Package 'tidypmc'

August 27, 2024

Type Package

Title Parse Full Text XML Documents from PubMed Central

Version 2.0

Description Parse XML documents from the Open Access subset of Europe PubMed Central <https://europepmc.org>
including section paragraphs, tables, captions and references.

URL https://github.com/ropensci/tidypmc

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

License GPL-3

Encoding UTF-8

VignetteBuilder knitr

Imports xml2, tokenizers, stringr, tibble, dplyr, readr

Suggests europepmc, tidytext, rmarkdown, knitr, testthat, covr

RoxygenNote 7.3.2

NeedsCompilation no

Author Chris Stubben [aut, cre]

Maintainer Chris Stubben <chris.stubben@hci.utah.edu>

Repository CRAN

Date/Publication 2024-08-27 04:10:03 UTC

Contents

`collapse_rows` *Collapse a list of PubMed Central tables*

Description

Collapse rows into a semi-colon delimited list with column names and cell values

Usage

```
collapse_rows(pmc, na.string)
```

Arguments

`pmc`	a list of tables, usually from `pmc_table`
`na.string`	additional cell values to skip, default is NA and ""

Value

A tibble with table and row number and collapsed text

Author(s)

Chris Stubben

Examples

```
x <- data.frame(
  genes = c("aroB", "glnP", "ndhA", "pyrF"),
  fold_change = c(2.5, 1.7, -3.1, -2.6)
)
collapse_rows(list(`Table 1` = x))
```

extract_acronyms *Find acronyms in parentheses*

Description

This function searches for words preceding the acronym that start with the same initial letter and will likely fail in many situations.

Usage

```
extract_acronyms(txt)
```

Arguments

txt	A tibble from pmc_text or character vector

Value

A tibble with acronyms

Author(s)

Chris Stubben

Examples

```
txt <- c(
  "An acronym like multinucleated giant cell (MGC)",
  "is later mentioned as MGC in the paper.")
extract_acronyms(txt)
```

path_string *Print a hierarchical path string*

Description

Print a hierarchical path string from a vector of names and levels

Usage

```
path_string(x, n)
```

Arguments

x	a vector of names
n	a vector of numbers with indentation level

Value

a character vector

Note

Used by `pmc_text` to print full path to subsection title

Author(s)

Chris Stubben

Examples

```
x <- c("carnivores", "bears", "polar", "grizzly", "cats", "tiger", "rodents")
n <- c(1, 2, 3, 3, 2, 3, 1)
path_string(x, n)
```

---

pmc_caption *Split captions into sentences*

Description

Split figure, table and supplementary material captions into sentences

Usage

```
pmc_caption(doc)
```

Arguments

doc	`xml_document` from PubMed Central

Value

a tibble with tag, label, sentence number and text

Author(s)

Chris Stubben

Examples

```
# doc <- pmc_xml("PMC2231364") # OR
doc <- xml2::read_xml(system.file("extdata/PMC2231364.xml",
  package = "tidypmc"
))
x <- pmc_caption(doc)
x
dplyr::filter(x, sentence == 1)
```

pmc_metadata *Get article metadata*

Description

Get a list of journal and article metadata in /front tag

Usage

```
pmc_metadata(doc)
```

Arguments

doc	xml_document from PubMed Central

Value

a list

Author(s)

Chris Stubben

Examples

```
# doc <- pmc_xml("PMC2231364") # OR
doc <- xml2::read_xml(system.file("extdata/PMC2231364.xml",
  package = "tidypmc"
))
pmc_metadata(doc)
```

pmc_reference *Format references cited*

Description

Format references cited

Usage

```
pmc_reference(doc)
```

Arguments

doc	xml_document from PubMed Central

Value

a tibble with id, pmid, authors, year, title, journal, volume, pages, and doi.

Note

Mixed citations without any child tags are added to the author column.

Author(s)

Chris Stubben

Examples

```
# doc <- pmc_xml("PMC2231364")
doc <- xml2::read_xml(system.file("extdata/PMC2231364.xml",
  package = "tidypmc"
))
x <- pmc_reference(doc)
x
```

`pmc_table` *Convert table nodes to tibbles*

Description

Convert PubMed Central table nodes into a list of tibbles

Usage

```
pmc_table(doc)
```

Arguments

`doc`	`xml_document` from PubMed Central

Value

a list of tibbles

Note

Saves the caption and footnotes as attributes and collapses multiline headers, expands all rowspan and colspan attributes and adds subheadings to column one.

Author(s)

Chris Stubben

Examples

```
# doc <- pmc_xml("PMC2231364")
doc <- xml2::read_xml(system.file("extdata/PMC2231364.xml",
  package = "tidypmc"
))
x <- pmc_table(doc)
sapply(x, dim)
x
attributes(x[[1]])
```

`pmc_text` *Split section paragraphs into sentences*

Description

Split section paragraph tags into a table with subsection titles and sentences using `tokenize_sentences`

Usage

```
pmc_text(doc, sentence = TRUE)
```

Arguments

`doc`	`xml_document` from PubMed Central
`sentence`	split paragraphs into sentences, default TRUE

Value

a tibble with section, paragraph and sentence number and text

Note

Subsections may be nested to arbitrary depths and this function will return the entire path to the subsection title as a delimited string like "Results; Predicted functions; Pathogenicity". Tables, figures and formulas that are nested in section paragraphs are removed, superscripted references are replaced with brackets, and any other superscripts or subscripts are separared with ^ and _.

Author(s)

Chris Stubben

Examples

```
# doc <- pmc_xml("PMC2231364")
doc <- xml2::read_xml(system.file("extdata/PMC2231364.xml",
  package = "tidypmc"
))
txt <- pmc_text(doc)
txt
dplyr::count(txt, section, sort = TRUE)
```

pmc_xml *Download XML from PubMed Central*

Description

Download XML from PubMed Central

Usage

```
pmc_xml(id)
```

Arguments

id	a PMC id starting with 'PMC'

Value

xml_document

Source

https://europepmc.org/RestfulWebService

Examples

```
## Not run:
doc <- pmc_xml("PMC2231364")

## End(Not run)
```

repeat_sub *Repeat table subheadings*

Description

Repeat table subheadings in a new column

Usage

```
repeat_sub(x, column = "subheading", first = TRUE)
```

Arguments

x	a tibble with subheadings
column	new column name, default subheading
first	add subheader as first column, default TRUE

Details

Identifies subheadings in a data frame by checking for rows with a non-empty first column and all other columns are empty. Removes subheader rows and repeats values down a new column.

Value

a tibble

Author(s)

Chris Stubben

Examples

```
x <- data.frame(
  genes = c("Up", "aroB", "glnP", "Down", "ndhA", "pyrF"),
  fold_change = c(NA, 2.5, 1.7, NA, -3.1, -2.6)
)
x
repeat_sub(x)
repeat_sub(x, "regulated", first = FALSE)
```

---

separate_refs *Separate references cited into multiple rows*

Description

Separates references cited in brackets or parentheses into multiple rows and splits the comma-delimited numeric strings and expands ranges like 7-9 into new rows

Usage

```
separate_refs(txt, column = "text")
```

Arguments

txt	a table
column	column name, default "text"

Value

a tibble

Author(s)

Chris Stubben

Examples

```
x <- data.frame(row = 1, text = "some important studies [7-9,15]")
separate_refs(x)
```

`separate_tags` *Separate locus tag into multiple rows*

Description

Separates locus tags mentioned in full text and expands ranges like YPO1970-74 into new rows

Usage

```
separate_tags(txt, pattern, column = "text")
```

Arguments

`txt`	a table
`pattern`	regular expression to match locus tags like YPO[0-9-]+ or the locus tag prefix like YPO.
`column`	column name to search, default "text"

Value

a tibble with locus tag, matching text and rows.

Author(s)

Chris Stubben

Examples

```
x <- data.frame(row = 1, text = "some genes like YPO1002 and YPO1970-74")
separate_tags(x, "YPO")
```

separate_text *Separate all matching text into multiple rows*

Description

Separate all matching text into multiple rows

Usage

```
separate_text(txt, pattern, column = "text")
```

Arguments

txt	a tibble, usually results from pmc_text
pattern	either a regular expression or a vector of words to find in text
column	column name, default "text"

Value

a tibble

Note

passed to grepl and str_extract_all

Author(s)

Chris Stubben

Examples

```
# doc <- pmc_xml("PMC2231364")
doc <- xml2::read_xml(system.file("extdata/PMC2231364.xml",
        package = "tidypmc"))
txt <- pmc_text(doc)
separate_text(txt, "[ATCGN]{5,}")
separate_text(txt, "\\([A-Z]{3,6}s?\\)")
# pattern can be a vector of words
separate_text(txt, c("hmu", "ybt", "yfe", "yfu"))
# wrappers for separate_text with extra step to expand matched ranges
separate_refs(txt)
separate_tags(txt, "YPO")
```

Index