Package ‘nciTools’

August 15, 2025

Title Download and Process National Cancer Institute Thesaurus

Version 0.0.0.9000

Description Functions to download, prune and save NCI thesaurus. Plus additional functions for creating quanteda dictionaries and traversing ontology.

License `use_mit_license()`, `use_gpl3_license()` or friends to pick a
license

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Imports DBI,
dplyr,
httr,
purrr,
quanteda,
stringr,
tibble,
tidyr

R topics documented:

createDictionaries *Create dictionaries based on NCI thesaurus*

Description

Uses the `createDictionary` function to create `quanteda` dictionaries of cancer types, drugs, genes and molecular alterations.
Dictionary keys are NCI thesaurus codes, dictionary values are NCI thesaurus synonyms.

Usage

```
createDictionaries(thesaurus)
```

Arguments

`thesaurus`	(processed) NCI thesaurus.

Value

A list containing four dictionaries:

- `cancer_dict`: Entities of semantic type 'Neoplastic Process' .
- `drug_dict`: Entities of semantic type 'Pharmacologic Substance', 'Biologically Active Substance', 'Clinical Drug', 'Steroid', 'Immunologic Factor', and 'Therapeutic or Preventive Procedure'.
- `gene_dict`: Entities of semantic type 'Gene or Genome'.
- `alteration_dict`: Entities of semantic type 'Cell or Molecular Dysfunction'.

createDictionary *Create a Specific Dictionary from NCI Thesaurus Data*

Description

This function constructs a dictionary for a specific set of semantic types from the NCI Thesaurus data. It processes the source data to filter and organize terms based on the specified semantic types, facilitating the creation of targeted dictionaries for use in annotation tasks. The output dictionary is structured to associate unique codes with their corresponding synonyms, enhancing the annotation process by allowing for the identification of terms through various synonymous expressions.

Usage

```
createDictionary(source_data, semantic_types)
```

Arguments

`source_data`	A dataset derived from the NCI Thesaurus, expected to contain detailed term information including unique codes, synonyms, and semantic types. This data serves as the input for generating the dictionary.
`semantic_types`	A character vector specifying the semantic types of interest. These types define the scope of the dictionary, determining which terms from the source data are included based on their semantic classification.

Value

Returns a dictionary object structured for annotation purposes. The dictionary is specifically designed to map unique codes (representing specific terms) to a list of synonyms, allowing for comprehensive term identification. This is particularly useful for annotating texts with terms that may appear in various synonymous forms.

`downloadThesaurus` *Download and Extract the Latest NCI Thesaurus*

Description

The latest version of the NCI thesaurus is downloaded and unzipped to the data/raw subfolder.
The NCI Thesaurus is a reference terminology covering a broad range of topics relevant to cancer and biomedical research.

Usage

```
downloadThesaurus()
```

Details

Thesaurus downloaded from `https://evs.nci.nih.gov/ftp1/NCI_Thesaurus/Thesaurus.FLAT.zip` - this should always be the latest version.

Value

Path to the extracted NCI Thesaurus flat file.

`expandAlterations` *Expand Molecular Dysfunctions*

Description

Expand a set of NCI thesaurus codes related to cell/molecular dysfunctions to include all protein and DNA terms, and all of their ancestors.
For example, `C98365` (KRAS G12C, a gene product variation) is expanded to include the respective gene variation(s) (e.g. `C98366`), plus all their ancestors (e.g. `C135715` KRAS exon 2 mutation, and `C98362` KRAS Protein Variant etc)

Usage

```
expandAlterations(codes)
```

Arguments

`codes`	One or more codes for NCI thesaurus entities of semantic type `Cell or Molecular Dysfunction`.

Details

Relationships described by `Gene_Product_Sequence_Variation_Encoded_By_Gene_Mutant` and `Gene_Mutant_Encodes_Gene_Product_Sequence_Variation` are added to the specified codes, and then all codes are expanded to include all ancestor terms

Value

The expanded set of codes

See Also

`getNCIt_relationships` get relationships from NCI thesaurus API.

`getAncestors` get higher level terms

`getAncestors` *Retrieve Ancestors of NCI Thesaurus Terms*

Description

Given a set of codes, finds all ancestor concepts (e.g., parents, grandparents) from the NCI Thesaurus hierarchy.

Usage

```
getAncestors(codes, db_connection)
```

Arguments

`codes`	A character vector of NCI Thesaurus concept codes whose ancestors should be retrieved.
`db_connection`	connection to a database that includes the `thesaurus_flat` and `parents` tables

Details

This function builds and executes a recursive SQL query to retrieve all ancestor nodes for a given list of concept codes.
Ancestors are identified using a parent-child relationship table (`NCIt_parents`).
The results are then merged with a pruned thesaurus (`NCIt_pruned`) to attach preferred terms (`PT`) for easier interpretation.

Internally, the function:

- Quotes and formats the codes for SQL `IN` syntax
- Uses a recursive CTE (Common Table Expression) to traverse upward through the concept hierarchy
- Merges results with the pruned thesaurus for human-readable output

Value

A data frame of ancestor concepts, including their codes and preferred terms.

See Also

`sqldf::sqldf()`, `dplyr::select()`, `base::merge()`

`getCodes` *Get NCI thesaurus codes from preferred terms*

Description

Given one or more preferred terms (i.e. first synonyms in NCI thesaurus), look up relevant NCI thesaurus codes.

Usage

```
getCodes(PTs, db_connection)
```

Arguments

`PTs`	the preferred term(s)
`db_connection`	connection to a database that includes the thesaurus_flat table

Value

Matching NCI thesaurus code(s)

`getDescendants` *Retrieve Descendants of NCI Thesaurus Terms*

Description

Given a set of codes, finds all descendant concepts (e.g., children, grandchildren) from the NCI Thesaurus hierarchy.

Usage

```
getDescendants(codes, db_connection)
```

Arguments

`codes`	A character vector of NCI Thesaurus concept codes whose descendants should be retrieved.
`db_connection`	connection to a database that includes the `thesaurus_flat` and `parents` tables

Details

This function builds and executes a recursive SQL query to retrieve all descendant nodes for a given list of concept codes.
Descendants are identified using a parent-child relationship table (`NCIt_parents`).
The results are returned as a unique character vector of descendant codes.

Internally, the function:

- Quotes and formats the input codes for SQL `IN` syntax
- Uses a recursive CTE (Common Table Expression) to traverse downward through the concept hierarchy
- Returns a de-duplicated vector of descendant concept codes

Value

A character vector of descendant concept codes.

See Also

`sqldf::sqldf()`, `dplyr::pull()`, `base::unique()`

`getNCIt_relationships` *Get NCIt Relationships from NCI Thesaurus API*

Description

This function queries the NCI Thesaurus to retrieve relationships based on a given code and inclusion criteria. It fetches information from the National Cancer Institute's EVS API for a given concept and returns the specified relationships.

Usage

```
getNCIt_relationships(code, include_what)
```

Arguments

`code`	A character string representing the NCIt code (concept ID) for which to retrieve relationships.
`include_what`	A character string indicating the type of relationships to include. Valid options are: `roles`, `inverseRoles`, `associations`, or `inverseAssociations`. This determines which relationship information will be returned for the specified code.

Details

This function constructs a request to the NCI EVS API endpoint to search for the specified NCIt code and includes the relationships based on the `include_what` parameter. The result is parsed into a tibble (data frame), with each relationship type presented in separate columns.

The following values are valid for `include_what`:

- `roles`: Returns roles associated with the concept.
- `inverseRoles`: Returns inverse roles related to the concept.
- `associations`: Returns associations linked to the concept.
- `inverseAssociations`: Returns inverse associations for the concept.

Value

A data frame containing the specified relationships for the NCIt code. If no relationships are found, it returns `NULL`.

See Also

https://evsexplore.semantics.cancer.gov/evsexplore/evsapi

`processThesaurus` *Process the downloaded NCI Thesaurus*

Description

This function processes the NCI Thesaurus from a flat file format, applying a series of transformations to: filter obsolete concepts, extract key information, prepare the data for analysis.

The transformations include normalising synonyms, identifying solid neoplasm children, and handling generic entities and synonyms.

Usage

```
processThesaurus(thesaurus_file)
```

Arguments

`thesaurus_file`	The path to the downloaded (raw) version of the NCI Thesaurus.

Details

The NCI Thesaurus is processed through several steps to make it suitable for analysis:

- Column names are set.
- A new column 'PT' (preferred term) is added.
- The first synonym is used as the PT for each entity.
- Obsolete and retired concepts are removed.
- New parent-child relationships are added, e.g. between C9292 Solid Neoplasm and various solid tumour types.
- Generic entities and synonyms that could lead to ambiguous matches are removed.
- New synonyms are added where appropriate.
- Parent-child relationships and synonyms are extracted into separate tables for convenient reuse in the pipeline.

Value

Returns a list containing three dataframes:

- `thesaurus`: The processed NCI Thesaurus data with relevant fields extracted and formatted for analysis, including codes, preferred terms (PT), parents, synonyms, and semantic types.
- `parents`: A table mapping codes to their parent codes to facilitate hierarchical analysis of the thesaurus entries.
- `synonyms`: A table of synonyms for each code, expanded from the compressed format in the source file to aid in text matching and lookup tasks.

`saveThesaurus` *Save annotated trial data to SQLite database*

Description

This is the final step in the trial annotation pipeline.
Data are saved to a local SQLite database for ingestion by user interfaces.

Usage

```
saveThesaurus(thesaurus_processed, db_connection)
```

Arguments

`thesaurus_processed`
A list of dataframes containing processed information from NCI thesaurus.

`db_connection` connection to a SQLite database

Details

list-columns are flattened into pipe-delimited strings

Value

Returns the relative path to the populated SQLite database

Index