Package: PubChemR (via r-universe)

September 10, 2024

Type Package

Title Interface to the 'PubChem' Database for Chemical Data Retrieval

Version 2.0

Description Provides an interface to the 'PubChem' database via the PUG REST <https://pubchem.ncbi.nlm.nih.gov/docs/pug-rest> and PUG View <https://pubchem.ncbi.nlm.nih.gov/docs/pug-view> services. This package allows users to automatically access chemical and biological data from 'PubChem', including compounds, substances, assays, and various other data types. Functions are available to retrieve data in different formats, perform searches, and access detailed annotations.

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

VignetteBuilder knitr

URL https://github.com/selcukorkmaz/PubChemR

Depends R (>= 3.6.0)

Imports dplyr, tibble, magrittr, stringr, tidyr, RJSONIO, httr, utils, RCurl, magick, rsvg, png, testthat

Suggests knitr, rmarkdown

Config/testthat/edition 3

Repository https://selcukorkmaz.r-universe.dev

RemoteUrl https://github.com/selcukorkmaz/pubchemr

RemoteRef HEAD

RemoteSha 7b57e70abb128c81cf12e2ddf2438e4ed0a3a767

Contents

`AIDs-SIDs-CIDs` *Assay, Compound, and Substance Identifiers*

Description

These functions are used to retrieve identification information for assays, substances, and compounds from the PubChem database.

Usage

```
AIDs(object, ...)

CIDs(object, ...)

SIDs(object, ...)

## S3 method for class 'PubChemInstance_AIDs'
AIDs(object, .to.data.frame = TRUE, ...)

## S3 method for class 'PubChemInstance_CIDs'
```

```
CIDs(object, .to.data.frame = TRUE, ...)

## S3 method for class 'PubChemInstance_SIDs'
SIDs(object, .to.data.frame = TRUE, ...)
```

Arguments

object	An object returned from a PubChem request, typically generated by functions such as get_cids, get_aids, and get_sids.
...	Additional arguments passed to other methods. Currently, these arguments have no effect.
.to.data.frame	a logical. If TRUE, returned object will be forced to be converted into a data.frame (or tibble). If failed to convert into a data.frame, a list will be returned with a warning. Be careful for complicated lists (i.e., many elements nested within each other) since it may be time consuming to convert such lists into a data frame.

Examples

```
# Retrieve Assay IDs
aids <- get_aids(identifier = c("aspirin", "caffeine"), namespace = "name")
AIDs(aids)

# Compound IDs
cids <- get_cids(identifier = c("aspirin", "caffein"), namespace = "name")
CIDs(cids)

# Substance IDs
sids <- get_sids(identifier = c("aspirin", "caffein"), namespace = "name")
SIDs(sids)
```

---

download *Download Content from PubChem and Save to a File*

Description

This function sends a request to PubChem to retrieve content in the specified format for a given identifier. It then writes the content to a specified file path.

Usage

```
download(
  filename = NULL,
  outformat,
  path,
  identifier,
  namespace = "cid",
```

```
  domain = "compound",
  operation = NULL,
  searchtype = NULL,
  overwrite = FALSE,
  options = NULL
)
```

Arguments

filename	a character string specifying the file name to be saved. If not specified, a default file name "file" is used.
outformat	A character string specifying the desired output format (e.g., "sdf", "json").
path	A character string specifying the path where the content should be saved.
identifier	A vector of positive integers (e.g. cid, sid, aid) or identifier strings (source, inchikey, formula). In some cases, only a single identifier string (name, smiles, xref; inchi, sdf by POST only).
namespace	Specifies the namespace for the query. For the 'compound' domain, possible values include 'cid', 'name', 'smiles', 'inchi', 'sdf', 'inchikey', 'formula', 'substructure', 'superstructure', 'similarity', 'identity', 'xref', 'listkey', 'fastidentity', 'fastsimilarity_2d', 'fastsimilarity_3d', 'fastsubstructure', 'fastsuperstructure', and 'fastformula'. For other domains, the possible namespaces are domain-specific.
domain	Specifies the domain of the query. Possible values are 'substance', 'compound', 'assay', 'gene', 'protein', 'pathway', 'taxonomy', 'cell', 'sources', 'sourcetable', 'conformers', 'annotations', 'classification', and 'standardize'.
operation	Specifies the operation to be performed on the input records. For the 'compound' domain, possible operations include 'record', 'property', 'synonyms', 'sids', 'cids', 'aids', 'assaysummary', 'classification', 'xrefs', and 'description'. The available operations are domain-specific.
searchtype	Specifies the type of search to be performed. For structure searches, possible values are combinations of 'substructure', 'superstructure', 'similarity', 'identity' with 'smiles', 'inchi', 'sdf', 'cid'. For fast searches, possible values are combinations of 'fastidentity', 'fastsimilarity_2d', 'fastsimilarity_3d', 'fastsubstructure', 'fastsuperstructure' with 'smiles', 'smarts', 'inchi', 'sdf', 'cid', or 'fastformula'.
overwrite	A logical value indicating whether to overwrite the file if it already exists. Default is FALSE.
options	Additional arguments.

Value

No return value. The function writes the content to the specified file path and prints a message indicating the save location.

Examples

```
# Download JSON file for the compound "aspirin" into "Aspirin.JSON"
# A folder named "Compound" will be created under current directory"
download(
  filename = "Aspirin",
  outformat = "json",
  path = "./Compound",
  identifier = "aspirin",
  namespace = "name",
  domain = "compound",
  overwrite = TRUE
)

# Remove downloaded files and folders.
file.remove("./Compound/Aspirin.json")
file.remove("./Compound/")
```

---

get_aids *Retrieve Assay IDs (AIDs) from PubChem*

Description

This function queries the PubChem database to retrieve Assay IDs (AIDs) based on a given identifier.

Usage

```
get_aids(
  identifier,
  namespace = "cid",
  domain = "compound",
  searchtype = NULL,
  options = NULL
)
```

Arguments

identifier	A vector of positive integers (e.g., cid, sid, aid) or identifier strings (source, inchikey, formula). In some cases, it may be a single identifier string (e.g., name, smiles, xref; inchi, sdf by POST only). Multiple elements can be included as a vector. See Notes for details.
namespace	Specifies the namespace for the query. For the 'compound' domain, possible values include 'cid', 'name', 'smiles', 'inchi', 'sdf', 'inchikey', 'formula', 'substructure', 'superstructure', 'similarity', 'identity', 'xref', 'listkey', 'fastidentity', 'fastsimilarity_2d', 'fastsimilarity_3d', 'fastsubstructure', 'fastsuperstructure', and 'fastformula'. For other domains, the possible namespaces are domain-specific.

domain	Specifies the domain of the query. Possible values are 'substance', 'compound', assay', 'gene', 'protein', 'pathway', 'taxonomy', 'cell', 'sources', 'sourcetable', 'conformers', 'annotations', 'classification', and 'standardize'.
searchtype	Specifies the type of search to be performed. For structure searches, possible values are combinations of 'substructure', 'superstructure', 'similarity', 'identity' with 'smiles', 'inchi', 'sdf', 'cid'. For fast searches, possible values are combinations of 'fastidentity', 'fastsimilarity_2d', 'fastsimilarity_3d', 'fastsubstructure', 'fastsuperstructure' with 'smiles', 'smarts', 'inchi', 'sdf', 'cid', or 'fastformula'.
options	Additional arguments to be passed to the PubChem Database API.

Value

An object of class 'PubChemInstance_AIDs', which is a list containing information retrieved from the PubChem database. Assay IDs can be extracted from the returned object using the getter function AIDs.

Note

To extract assay IDs from returned object, one may use AIDs function. See examples.

See Also

AIDs, get_pug_rest

Examples

```
# Request for multiple assays
# If assay identifier is unknown or incorrect, an error returns from PubChem Database
aids <- get_aids(
  identifier = c("aspirin", "ibuprofen", "rstudio"),
  namespace = "name"
)

print(aids)

# Return all Assay IDs.
AIDs(aids)
```

---

get_all_sources *Retrieve All Sources from PubChem*

Description

This function retrieves a list of all current depositors of substances or assays from PubChem.

Usage

```
get_all_sources(domain = "substance")
```

Arguments

domain	A character string specifying the domain for which sources ('substance', 'assay') are to be retrieved. Default is 'substance'.

Value

A character vector containing the names of all sources for the specified domain.

Examples

```
get_all_sources(
  domain = 'substance'
)
```

---

get_assays *Retrieve Assays from PubChem*

Description

This function sends a request to PubChem to retrieve assay data based on the specified parameters.

Usage

```
get_assays(
  identifier,
  namespace = "aid",
  operation = NULL,
  searchtype = NULL,
  options = NULL
)
```

Arguments

identifier	A vector of positive integers (e.g., cid, sid, aid) or identifier strings (source, inchikey, formula). In some cases, only a single identifier string (name, smiles, xref; inchi, sdf by POST only) can be provided. Multiple elements can be included as a vector. See Notes for details.
namespace	Specifies the namespace for the query. For the 'compound' domain, possible values include 'cid', 'name', 'smiles', 'inchi', 'sdf', 'inchikey', 'formula', 'substructure', 'superstructure', 'similarity', 'identity', 'xref', 'listkey', 'fastidentity', 'fastsimilarity_2d', 'fastsimilarity_3d', 'fastsubstructure', 'fastsuperstructure', and 'fastformula'. For other domains, the possible namespaces are domain-specific.

operation	The operation to be performed (default: NULL).
searchtype	The type of search to be performed (default: NULL).
options	Additional parameters (currently has no effect on the results).

Value

An object of class 'PubChemInstanceList' containing the information retrieved from the PubChem database.

Note

To extract information about a specific assay from the returned list, use the instance function.

Each assay may include information on several properties. Specific information from the assay can be extracted using the retrieve function. See examples.

See Also

retrieve, instance

Examples

```
# Retrieve a list of assays from the PubChem database
assays <- get_assays(
  identifier = c(1234, 7815),
  namespace = 'aid'
)

# Return assay information for assay ID '1234'
assay1234 <- instance(assays, "1234")
print(assay1234)

# Retrieve specific elements from the assay output
retrieve(assay1234, "aid")
```

---

get_cids *Retrieve Compound IDs (CIDs) from PubChem*

Description

This function sends a request to PubChem to retrieve Compound IDs (CIDs) for given identifier(s).

Usage

```
get_cids(
  identifier,
  namespace = "name",
  domain = "compound",
  searchtype = NULL,
  options = NULL
)
```

Arguments

identifier	A vector of positive integers (e.g., cid, sid, aid) or identifier strings (e.g., source, inchikey, formula). In some cases, only a single identifier string is required (e.g., name, smiles, xref; inchi, sdf by POST only). Multiple elements can be included as a vector. See Notes for details.
namespace	Specifies the namespace for the query. For the 'compound' domain, possible values include 'cid', 'name', 'smiles', 'inchi', 'sdf', 'inchikey', 'formula', 'substructure', 'superstructure', 'similarity', 'identity', 'xref', 'listkey', 'fastidentity', 'fastsimilarity_2d', 'fastsimilarity_3d', 'fastsubstructure', 'fastsuperstructure', and 'fastformula'. For other domains, the possible namespaces are domain-specific.
domain	Specifies the domain of the query. Possible values are 'substance', 'compound', 'assay', 'gene', 'protein', 'pathway', 'taxonomy', 'cell', 'sources', 'sourcetable', 'conformers', 'annotations', 'classification', and 'standardize'.
searchtype	Specifies the type of search to be performed. For structure searches, possible values include combinations of 'substructure', 'superstructure', 'similarity', 'identity' with 'smiles', 'inchi', 'sdf', or 'cid'. For fast searches, possible values include combinations of 'fastidentity', 'fastsimilarity_2d', 'fastsimilarity_3d', 'fastsubstructure', 'fastsuperstructure' with 'smiles', 'smarts', 'inchi', 'sdf', 'cid', or 'fastformula'.
options	Additional arguments to be passed to the PubChem Database API.

Value

An object of class 'PubChemInstance_CIDs', which is a list containing information retrieved from the PubChem database. Compound IDs can be extracted from the returned object using the CIDs function.

Note

To extract compoud IDs from returned object, one may use CIDs function. See examples.

See Also

CIDs, get_pug_rest

Examples

```
compound <- get_cids(
  identifier = "aspirin",
  namespace = "name"
)

print(compound)

# Extract compound IDs.
CIDs(compound)
```

`get_compounds` *Retrieve Compounds from PubChem*

Description

This function sends a request to the PubChem database to retrieve compound data based on specified parameters.

Usage

```
get_compounds(
  identifier,
  namespace = "cid",
  operation = NULL,
  searchtype = NULL,
  options = NULL
)
```

Arguments

`identifier`	A vector of positive integers (e.g., cid, sid, aid) or identifier strings (source, inchikey, formula). In some cases, a single identifier string (e.g., name, smiles, xref; inchi, sdf by POST only) is sufficient.
`namespace`	Specifies the namespace for the query. For the 'compound' domain, possible values include 'cid', 'name', 'smiles', 'inchi', 'sdf', 'inchikey', 'formula', 'substructure', 'superstructure', 'similarity', 'identity', 'xref', 'listkey', 'fastidentity', 'fastsimilarity_2d', 'fastsimilarity_3d', 'fastsubstructure', 'fastsuperstructure', and 'fastformula'. For other domains, the possible namespaces are domain-specific.
`operation`	The operation to be performed (default: NULL).
`searchtype`	Specifies the type of search to be performed. For structure searches, possible values are combinations of 'substructure', 'superstructure', 'similarity', 'identity' with 'smiles', 'inchi', 'sdf', 'cid'. For fast searches, possible values are combinations of 'fastidentity', 'fastsimilarity_2d', 'fastsimilarity_3d', 'fastsubstructure', 'fastsuperstructure' with 'smiles', 'smarts', 'inchi', 'sdf', 'cid', or 'fastformula'.

options Additional parameters passed to get_json.

Value

An object of class 'PubChemInstanceList' and 'PC_Compounds' containing compound information from the PubChem database.

See Also

retrieve, instance

Examples

```
compound <- get_compounds(
  identifier = c("aspirin", "ibuprofen", "rstudio"),
  namespace = "name"
)

print(compound)

# Return results for selected compound.
instance(compound, "aspirin")
instance(compound, "rstudio")
# instance(compound, "unknown"). # returns error.

# Extract compound properties for the compound "aspirin".
# Use the 'retrieve()' function to extract specific slots from the compound list.
retrieve(instance(compound, "aspirin"), "props")
```

---

get_properties *Retrieve Compound Properties from PubChem*

Description

This function sends a request to PubChem to retrieve compound properties based on the specified parameters.

Usage

```
get_properties(
  properties = NULL,
  identifier,
  namespace = "cid",
  searchtype = NULL,
  options = NULL,
  propertyMatch = list(.ignore.case = FALSE, type = "contain")
)
```

```
property_map(
  x,
  type = c("match", "contain", "start", "end", "all"),
  .ignore.case = TRUE,
  ...
)
```

Arguments

properties	A character vector specifying the properties to be retrieved. It is ignored if all available properties are requested from PubChem. See examples.
identifier	A vector of positive integers (e.g., CID, SID, AID) or identifier strings (e.g., source, InChIKey, formula). In some cases, only a single identifier string is allowed (e.g., name, SMILES, xref; InChI, SDF by POST only).
namespace	Specifies the namespace for the query. For the 'compound' domain, possible values include 'cid', 'name', 'smiles', 'inchi', 'sdf', 'inchikey', 'formula', 'substructure', 'superstructure', 'similarity', 'identity', 'xref', 'listkey', 'fastidentity', 'fastsimilarity_2d', 'fastsimilarity_3d', 'fastsubstructure', 'fastsuperstructure', and 'fastformula'. For other domains, the possible namespaces are domain-specific.
searchtype	Specifies the type of search to be performed. For structure searches, possible values are combinations of 'substructure', 'superstructure', 'similarity', 'identity' with 'smiles', 'inchi', 'sdf', 'cid'. For fast searches, possible values are combinations of 'fastidentity', 'fastsimilarity_2d', 'fastsimilarity_3d', 'fastsubstructure', 'fastsuperstructure' with 'smiles', 'smarts', 'inchi', 'sdf', 'cid', or 'fastformula'.
options	Additional arguments passed to `get_json`.
propertyMatch	A list containing the arguments passed to the property_map function. See examples of the `property_map()` function.
x	A character vector of compound properties. The property_map function will search for each property provided here within the available properties. The search can be customized using the `type` argument. This argument is ignored if `type = "all"`.
type	Defines how to search within the available properties. The default is "match". See Notes for details.
.ignore.case	A logical value. If TRUE, the pattern match ignores case letters. This argument is ignored if `type = "all"`. The default is TRUE.
...	Other arguments. Currently, these have no effect on the function's return.

Value

An object of class "PubChemInstanceList" containing all the properties of the requested compounds.

Note

Property Map:: `property_map()` is not used to request properties directly from the PubChem database. This function is intended to list the available compound properties that can be requested from PubChem. It has flexible options to search properties from the available property list of the PubChem database. The output of property_map is used as the `property` input in the get_properties function. This function may be practically used to request specific properties across a range of compounds. See examples for usage.

Examples

```
# Isomeric SMILES of the compounds
props <- get_properties(
  properties = c("MolecularWeight", "MolecularFormula", "InChI"),
  identifier = c("aspirin", "ibuprofen", "caffeine"),
  namespace = "name"
)

# Properties for a selected compound
instance(props, "aspirin")
retrieve(props, .which = "aspirin", .slot = NULL)
retrieve(instance(props, "aspirin"), .slot = NULL)

# Combine properties of all compounds into a single data frame (or list)
retrieve(props, .combine.all = TRUE)

# Return selected properties
retrieve(props, .combine.all = TRUE,
  .slot = c("MolecularWeight", "MolecularFormula"))

# Return properties for the compounds in a range of CIDs
props <- get_properties(
  properties = c("mass", "molecular"),
  identifier = 2244:2255,
  namespace = "cid",
  propertyMatch = list(
    type = "contain"
  )
)

retrieve(props, .combine.all = TRUE, .to.data.frame = TRUE)

# Return all available properties of the requested compounds
props <- get_properties(
  properties = NULL,
  identifier = 2244:2245,
  namespace = "cid",
  propertyMatch = list(
    type = "all"
  )
)

retrieve(props, .combine.all = TRUE)
```

```
#### EXAMPLES FOR property_map() ####
# List all available properties:
property_map(type = "all")

# Exact match:
property_map("InChI", type = "match")
property_map("InChi", type = "match",
  .ignore.case = TRUE) # Returns no match. Ignores '.ignore.case'

# Match at the start/end:
property_map("molecular", type = "start", .ignore.case = TRUE)
property_map("mass", type = "end", .ignore.case = TRUE)

# Partial match with multiple search patterns:
property_map(c("molecular", "mass", "inchi"),
  type = "contain", .ignore.case = TRUE)
```

`get_pug_rest` *Retrieve Data from PubChem PUG REST API*

Description

This function sends a request to the PubChem PUG REST API to retrieve various types of data for a given identifier. It supports fetching data in different formats and allows saving the output.

Usage

```
get_pug_rest(
  identifier = NULL,
  namespace = "cid",
  domain = "compound",
  operation = NULL,
  output = "JSON",
  searchtype = NULL,
  property = NULL,
  options = NULL,
  save = FALSE,
  dpi = 300,
  path = NULL,
  file_name = NULL,
  ...
)
```

Arguments

`identifier`	A vector of identifier for the query, either numeric or character.

namespace	A character string specifying the namespace for the request. Default is 'cid'.
domain	A character string specifying the domain for the request. Default is 'compound'.
operation	An optional character string specifying the operation for the request.
output	A character string specifying the output format. Possible values are 'SDF', 'JSON', 'JSONP', 'CSV', 'TXT', and 'PNG'. Default is 'JSON'.
searchtype	An optional character string specifying the search type.
property	An optional character string specifying the property for the request.
options	A list of additional options for the request.
save	A logical value indicating whether to save the output as a file or image. Default is FALSE.
dpi	An integer specifying the DPI for image output. Default is 300.
path	description
file_name	a character of length 1. Define the name of file (without file extension) to save. If NULL, default file name is set as "files_downloaded".
...	description

Value

Depending on the output format, this function returns different types of content: JSON or JSONP format returns parsed JSON content. CSV format returns a data frame. TXT format returns a table. SDF returns SDF file of requested identifier. PNG format returns an image object or saves an image file.

Examples

```
result <- get_pug_rest(identifier = "2244", namespace = "cid", domain = "compound", output = "JSON")
```

---

get_pug_view *Retrieve PUG View Data from PubChem*

Description

This function sends a request to the PubChem PUG View API to retrieve various types of data for a given identifier. It supports fetching annotations, QR codes, and more, with options for different output formats including JSON and SVG.

Usage

```
get_pug_view(
  annotation = NULL,
  identifier = NULL,
  domain = "compound",
  output = "JSON",
```

```
  heading = NULL,
  headingType = NULL,
  page = NULL,
  qrSize = "short",
  save = FALSE
)
```

Arguments

annotation	A character string specifying the type of annotation to retrieve.
identifier	A single identifier for the query, either numeric or character.
domain	A character string specifying the domain for the request. Default is 'compound'.
output	A character string specifying the output format. Possible values are 'JSON' and 'SVG'. Default is 'JSON'.
heading	An optional character string for specifying a heading in the request.
headingType	An optional character string for specifying a heading type in the request.
page	An optional character string for specifying a page number in the request.
qrSize	A character string specifying the size of the QR code. Possible values are 'short' and 'long'. Default is 'short'.
save	A logical value indicating whether to save the output. Default is FALSE.

Value

Depending on the output format, this function returns different types of content: JSON or JSONP format returns parsed JSON content. SVG format returns an image object. For QR codes, it returns an image object or saves a PNG file.

Examples

```
get_pug_view(identifier = "2244", annotation = "linkout", domain = "compound")
```

---

get_sdf *Retrieve/Save SDF Data from PubChem*

Description

This function sends a request to PubChem to retrieve data in SDF format based on the specified parameters. It then saves the retrieved data as an SDF file in the current working directory (or into the system-specific temporary folder).

Usage

```
get_sdf(
  identifier,
  namespace = "cid",
  domain = "compound",
  operation = NULL,
  searchtype = NULL,
  path = NULL,
  file_name = NULL,
  options = NULL
)
```

Arguments

identifier	A character or numeric value specifying the identifier for the request.
namespace	A character string specifying the namespace for the request. Default is 'cid'.
domain	A character string specifying the domain for the request. Default is 'compound'.
operation	An optional character string specifying the operation for the request.
searchtype	An optional character string specifying the search type.
path	A string indicating the path to the folder where the SDF files will be saved. Default is NULL (i.e., saves the file into a temporary folder).
file_name	A string. File name for the downloaded SDF file. If NULL, "file" is used as the file name. Default is NULL.
options	Additional parameters to be passed to the request.

Value

NULL. The function saves the retrieved data as an SDF file in the current working directory and prints a message indicating the file's location.

Examples

```
get_sdf(
  identifier = "aspirin",
  namespace = "name",
  path = NULL
)
```

get_sids *Retrieve Substance IDs (SIDs) from PubChem*

Description

This function sends a request to PubChem to retrieve Substance IDs (SIDs) for a given identifier.

Usage

```
get_sids(
  identifier,
  namespace = "cid",
  domain = "compound",
  searchtype = NULL,
  options = NULL
)
```

Arguments

identifier	A numeric or character vector specifying the identifiers for the request.
namespace	A character string specifying the namespace for the request. Default is 'cid'.
domain	A character string specifying the domain for the request. Default is 'compound'.
searchtype	A character string specifying the search type. Default is NULL.
options	Additional arguments passed to the PubChem request.

Value

An object of class 'PubChemInstance_SIDs', which is a list containing information retrieved from the PubChem database. Substance IDs can be extracted from the returned object using the SIDs function.

Examples

```
result <- get_sids(
  identifier = c("aspirin", "ibuprofen"),
  namespace = "name"
)

# Extract substance IDs of all compounds
SIDs(result)
```

get_substances *Retrieve Substances from PubChem*

Description

This function sends a request to PubChem to retrieve substance data based on the specified parameters.

Usage

```
get_substances(
  identifier,
  namespace = "sid",
  operation = NULL,
  searchtype = NULL,
  options = NULL
)
```

Arguments

identifier	A character or numeric vector specifying the identifiers for the request.
namespace	A character string specifying the namespace for the request. Default is 'sid'.
operation	Specifies the operation to be performed on the input records. For the 'compound' domain, possible operations include 'record', 'property', 'synonyms', 'sids', 'cids', 'aids', 'assaysummary', 'classification', 'xrefs', and 'description'. The available operations are domain-specific.
searchtype	Specifies the type of search to be performed. For structure searches, possible values are combinations of 'substructure', 'superstructure', 'similarity', 'identity' with 'smiles', 'inchi', 'sdf', 'cid'. For fast searches, possible values are combinations of 'fastidentity', 'fastsimilarity_2d', 'fastsimilarity_3d', 'fastsubstructure', 'fastsuperstructure' with 'smiles', 'smarts', 'inchi', 'sdf', 'cid', or 'fastformula'.
options	Additional parameters passed to PubChem request.

Value

An object of class 'PubChemInstanceList' containing all the substance information of requested compounds.

Examples

```
subs <- get_substances(
  identifier = c("aspirin", "ibuprofen"),
  namespace = "name"
)

instance(subs, "aspirin")
```

```
retrieve(instance(subs, "aspirin"), "source")
```

`get_synonyms` *Retrieve Synonyms from PubChem*

Description

This function sends a request to PubChem to retrieve synonyms for a given identifier. It returns a list of synonyms corresponding to the provided identifier.

Usage

```
get_synonyms(
  identifier,
  namespace = "cid",
  domain = "compound",
  searchtype = NULL,
  options = NULL
)
```

Arguments

`identifier`	A character or numeric value specifying the identifier for the request.
`namespace`	A character string specifying the namespace for the request. Default is 'cid'.
`domain`	A character string specifying the domain for the request. Default is 'compound'.
`searchtype`	A character string specifying the search type. Default is NULL.
`options`	Additional arguments passed to PubChem request.

Value

An object of class 'PubChemInstance_Synonyms', which is a list containing information retrieved from the PubChem database. Synonyms data can be extracted from the returned object using the synonyms function.

Examples

```
syns <- get_synonyms(
  identifier = "aspirin",
  namespace = "name"
)

synonyms(syns)
```

instance *Retrieve Information for Requested Instances*

Description

This function extracts the results of a PubChem instance from an `object`. It is designed to retrieve information about a compound from a comprehensive list where multiple elements (such as assay, compound, etc.) are requested.

Usage

```
instance(object, ...)

## S3 method for class 'PubChemInstanceList'
instance(object, .which = NULL, ...)
```

Arguments

`object`	An object of class `'PubChemInstanceList'` returned from a PubChem request.
`...`	Additional arguments passed to other methods. Currently, these have no effect.
`.which`	A string specifying which instance's results to return. If NULL, the results of the first instance in the `object` are returned. The default value is NULL.

Examples

```
compounds <- get_compounds(
  identifier = c("aspirin", "ibuprofen"),
  namespace = "name"
)

instance(compounds)  # Returns the results for "aspirin"
instance(compounds, "ibuprofen")
```

pubChemData *Retrieve Raw Data from PUG REST Object*

Description

A short description...

Usage

```
pubChemData(object, ...)

## S3 method for class 'PugRestInstance'
pubChemData(object, ...)
```

Arguments

object	an object of class 'PugRestInstance' returned from get_pug_rest function.
...	additional arguments. Currently has no effect on results.

Value

a vector, list, or data.frame containing the raw data retrieved from Pub Chem database through PUG REST API.

See Also

get_pug_rest

Examples

```
result <- get_pug_rest(identifier = "2244", namespace = "cid", domain = "compound", output = "JSON")
pubChemData(result)
```

PubChemR-classes PubChemInstanceList *and* PubChemInstance *Classes*

Description

The PubChemInstanceList object is a superclass returned by a request for compound(s) from the PubChem Database, such as the output from get_compounds, get_assays, etc.

The PubChemInstance object is another superclass for a PubChem instance, such as an assay, compound, substance, etc. These instances are nested within the results slot of a PubChemInstanceList object. Similar to PubChemInstanceList, the PubChemInstance also contains the same slots as described below. For more details, see instance.

Slots

results: A list containing elements of each of the requested compounds, assays, substances, etc.

request_args: A list containing the input arguments of a PubChem request.

success: A logical value indicating whether the request was successfully completed (TRUE) or not (FALSE).

error: A list detailing any errors encountered during the request, if applicable.

Note

There is no constructor function for the PubChemInstanceList or PubChemInstance classes. These objects are constructed within related functions and returned as the output of PubChem requests.

There are several subclasses defined under the PubChemInstanceList and PubChemInstance superclasses. The PubChem API returns request results in a list; however, each request may have a different list structure and/or items within the returned list. Therefore, we have defined subclasses to make generic functions compatible with any PubChem request, such as assays, instances, substances, etc. These subclasses may include PC_Compounds, PC_Substance, PC_Properties, PubChemInstance_AIDs, PubChemInstance_SIDs, PubChemInstance_CIDs, PubChemInstance_Synonyms, and PubChemInstance_Substances.

Most of the defined subclasses have similar slots as described above. However, some classes may have additional slots not described here. Please refer to the contents of the returned object for more details.

PugView-classes *Classes for Pug View Request*

Description

The Pug View API of PubChem database returns more detailed information about a PubChem request, such as assays, compounds, substances, etc. A super-class PugViewInstance is defined, which is returned from get_pug_view function. This class has slots detailed below.

Slots

results: A list containing elements of each of the requested compounds, assays, substances, etc.

request_args: A list containing the input arguments of a PubChem request.

success: A logical value indicating whether the request was successfully completed (TRUE) or not (FALSE).

error: A list detailing any errors encountered during the request, if applicable.

Note

Pug View API returns many section about the requested instance, which includes detailed information from PubChem database. There may be many nested sections, where each contains details about different features of the instance requested. These sections can be listed via sectionList function.

Other classes, called PugViewSectionList and PugViewSection, are defined to control the outputs of available sections and sub-sections returned from get_pug_view. See related functions for details.

See Also

section, sectionList

request_args *Retrieve Function Inputs*

Description

This function retrieves the input arguments from a specified PubChem database request object.

Usage

```
request_args(object, .which = NULL, ...)
```

Arguments

object	An object returned from related request functions of the PubChem database.
.which	A string specifying which argument's content to retrieve from object. If NULL, all function inputs will be returned.
...	Additional arguments. These have no effect on the returned outputs and are included for compatibility with S3 methods in the PubChemR package.

Value

A list or string vector containing the options used in the function call.

Examples

```
request <- get_cids("aspirin", namespace = "name")

request_args(request, "identifier")
request_args(request)
```

retrieve *Retrieve Information from PubChem Instances*

Description

This generic function extracts a specific slot from a PubChem instance.

Usage

```
retrieve(object, ...)

## S3 method for class 'PubChemInstance'
retrieve(object, .slot = NULL, .to.data.frame = TRUE, .verbose = FALSE, ...)

## S3 method for class 'PubChemInstanceList'
retrieve(
  object,
  .which = NULL,
  .slot = NULL,
  .to.data.frame = TRUE,
  .combine.all = FALSE,
  ...
)

## S3 method for class 'PC_Substance'
retrieve(
  object,
  .slot = NULL,
  .idx = 1,
  .to.data.frame = TRUE,
  .verbose = FALSE,
  ...
)

## S3 method for class 'PugViewInstance'
retrieve(object, .slot = NULL, .to.data.frame = TRUE, ...)

## S3 method for class 'PugViewSection'
retrieve(object, .slot = NULL, .to.data.frame = FALSE, ...)
```

Arguments

`object`	An object returned from a PubChem request.
`...`	Additional arguments passed to other methods.
`.slot`	A string specifying which slot to return. Should not be NULL or length of >1 with some exceptions. See the notes for details.
`.to.data.frame`	A logical value. If TRUE, the returned object will be converted into a data.frame (or tibble). If conversion to a data.frame fails, a list will be returned with a warning. Be cautious with complex lists (i.e., many elements nested within each other) as it may be time-consuming to convert such lists into a data frame. Additionally, `.to.data.frame` is ignored in specific scenarios.
`.verbose`	A logical value. Should the resulting object be printed to the R console? If TRUE, the object is returned invisibly and the output is printed nicely to the R console. This option may not be available for some slots (or classes). See Notes/Details.

.which	A character value. This is the identifier of the PubChem request that will be extracted from the complete list. It is ignored if .combine.all = TRUE.
.combine.all	a logical value. If TRUE, the properties of all requested instances are combined into a single data frame (or a list if .to.data.frame = FALSE).
.idx	An integer indicating which substance result should be returned. A PubChem request may return multiple substances in the output. .idx specifies the index of the substance to be extracted from the complete list.

Details on 'PugViewInstance' **and** 'PugViewSection'

The PugView API returns a detailed list related to PubChem requests. The 'Section' slot in this list is structured into a sub-class called 'PugViewSection'. This object contains information organized through several sections (or sub-sections), which can be retrieved using *section-specific* functions such as section and sectionList.

The function argument .to.data.frame is ignored if the "Section" slot is being extracted from the complete list. For other slots, .to.data.frame is considered as usual. See examples for usage.

Note

If the object is from the 'PC_Properties' class, the .slot can be defined as NULL. If .slot = NULL, retrieve() will return all available properties. If 'object' is of class other than 'PC_Properties', .slot should be length of 1.

Extracting multiple slots.: In some cases, it may be practical to extract multiple slots from 'object'. For example, one may wish to extract properties from the output of get_properties by running the functions in a loop. See codes below for a practical example:

```
library(dplyr)

props <- get_properties(
  properties = c("MolecularWeight", "MolecularFormula", "HBondDonorCount",
                 "HBondAcceptorCount", "InChIKey", "InChI"),
  identifier = 2244,
  namespace = "cid",
  propertyMatch = list(
    .ignore.case = TRUE,
    type = "contain"
  )
)

bind_columns <- function(x, ...){
  part1 <- x[[1]][ ,"Identifier"]
  part2 <- lapply(x, "[", 2)
    bind_cols()

  bind_cols(part1, part2)
}

propsToExtract <- c("MolecularWeight", "MolecularFormula", "HBondDonorCount")
```

```
tmp <- lapply(propsToExtract, retrieve, object = props, .which = "2244")
bind_columns(tmp)
```

Use of the '.verbose' argument: `retrieve` returns output silently (invisibly) when `.verbose = TRUE`. However, the function behaves differently under the following scenarios:

- `.verbose` is ignored if `.combine.all = TRUE`. The output is returned silently.
- `.verbose` is ignored if the requested slot is not printable to the R console because it is too complicated to print.

Examples

```
compounds <- get_compounds(
  identifier = c("aspirin", "ibuprofen", "rstudio"),
  namespace = "name"
 )

# Extract information for "aspirin"
aspirin <- instance(compounds, "aspirin")
# print(aspirin)

# Extract a specific slot from the "aspirin" compound.
retrieve(aspirin, "props", .to.data.frame = TRUE)

# Examples (PubChemInstanceList)
retrieve(compounds, "aspirin", "props", .to.data.frame = TRUE)

# Verbose Assay References to R Console
assays <- get_assays(identifier = c(1234, 7815), namespace = "aid")

instance(assays, "7815")
retrieve(assays, "7815", "xref", .verbose = TRUE)

# Print assay protocol to R console (if available)
# Note that it may be too long to print for some assays.
# retrieve(assays, "1234", "protocol", .verbose = TRUE)

# No protocol is available for assay "1234".
# retrieve(assays, "7815", "protocol", .verbose = TRUE)

# Ignores ".verbose" and ".which" if ".combine.all = TRUE".
retrieve(assays, .slot = "xref", .verbose = TRUE, .combine.all = TRUE)

### PUG VIEW EXAMPLES ###
pview <- get_pug_view(identifier = "2244", annotation = "data", domain = "compound")

# PugViewSectionList object.
# This object contains all the section information related to the PubChem request.
sect <- retrieve(pview, .slot = "Section")
print(sect)

retrieve(pview, .slot = "RecordType", .to.data.frame = TRUE)
```

`section` *Extract Sections from Pug View Request*

Description

`section` returns section details from a Pug View request.

Usage

```
section(object, ...)

## S3 method for class 'PugViewInstance'
section(object, .id = "S1", .verbose = FALSE, ...)

## S3 method for class 'PugViewSectionList'
section(object, .id = "S1", .verbose = FALSE, ...)

## S3 method for class 'PugViewSection'
section(object, .id = "S1", .verbose = FALSE, ...)
```

Arguments

`object`	an object returned from get_pug_view.
`...`	other arguments. Currently has no effect on the outputs. Can be ignored.
`.id`	A character value that corresponds to the ID of a specific section. Detailed information about the section with the given section ID will be returned. If NULL, the first section (i.e., "S1") is returned. If there is no section under `object`, it returns NULL with a warning. See Note/Details for more information.
`.verbose`	A logical value. Should the resulting object be printed to the R console? If TRUE, the object is returned invisibly and the output is printed nicely to the R console. This option may not be available for some slots (or classes). See Notes/Details.

Note

Sections in a Pug View Request: A Pug View Request returns a detailed list from the PubChem database. This list may include data under many nested sections, each corresponding to a different property structured within further nested sections. The complicated structure of the returned object makes it impossible to print all information to the R console at once. Therefore, it is recommended to print sections selectively. Furthermore, one may navigate through the nested sections using the section function. See Examples.

Use the sectionList function to list available sections (or subsections of a section) of a Pug View request and related section IDs.

Use of `'.verbose'` to Print Section Details: It is possible to print section details to the R console. If `.verbose = TRUE`, the resulting object is returned invisibly and a summary of section

details is printed to the R console. This might be useful to navigate through nested sections and sequentially print multiple sections to the R console. For example, consider following command:

```
> section(section(request, "S1", .verbose = TRUE), "S3", .verbose = TRUE)
```

This command will print section "S1" and the subsection "S3" located under "S1" to the R console. One may navigate through sections under other sections, similar to exploring dreams within dreams as depicted in the exceptional movie **Inception**. (**SPOILER WARNING!!**) However, be careful not to get lost or stuck in the dreams!! Also, note that this strategy works only if `.verbose = TRUE` for all sections and/or subsections.

See Also

sectionList

Examples

```
# Pug View request for the compound "aspirin (CID = 2244)".
pview <- get_pug_view(identifier = "2244", annotation = "data", domain = "compound")

section(pview, "S1")
section(pview, "S1", .verbose = TRUE)

# List all available sections
sectionList(pview)

# Subsections under the section "S1"
sectionList(section(pview, "S1"))

# Print multiple sections
# section(section(pview, "S1", .verbose = TRUE), "S3", .verbose = TRUE)
```

---

sectionList *List Available Section/Subsections*

Description

This function may be used to list available sections (or subsections) of a PubChem request returned from get_pug_view. It is useful when one wants to extract a specific section (or subsection) from PubChem request. It supports patteern-specific searches within sections. See Detail/Note below for more information.

Usage

```
sectionList(object, ...)

## S3 method for class 'PugViewInstance'
sectionList(object, ...)
```

```
## S3 method for class 'PugViewSectionList'
sectionList(
  object,
  .pattern = NULL,
  .match_type = c("contain", "match", "start", "end"),
  ...
)

## S3 method for class 'PugViewSection'
sectionList(
  object,
  .pattern = NULL,
  .match_type = c("contain", "match", "start", "end"),
  ...
)
```

Arguments

`object`	an object of PubChem request, generally returned from get_pug_view.
`...`	other arguments. Currently has no effect on the outputs. Can be ignored.
`.pattern`	a character vector. Each text pattern given here will be searched within Pug View sections by using the pattern matching strategy defined with `.match_type`. If not specified or NULL, all available sections will be returned.
`.match_type`	a string. How should search patterns (i.e., `.pattern`) matched with section names? Available options are "contain", "match", "start", and "end" which can be used for partial and/or exact pattern matching. Default is "contain". See Details below for more information.

Details

Pattern matching is used to filter sections that match user-defined patterns. It is useful when there are more sections than allowed to print R console. In such situations, it may be reasonable to print a subset of all section list to R console that meets search criteria. There are several pattern matching methods as described below

- **Partial Matching** ("contain", "start", "end"): Returns the section names that contains or starts/ends by given text patterns.
- **Exact Matching** ("match"): Returns the section names that exactly matches given text patterns.

See Also

section

Examples

```
pview <- get_pug_view(identifier = "2244", annotation = "data", domain = "compound")

# List all section names
```

```
sectionList(pview)

# Pattern-matched section names
sectionList(pview, .pattern = c("safety", "chemical"), .match_type = "contain")
sectionList(pview, .pattern = "safety", .match_type = "match")
sectionList(pview, .pattern = "properties", .match_type = "end")

# Use section IDs to extract section data from Pug View request
section(pview, "S12") # Safety and Hazards
```

---

synonyms *Getter function for 'Synonyms'*

Description

Extracts synonym data from a PubChem request using the function get_synonyms.

Usage

```
synonyms(object, ...)

## S3 method for class 'PubChemInstance_Synonyms'
synonyms(object, .to.data.frame = TRUE, ...)
```

Arguments

`object`	An object of class `'PubChemInstance_Synonyms'`.
`...`	Additional arguments passed to other methods. Currently, these have no effect.
`.to.data.frame`	a logical. If TRUE, returned object will be forced to be converted into a data.frame (or tibble). If failed to convert into a data.frame, a list will be returned with a warning. Be careful for complicated lists (i.e., many elements nested within each other) since it may be time consuming to convert such lists into a data frame.

Value

A `data.frame` (or `list`) object containing the synonym data.

Examples

```
syns <- get_synonyms(identifier = c("aspirin", "caffeine"), namespace = "name")
synonyms(syns)
```

Index