi Guha <rajarshi.guha@gmail.com>

Depends R (>= 2.0.0)

Imports XML, car, RCurl, RJSONIO, data.table, iterators, itertools, stringr, fingerprint, base64enc, methods

Suggests testthat

License GPL (>= 2)

URL https://github.com/rajarshi/cdkr, https://pubchem.ncbi.nlm.nih.gov/

Description Access PubChem data (compounds, substance, assays) using R. Structural information is provided in the form of SMILES strings. It currently only provides access to a subset of the precalculated data stored by PubChem. Bio-assay data can be accessed to obtain descriptions as well as the actual data. It is also possible to search for assay ID's by keyword.

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2016-12-27 12:03:57

R topics documented:

decodeCACTVS 2
find.assay.id 3
get.aid.by.cid 4
get.assay 5

get.assay.desc 6
get.assay.summary 7
get.cid 8
get.cids.by.aid 9
get.sid 10
get.sid.list 11
get.sids.by.aid 12
get.synonyms 13

Index **14**

`decodeCACTVS` *Convert a Base64 encoded Pubchem 881-bit fingerprint to a* `fingerprint` *object*

Description

Pubchem computes 881-bit structural keys using the CACTVS toolkit, which are made available as Base64 encoded strings. This method converts the Pubchem string to a `fingerprint` object, which can be manipulated using the `fingerprint` package.

Usage

```
decodeCACTVS(cactvs)
```

Arguments

`cactvs`	A character string containing the Base64 encoded fingerprint

Value

A `fingerprint` object

See Also

`get.cid`

`find.assay.id` *Search for Assay ID's*

Description

PubChem allows one to obtain the ID's of bio-assays that match a search string. This function uses the Entrez interface to supply a query string and return the ID's of matching bio-assays

Usage

```
find.assay.id(query, quiet=TRUE)
```

Arguments

`query`	A character string containing the query
`quiet`	If `FALSE` the output is verbose

Value

A numeric vector containing the ID's that match the search query

Author(s)

Rajarshi Guha <rajarshi.guha@gmail.com>

See Also

`get.assay.desc`, `get.assay`

Examples

```
## Not run:
## find assay ID's related to yeast
aids <- find.assay.id('yeast')

## get the description of the first 10 assays
descs <- lapply( lapply(as.list(aids[1:10]), get.assay.desc), function(x)
x$assay.desc )

## End(Not run)
```

`get.aid.by.cid` *Get Assay ID Based on Compound Activity*

Description

This function allows you to identify PubChem assays in which a compound, specified by CID, has been tested in. The method uses the PubChem Power User Gateway (PUG) and as a result can be slow.

The function can be used to identify assays in which the CID is active, inactive or simply the assays in which it has been tested.

Usage

```
get.aid.by.cid(cid, type="tested", quiet=TRUE)
```

Arguments

`cid`	A single compound ID
`type`	What type of query should be performed. Valid values are 'active', 'inactive', 'tested'
`quiet`	If `FALSE`, output is verbose

Value

If the `type` argument was one of 'active', 'inactive', or 'tested' a numeric vector of assay IDs.

In case an invalid CID was specified or the query failed, `NULL` is returned.

Author(s)

Rajarshi Guha <rajarshi.guha@gmail.com>

See Also

`get.assay`

`get.assay` *Get a PubChem Bio-Assay*

Description

PubChem provides access to a number of bio-assays which are generally results obtained from High Throughput Screens (HTS). The number of observations in a given assay can be as high as 42000. This method allows one to obtain the assay data for a given assay ID. Assay ID's can be obtained using a text search using the `find.assay.id` function.

Usage

```
get.assay(aid, cid=NULL, sid=NULL, quiet=TRUE)
```

Arguments

`aid`	An assay ID
`cid`	A list of CID's
`sid`	A list of SID's
`quiet`	If `FALSE` the output is verbose

Details

The assay data are obtained for a variety of targets using a variety of techniques. As a result though each assay dataset contains a set of fixed fields, they can have additional fields.

If `cid` or `sid` is not specified the entire bioassay is retrieved. This can be time consuming for primary screening assays. If both arguments are specified, then `sid` is used in preference to `sid`.

Value

A data frame with the observations in the rows. The number of columns varies from assay to assay. Any assay will, however, have the following columns:

`PUBCHEM.SID` PubChem SID

`PUBCHEM.CID` PubChem CID

`PUBCHEM.ACTIVITY.OUTCOME`
Activity outcome

`PUBCHEM.ACTIVITY.SCORE`
Activity score, higher is more active

`PUBCHEM.ASSAYDATA.COMMENT`
Test result specific comment

The activity outcome field is provided as a numeric but is recoded as described in the PubChem documentation. The remaining fields are obtained by parsing the description file for the corresponding assay.

In addition to the usual attributes for a `data.frame` object this function adds some extra attributes:

- descriptionA short description of the assay
- commentsComments associated with the assay
- typesA named `list` where the names are the assay specific field names. Each element of the `list` is a 2-element vector containing the description of the field along with the units. In case the field is unitless the unit is NA

Author(s)

Rajarshi Guha <rajarshi.guha@gmail.com>

See Also

get.assay.desc, find.assay.id

`get.assay.desc` *Get An Assay Description*

Description

PubChem stores a number of pieces of information for each bio-assay. These include the description of the assay, related comments as well as type information (name, units, description) for the extra columns in the assay data.

This method accesses the description information and extracts a subset of that available.

Usage

```
get.assay.desc(aid)
```

Arguments

`aid`	A valid assay ID. This can be obtained using find.assay.id if not already known

Value

A `list` object with the following named components

`assay.desc`	A short description of the assay
`assay.comments`	A `list` of comments for the assay
`types`	A `matrix` with 3 columns. The first column is the name of the assay specific columns. The second column contains the descriptions of each assay specific column. The final column lists the units for each of the assay specific columns. In case an assay column is unitless, the value of the unit for that column is `NA`

Author(s)

Rajarshi Guha <rajarshi.guha@gmail.com>

See Also

find.assay.id, get.assay

`get.assay.summary` *Get a PubChem Bio-Assay Summary*

Description

Obtain the assay summary for a given assay id.

Usage

```
get.assay.summary(aid)
```

Arguments

`aid`	An assay ID

Details

The Pubchem assay summary has a number of sections, with each section seperated into chunks. The method will concatenate all chunks for a given section.

Value

A list with three elements

- Comment
- Protocol
- Description

Author(s)

Rajarshi Guha <rajarshi.guha@gmail.com>

See Also

get.assay, get.assay.desc, find.assay.id

get.cid *Get PubChem Compound Information*

Description

The PubChem compound collection stores a variety of information for each molecule. These include canonical SMILES, molecular properties, substance associations, synonyms etc.

This function will extract a subset of the molecular property information for a single CID.

Usage

```
get.cid(cid, quiet=TRUE)
```

Arguments

cid	A single numeric CID
quiet	If FALSE, output is verbose

Details

The method currently queries PubChem via the PUG REST interface. Since the method processes a single CID at a time, the user can parallelize processing. However, this is usually not recommended, at least in an unrestricted manner.

In addition, since the data.frame for each CID may have a different set of physical properties, it is recommended to either extract the common set of columns or else use something like bind_rows from the dplyr package to get a uniform data.frame if processing multiple CIDs

Value

A data.frame with at least 23 columns including the CID, IUPAC name, InChI and InChI key, canonical SMILES and a variety of molecular descriptors. In addition, a few physical properties are also included.

Author(s)

Rajarshi Guha <rajarshi.guha@gmail.com>

See Also

get.assay, get.sid, get.sid.list

Examples

```
## Not run:
cids <- c(5282108, 5282148, 91754124)
dat <- lapply(cids, get.cid)
dat <- dplyr::bind_rows(dat)
str(dat)

## End(Not run)
```

`get.cids.by.aid` *Retreive CID's for the given bioassay*

Description

Retreive CID's for the given bioassay

Usage

```
get.cids.by.aid(aid, quiet = TRUE)
```

Arguments

`aid`	The bioassay ID
`quiet`	If `TRUE` verbose output is provided

Value

A vector of CIDs

See Also

`get.sids.by.aid`, `get.sid.list`

Examples

```
get.cids.by.aid(2044)
```

`get.sid` *Get PubChem Substance Information*

Description

The PubChem substance collection stores a variety of information for each molecule. These include canonical SMILES, molecular properties, substance associations, synonyms etc.

This function will extract a subset of the molecular property information for one or more compound ID's

Usage

```
get.sid(sid, quiet=TRUE, from.file=FALSE)
```

Arguments

`sid`	A vector of one or more compound ID's
`quiet`	If `FALSE`, output is verbose
`from.file`	If `TRUE` then the first argument is considered to be the name of a file containing the XML data. If `FALSE` the first argument must be a sequence of compound ID's and the data will be downloaded from the PubChem FTP site

Details

Processing a large number of substance ID's can take a long time. For large numbers of SID's the resultant XML file can be many megabytes. This may take a long time to download. After download it takes approximate 20 sec to process a 23MB data file.

It should also be noted that the data files are downloaded using the R interface to Curl. In addition, the PubChem servers do not allow very large query URL's. This limits the number of substance ID's that can be directly pulled of the PubChem servers to about 1000

Value

A `data.frame` with 9 columns:

`SID`	The substance ID
`IUPACName`	The IUPAC name of the compound
`CanonicalSmiles`	The canonical SMILES for the compound
`MolecularWeight`	Molecular weight
`TotalFormalCharge`	The formal charge
`MolecularFormula`	The molecular formula

TPSA	Topological polar surface area
HeavyAtomCount	Heavy atom count
FormalCharge	Total formal charge
HydrogenBondDonor	Hydrogen bond donor count
HydrogenBondAcceptor	Hydrogen bond acceptor count

Author(s)

Rajarshi Guha <rajarshi.guha@gmail.com>

See Also

get.assay, get.cid, get.sid.list

---

get.sid.list *Get PubChem Substance ID's Associated With A Compound and Vice Versa*

Description

Each unique compound is associated with a number of substances. Given a CID it is possible to determine the associated substance ID's. Conversely given a SID it is useful to identify all CIDs that are associated with it

Usage

```
get.sid.list(cid, quiet=TRUE)
get.cid.list(sid, quiet=TRUE)
```

Arguments

cid	A single compound ID
sid	A single substance ID
quiet	If FALSE, output is verbose

Details

Even though PUG REST allows one to specify multiple input ID's these methods operate on single identifiers, allowing the user to parallelize multiple queries. In addition, this approach allows the package to cache results for individual input identifiers

Value

Depending on whether the input was a CID or SID, the return value is a numeric vector of SID's or a single numeric CID, respectively.

Author(s)

Rajarshi Guha <rajarshi.guha@gmail.com>

See Also

get.cid, get.sid, get.assay

---

get.sids.by.aid *Retreive SID's for the given bioassay*

Description

Retreive SID's for the given bioassay

Usage

```
get.sids.by.aid(aid, quiet = TRUE)
```

Arguments

aid	The bioassay ID
quiet	If TRUE verbose output is provided

Value

A vector of SIDs

See Also

get.cids.by.aid

Examples

```
get.sids.by.aid(2044)
```

get.synonyms *Get PubChem Compound ID's and Synonyms*

Description

PubChem allows one to obtain the compound ID's and synonyms of compounds that match a search string. This function uses the PubChem Power User Gateway (PUG) REST API to supply a character vector of one or more compound names and return the compound ID's and synonyms of matching compounds. Additional information on compounds can be obtained using the get.cid function.

Usage

```
get.synonyms(name, idtype = NULL, quiet=TRUE)
```

Arguments

name	A vector of one or more compound names
idtype	The default value of NULL indicates that name should be considered a compound name. Alternative values are inchikey or cid, in which case name should be an InChI key or a Pubchem CID
quiet	If FALSE, output is verbose

Details

Processing a large number of compounds can take a long time. The PUG REST API is not designed for very large volumes (millions) of requests. In order to avoid overloading the PubChem servers, this function is limited to 5 requests per second.

Value

A data.frame with 4 columns:

Name	The compound name provided
CID	The compound ID
Synonym	Synonyms associated with the compound ID
CAS	Logical indicating whether the synonym is a CAS RN

Author(s)

John Buonagurio <jbuonagurio@exponent.com>

See Also

get.cid

Index

∗Topic **programming**
find.assay.id, 3
get.aid.by.cid, 4
get.assay, 5
get.assay.desc, 6
get.assay.summary, 7
get.cid, 8
get.sid, 10
get.sid.list, 11
get.synonyms, 13

decodeCACTVS, 2

find.assay.id, 3, *5–7*

get.aid.by.cid, 4
get.assay, *3, 4*, 5, *7, 8*, *11, 12*
get.assay.desc, *3*, *6*, 6, *7*
get.assay.summary, 7
get.cid, *2*, 8, *11–13*
get.cid.list *(*get.sid.list*)*, 11
get.cids.by.aid, 9, *12*
get.sid, *8*, 10, *12*
get.sid.list, *8, 9*, *11*, 11
get.sids.by.aid, *9*, 12
get.synonyms, 13