

# Package: tidypmc (via r-universe)

December 21, 2024

**Type** Package

**Title** Parse Full Text XML Documents from PubMed Central

**Version** 1.8

**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://docs.ropensci.org/tidypmc>,  
<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** 6.1.1

**Config/pak/sysreqs** libicu-dev libxml2-dev libx11-dev

**Repository** <https://ropensci.r-universe.dev>

**RemoteUrl** <https://github.com/ropensci/tidypmc>

**RemoteRef** master

**RemoteSha** d9739ce3ef6f57832f56bb488de80c48b58e1f91

## Contents

collapse_rows . . . . .	2
pmc_caption . . . . .	3
pmc_metadata . . . . .	3
pmc_reference . . . . .	4
pmc_table . . . . .	5
pmc_text . . . . .	6

pmc_xml	7
separate_genes	7
separate_refs	8
separate_tags	9
separate_text	10
<b>Index</b>	<b>11</b>

---

collapse_rows	<i>Collapse a list of PubMed Central tables</i>
---------------	---

---

## 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 <a href="#">pmc_table</a>
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))
```

---

pmc_caption	<i>Split captions into sentences</i>
-------------	--------------------------------------

---

**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	<i>Get article metadata</i>
--------------	-----------------------------

---

**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	<i>Convert table nodes to tibbles</i>
-----------	---------------------------------------

---

**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)
```

### **Arguments**

doc                   xml\_document from PubMed Central

### **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 separated 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)
```

---

separate\_genes

*Separate genes and operons into multiple rows*

---

### **Description**

Separate genes and operons mentioned in full text into multiple rows

### **Usage**

```
separate_genes(txt, pattern = "\\b[A-Za-z][a-z]{2}[A-Z0-9]+\\b",  
              genes, operon = 6, column = "text")
```

**Arguments**

txt	a table
pattern	regular expression to match genes, default is to match microbial genes like AbcD, default [A-Za-z][a-z]2[A-Z0-9]+
genes	an optional vector of genes, set pattern to NA to only match this list.
operon	operon length, default 6. Split genes with 6 or more letters into separate genes, for example AbcDEF is split into abcD, abcE and abcF.
column	column name to search, default "text"

**Value**

a tibble with gene name, matching text and rows.

**Note**

Check for genes in italics using `xml_text(xml_find_all(doc, "//sec//p//i"))` and update the pattern or add additional genes as an optional vector if needed

**Author(s)**

Chris Stubben

**Examples**

```
x <- data.frame(row = 1, text = "Genes like Yack, hmu and sufABC")
separate_genes(x)
separate_genes(x, genes = "hmu")
```

---

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 YP01002 and YP01970-74")
separate_tags(x, "YPO")
```

---

separate_text	<i>Separate all matching text into multiple rows</i>
---------------	--

---

**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_genes(txt)
separate_tags(txt, "YPO")
```

# Index

`collapse_rows`, [2](#)

`pmc_caption`, [3](#)

`pmc_metadata`, [3](#)

`pmc_reference`, [4](#)

`pmc_table`, [2](#), [5](#)

`pmc_text`, [6](#)

`pmc_xml`, [7](#)

`separate_genes`, [7](#)

`separate_refs`, [8](#)

`separate_tags`, [9](#)

`separate_text`, [10](#)