Package: dbparser (via r-universe)

October 10, 2026

Title Drugs Databases Parser

Version 2.2.3.9000

Description This tool is for parsing public drug databases such as 'DrugBank' XML database <https://go.drugbank.com/>. The parsed data are then returned in a proper 'R' object called 'dvobject'.

License MIT + file LICENSE

Encoding UTF-8

Imports data.table, dplyr, progress, purrr, tibble, tools, utils, XML

Suggests canvasXpress, knitr, rmarkdown, testthat, tidyr

VignetteBuilder knitr

URL https://docs.ropensci.org/dbparser/, https://github.com/ropensci/dbparser

BugReports https://github.com/ropensci/dbparser/issues

Depends R (>= 3.5)

Config/roxygen2/version 8.0.0

Config/pak/sysreqs libxml2-dev

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

Date/Publication 2026-10-10 19:07:07 UTC

RemoteUrl https://github.com/ropensci/dbparser

RemoteRef cran

RemoteSha 94ae3dd888f2a349b5ff8f41fe04803be463c987

Contents

add_database_info 2
cett_nodes_options 3
drug_node_options 3
get_genes_for_phenotype 4

merge_drugbank_hpo 5
merge_drugbank_onsides 6
merge_drugbank_twosides 7
parseDrugBank 8
parseHPO 9
parseOnSIDES 10
parseTWOSIDES 11
references_node_options 12
search_phenotypes 13
show_dvobject_metadata 13
subset_drugbank_dvobject 14
subset_onsides_dvobject 15

Index **16**

`add_database_info` *add_database_info Assign passed databases db metadata to passed dvobject*

Description

add_database_info Assign passed databases db metadata to passed dvobject

Usage

```
add_database_info(
  dvobject,
  db_type = "DrugBank",
  db_version = NULL,
  db_exported_date = NULL
)
```

Arguments

`dvobject`	dvobject to assign metadata to it
`db_type`	database type (defualt="DrugBank")
`db_version`	database version as string
`db_exported_date`	database official export date

Value

dvobject

See Also

Other utility: `get_genes_for_phenotype()`, `search_phenotypes()`, `show_dvobject_metadata()`, `subset_drugbank_dvobject()`, `subset_onsides_dvobject()`

cett_nodes_options *returns carriers, enzymes,targets and transporters node valid options.*

Description

returns carriers, enzymes,targets and transporters node valid options.

Usage

```
cett_nodes_options()
```

Value

list of CETT valid options

See Also

Other parsers: drug_node_options(), parseDrugBank(), parseHPO(), parseOnSIDES(), parseTWOSIDES(), references_node_options()

drug_node_options *returns drug node valid options.*

Description

returns drug node valid options.

Usage

```
drug_node_options()
```

Value

list of drug valid options

See Also

Other parsers: cett_nodes_options(), parseDrugBank(), parseHPO(), parseOnSIDES(), parseTWOSIDES(), references_node_options()

get_genes_for_phenotype *Get Genes Associated with a Phenotype*

Description

Returns the genes associated with a phenotype, given either its HPO identifier (e.g. `"HP:0001250"`) or its name.

Usage

```
get_genes_for_phenotype(hpo_db, phenotype, exact = FALSE)
```

Arguments

hpo_db	An HPO dvobject returned by parseHPO.
phenotype	Character. An HPO identifier or a phenotype name.
exact	Logical. Used when phenotype is a name. If TRUE, the name must match exactly (ignoring case). If FALSE (default), any phenotype whose name contains phenotype matches.

Value

A tibble with columns gene_symbol, entrez_id, hpo_id, phenotype_name and association_weight, ordered by gene symbol.

See Also

Other utility: add_database_info(), search_phenotypes(), show_dvobject_metadata(), subset_drugbank_dvobject
subset_onsides_dvobject()

Examples

```
hpo_db <- parseHPO(system.file("extdata", "hpo_gene_attribute_edges.txt",
                               package = "dbparser"))
get_genes_for_phenotype(hpo_db, "HP:0001250")
get_genes_for_phenotype(hpo_db, "thrombosis")
```

merge_drugbank_hpo *Merge DrugBank and HPO Gene-Phenotype Database Objects*

Description

Creates an integrated dvobject by linking DrugBank target gene data with HPO gene-phenotype associations using gene symbols as the bridge.

Usage

```
merge_drugbank_hpo(db_object, hpo_db)
```

Arguments

db_object	A dvobject from parseDrugBank() OR an existing merged dvobject (containing $drugbank).
hpo_db	A dvobject produced by dbparser::parseHPO().

Details

This function performs the following key steps: 1. Extracts gene names from DrugBank polypeptide data ($cett$targets$polypeptides$general_information). 2. Links gene names to DrugBank IDs via the target_id bridge ($cett$targets$general_information). 3. Optionally resolves DrugBank IDs to drug names. 4. Enriches the HPO gene-phenotype association table by adding drugbank_id and drug_name columns. 5. Assembles a merged object containing all original data plus the enriched table and the gene mapping bridge.

Supports piping and chaining with other merge functions.

Value

A new dvobject containing the integrated data.

See Also

Other mergers: merge_drugbank_onsides(), merge_drugbank_twosides()

Examples

```
## Not run:
drugbank <- parseDrugBank("path/to/drugbank.xml")
hpo      <- parseHPO("path/to/gene_attribute_edges.tsv")

# Standalone merge
merged_db <- merge_drugbank_hpo(drugbank, hpo)

# Chained with other merges (any order)
full_db <- drugbank %>%
  merge_drugbank_hpo(hpo) %>%
```

```
  merge_drugbank_onsides(onsides) %>%
  merge_drugbank_twosides(twosides)

# --- Example: Find drugs targeting genes associated with seizures ---
seizure_drugs <- full_db$integrated_data$gene_phenotype_enriched %>%
  filter(grepl("seizure", phenotype_name, ignore.case = TRUE)) %>%
  filter(!is.na(drugbank_id)) %>%
  select(drug_name, drugbank_id, gene_symbol, phenotype_name) %>%
  distinct()

## End(Not run)
```

`merge_drugbank_onsides` *Merge DrugBank and OnSIDES Database Objects*

Description

Creates an integrated dvobject object by linking DrugBank dvobject with OnSIDES dvobject using RxNorm CUIs as the bridge.

Usage

```
merge_drugbank_onsides(db_object, onsides_db)
```

Arguments

`db_object`	A dvobject from ‘parseDrugBank()‘ OR an existing merged dvobject (containing ‘$drugbank‘).
`onsides_db`	A dvobject produced by ‘dbparser::parseOnSIDES()‘.

Details

This function performs the following key steps: 1. Creates a mapping table between DrugBank IDs and RxNorm CUIs from the DrugBank object. 2. Enriches the relevant OnSIDES tables (‘vocab_rxnorm_ingredient‘ and optionally ‘high_confidence‘) by adding a ‘drugbank_id‘ column. 3. Assembles a new list object containing all original tables plus the enriched ones and the ID mapping table itself.

The resulting object allows for powerful queries that span both mechanistic data from DrugBank and clinical side-effect data from OnSIDES. Supports piping and chaining with other merge functions.

Value

A new dvobject containing the integrated data.

See Also

Other mergers: `merge_drugbank_hpo()`, `merge_drugbank_twosides()`

Examples

```
## Not run:
# First, parse the individual databases
drugbank <- parseDrugBank("path/to/drugbank.xml")
onsides <- parseOnSIDES("path/to/onsides_csvs/")

# Now, merge them into a single, powerful object
merged_db <- merge_drugbank_onsides(drugbank, onsides)

# --- Example Analysis: Find the protein targets of all drugs known to ---
# --- cause the side effect "Hepatitis" with high confidence.        ---

# 1. Find the MedDRA ID for "Hepatitis"
hepatitis_id <- merged_db$onsides$vocab_meddra_adverse_effect %>%
  filter(meddra_name == "Hepatitis") %>%
  pull(meddra_id)

# 2. Find all drug ingredients linked to this effect in the high_confidence table
drug_ids_causing_hepatitis <- merged_db$onsides$high_confidence_enriched %>%
  filter(effect_meddra_id == hepatitis_id) %>%
  pull(drugbank_id) %>%
  na.omit() %>%
  unique()

# 3. Look up the targets for these DrugBank IDs
targets_of_interest <- merged_db$targets %>%
  filter(parent_key %in% drug_ids_causing_hepatitis) %>%
  select(drug_id = parent_key, target_name = name, gene_name)

head(targets_of_interest)

## End(Not run)
```

`merge_drugbank_twosides` *Merge a DrugBank dvobject with a TWOSIDES dvobject*

Description

Integrates drug-drug interaction data from TWOSIDES with the rich mechanistic information from DrugBank. This function is chainable and can accept a raw DrugBank object or an already-merged dvobject.

Usage

```
merge_drugbank_twosides(db_object, twosides_db)
```

Arguments

db_object	A dvobject from ‘parseDrugBank()‘ or an existing merged dvobject.
twosides_db	A dvobject from ‘parseTWOSIDES()‘.

Value

A new, nested dvobject with the TWOSIDES data added.

See Also

Other mergers: merge_drugbank_hpo(), merge_drugbank_onsides()

---

parseDrugBank *parseDrugBank*

Description

parses given DrugBank XML database into a dvobject. dvobject is a list of data.frames in which each data.frame represents a part of parsed data (i.e drugs, prices, carriers, ...)

Usage

```
parseDrugBank(
  db_path,
  drug_options = NULL,
  parse_salts = FALSE,
  parse_products = FALSE,
  references_options = NULL,
  cett_options = NULL
)
```

Arguments

db_path	**string**, full path for the **DrugBank** xml or zip file.
drug_options	**character vector**, list of sub drug related nodes names options to parse (default = NULL). Check drug_node_options() for all available options. If its value is ‘NULL‘ ONLY ‘drug_general_information‘ will be placed in the returned dvobject.
parse_salts	**boolean**, parse salts info (default = FALSE)
parse_products	**boolean**, parse products info (default = FALSE)
references_options	**character vector**, list of sub references related nodes names options to parse (default = NULL). Check references_node_options() for all available options.
cett_options	**character vector**, list of sub cett related nodes names options to parse (default = NULL). Check cett_nodes_options() for all available options.

Value

dvobject

See Also

Other parsers: cett_nodes_options(), drug_node_options(), parseHPO(), parseOnSIDES(), parseTWOSIDES(), references_node_options()

---

parseHPO *Parse HPO Gene-Phenotype Associations*

Description

Reads the Human Phenotype Ontology (HPO) gene-phenotype associations as distributed by Harmonizome (gene_attribute_edges.txt) and returns a dvobject with gene, phenotype and association tables.

Usage

```
parseHPO(
  filepath,
  minimum_weight = 0,
  db_version = NULL,
  db_exported_date = NULL
)
```

Arguments

filepath	Character. Path to the Harmonizome HPO gene_attribute_edges.txt file, plain or gzip-compressed.
minimum_weight	Numeric. Minimum association weight to keep. The default 0 keeps all associations.
db_version	Character or NULL. Dataset version, stored in the object metadata.
db_exported_date	Character or NULL. Dataset release date, stored in the object metadata.

Details

The Harmonizome file is tab separated and has two header lines: the column names (source, source_desc, source_id, target, target_desc, target_id, weight) followed by a descriptor row (GeneSym, NA, GeneID, Phenotype, HPOID, NA, weight). The parser drops the descriptor row and renames the columns to domain-specific names. Columns that carry no information in this dataset (source_desc and the -666 placeholder in target_id) are not kept.

Gzip-compressed files (.gz) are read directly.

The resulting object can be linked to DrugBank target genes with merge_drugbank_hpo.

Value

A dvobject with three tibbles:

associations One row per gene-phenotype pair: gene_symbol, entrez_id, hpo_id, phenotype_name and association_weight.

genes Unique genes: gene_symbol, entrez_id and n_phenotypes, the number of associated phenotypes.

phenotypes Unique phenotypes: hpo_id, phenotype_name and n_genes, the number of associated genes.

See Also

Other parsers: cett_nodes_options(), drug_node_options(), parseDrugBank(), parseOnSIDES(), parseTWOSIDES(), references_node_options()

Examples

```
hpo_path <- system.file("extdata", "hpo_gene_attribute_edges.txt",
                        package = "dbparser")
hpo_db <- parseHPO(hpo_path)
hpo_db$phenotypes
show_dvobject_metadata(hpo_db)
```

---

parseOnSIDES *Parse the OnSIDES Core Relational Database*

Description

Parses the core relational tables from the OnSIDES database.

Usage

```
parseOnSIDES(
  dataDir,
  include_high_confidence = TRUE,
  db_version = NULL,
  db_exported_date = NULL
)
```

Arguments

dataDir	A string specifying the path to the directory containing the OnSIDES CSV files.
include_high_confidence	Logical. If TRUE (the default), the function will also parse the 'high_confidence.csv' file, which is a pre-aggregated summary of ingredient-to-effect relationships. If the file is not found, a warning is issued.

db_version used onside version (default = NULL)

db_exported_date
used onside release date (default = NULL)

Value

dvobject

See Also

Other parsers: cett_nodes_options(), drug_node_options(), parseDrugBank(), parseHPO(), parseTWOSIDES(), references_node_options()

---

parseTWOSIDES *Parse the TWOSIDES Drug-Drug Interaction Database*

Description

Reads the TWOSIDES data file, which contains adverse event data for pairs of drugs taken concurrently (N=2 interactions).

Usage

```
parseTWOSIDES(twosides_file_path, db_version = NULL, db_exported_date = NULL)
```

Arguments

twosides_file_path
Path to the TWOSIDES data file (e.g., 'TWOSIDES.csv.gz').

db_version used twoside version (default = NULL)

db_exported_date
used twoside release date (default = NULL)

Details

TWOSIDES is a database of drug-drug interaction safety signals mined from the FDA's Adverse Event Reporting System using the same approach as is used to generate OffSIDES.

Database fields as follow:

drug_1_rxnorn_id RxNORM identifier for drug 1

drug_1_concept_name RxNORM name string for drug 1

drug_2_rxnorm_id RxNORM identifier for drug 2

drug_2_concept_name RxNORM name string for drug 3

condition_meddra_id MedDRA identifier for the side effect

condition_concpet_name MedDRA name string for the side effect

A The number of reports for the pair of drugs that report the side effect

B The number of reports for the pair of drugs that do not report the side effect

C The number of reports for other PSM matched drugs (including perhaps the single versions of drug 1 or drug 2) that report the side effect

D The number of reports for other PSM matched drugs and other side effects

PRR Proportional reporting ratio, PRR=(A/(A+B))/(C/(C+D))

PRR_error Error estimate of the PRR

mean_reporting_frequency Proportion of reports for the drug that report the side effect, A/(A+B)

Value

A dvobject of class 'TWOSIDESDB' containing the 'drug_drug_interactions' data frame and associated metadata.

See Also

Other parsers: `cett_nodes_options()`, `drug_node_options()`, `parseDrugBank()`, `parseHPO()`, `parseOnSIDES()`, `references_node_options()`

`references_node_options`

returns references node valid options.

Description

returns references node valid options.

Usage

```
references_node_options()
```

Value

list of references valid options

See Also

Other parsers: `cett_nodes_options()`, `drug_node_options()`, `parseDrugBank()`, `parseHPO()`, `parseOnSIDES()`, `parseTWOSIDES()`

search_phenotypes *Search HPO Phenotypes by Keyword*

Description

Finds phenotypes whose name contains a keyword and lists their associated genes. Matching is case-insensitive and treats `keyword` as a plain string, not a regular expression.

Usage

```
search_phenotypes(hpo_db, keyword, max_results = 50L)
```

Arguments

`hpo_db`	An HPO `dvobject` returned by `parseHPO`.
`keyword`	Character. Search term, e.g. `"seizure"`.
`max_results`	Integer. Maximum number of phenotypes to return.

Value

A tibble with columns `hpo_id`, `phenotype_name`, `n_genes` and `genes` (comma-separated gene symbols), ordered by decreasing `n_genes`.

See Also

Other utility: `add_database_info()`, `get_genes_for_phenotype()`, `show_dvobject_metadata()`, `subset_drugbank_dvobject()`, `subset_onsides_dvobject()`

Examples

```
hpo_db <- parseHPO(system.file("extdata", "hpo_gene_attribute_edges.txt",
                               package = "dbparser"))
search_phenotypes(hpo_db, "seizure")
```

show_dvobject_metadata *Display dvobject Metadata*

Description

Displays information about passed dbobject object including basic info, database metadata, and all data.frames contained within nested lists.

Usage

```
show_dvobject_metadata(obj, return_df = FALSE)
```

Arguments

obj	A dvobject
return_df	Logical. If TRUE, returns metadata data.frame without printing. Default is FALSE.

Value

Invisibly returns a data.frame containing dvobject metadata

See Also

Other utility: add_database_info(), get_genes_for_phenotype(), search_phenotypes(), subset_drugbank_dvobjec
subset_onsides_dvobject()

Examples

```
## Not run:
display_merged_db_attrs(drugbank)
metadata <- display_merged_db_attrs(drugbank, return_df = TRUE)

## End(Not run)
```

---

subset_drugbank_dvobject

Subset a DrugBank dvobject by a vector of DrugBank IDs

Description

Subset a DrugBank dvobject by a vector of DrugBank IDs

Usage

```
subset_drugbank_dvobject(dvobject, drug_ids)
```

Arguments

dvobject	The dvobject from ‘parseDrugBank()‘.
drug_ids	A character vector of ‘drugbank_id‘ values to keep.

Details

Intelligently filters a DrugBank dvobject to retain only the data associated with a specified list of drugbank_ids. It correctly handles the deep, multi-level nested structure of the entire object, including the complex relationships within the ‘cett‘ list.

Value

A new, smaller dvobject with the same structure and attributes.

See Also

Other utility: add_database_info(), get_genes_for_phenotype(), search_phenotypes(), show_dvobject_metadata(
subset_onsides_dvobject()

Examples

```
## Not run:
library(dbparser)
one_drug <- subset_drugbank_dvobject(dvobject = dbdataset::drugbank,
                                     drug_ids = "DB00001")

## End(Not run)
```

---

subset_onsides_dvobject

Subset an OnSIDES dvobject by a vector of RxNorm Ingredient IDs (Schema-Aware)

Description

Intelligently filters an OnSIDES dvobject by cascading filters through the relational tables, ensuring the final subset is self-consistent.

Usage

```
subset_onsides_dvobject(dvobject, ingredient_ids)
```

Arguments

dvobject	A dvobject from 'parseOnSIDES()'.
ingredient_ids	A character vector of RxNorm CUIs (ingredients) to keep.

Value

A new, smaller dvobject with the same structure.

See Also

Other utility: add_database_info(), get_genes_for_phenotype(), search_phenotypes(), show_dvobject_metadata(
subset_drugbank_dvobject()

Index

∗ **mergers**
  merge_drugbank_hpo, 5
  merge_drugbank_onsides, 6
  merge_drugbank_twosides, 7
∗ **parsers**
  cett_nodes_options, 3
  drug_node_options, 3
  parseDrugBank, 8
  parseHPO, 9
  parseOnSIDES, 10
  parseTWOSIDES, 11
  references_node_options, 12
∗ **utility**
  add_database_info, 2
  get_genes_for_phenotype, 4
  search_phenotypes, 13
  show_dvobject_metadata, 13
  subset_drugbank_dvobject, 14
  subset_onsides_dvobject, 15

add_database_info, 2
add_database_info(), *4*, *13–15*

cett_nodes_options, 3
cett_nodes_options(), *3*, *9–12*

drug_node_options, 3
drug_node_options(), *3*, *9–12*

get_genes_for_phenotype, 4
get_genes_for_phenotype(), *2*, *13–15*

merge_drugbank_hpo, 5, *9*
merge_drugbank_hpo(), *6*, *8*
merge_drugbank_onsides, 6
merge_drugbank_onsides(), *5*, *8*
merge_drugbank_twosides, 7
merge_drugbank_twosides(), *5*, *6*

parseDrugBank, 8
parseDrugBank(), *3*, *10–12*
parseHPO, *4*, 9, *13*
parseHPO(), *3*, *9*, *11*, *12*
parseOnSIDES, 10
parseOnSIDES(), *3*, *9*, *10*, *12*
parseTWOSIDES, 11
parseTWOSIDES(), *3*, *9–12*

references_node_options, 12
references_node_options(), *3*, *9–12*

search_phenotypes, 13
search_phenotypes(), *2*, *4*, *14*, *15*
show_dvobject_metadata, 13
show_dvobject_metadata(), *2*, *4*, *13*, *15*
subset_drugbank_dvobject, 14
subset_drugbank_dvobject(), *2*, *4*, *13–15*
subset_onsides_dvobject, 15
subset_onsides_dvobject(), *2*, *4*, *13–15*