

# Package ‘rentrez’

July 23, 2025

**Version** 1.2.4

**Date** 2025-06-11

**Title** 'Entrez' in R

**Depends** R (>= 2.6.0)

**Imports** XML, httr (>= 0.5), jsonlite (>= 0.9)

**Suggests** testthat, knitr, rmarkdown

**URL** <https://github.com/ropensci/rentrez/>

**BugReports** <https://github.com/ropensci/rentrez/issues/>

**Description** Provides an R interface to the NCBI's 'EUtils' API, allowing users to search databases like 'GenBank' <<https://www.ncbi.nlm.nih.gov/genbank/>> and 'PubMed' <<https://pubmed.ncbi.nlm.nih.gov/>>, process the results of those searches and pull data into their R sessions.

**VignetteBuilder** knitr

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**NeedsCompilation** no

**Author** David Winter [aut, cre] (ORCID: <<https://orcid.org/0000-0002-6165-0029>>),  
Scott Chamberlain [ctb] (ORCID: <<https://orcid.org/0000-0003-1444-9135>>),  
Han Guangchun [ctb] (ORCID: <<https://orcid.org/0000-0001-9277-2507>>)

**Maintainer** David Winter <david.winter@gmail.com>

**Repository** CRAN

**Date/Publication** 2025-06-11 11:00:02 UTC

## Contents

entrez_citmatch . . . . .	2
entrez_dbs . . . . .	3
entrez_db_links . . . . .	4
entrez_db_searchable . . . . .	5
entrez_db_summary . . . . .	6
entrez_fetch . . . . .	7
entrez_global_query . . . . .	8
entrez_info . . . . .	9
entrez_link . . . . .	10
entrez_post . . . . .	11
entrez_search . . . . .	12
entrez_summary . . . . .	14
extract_from_esummary . . . . .	16
linkout_urls . . . . .	17
parse_pubmed_xml . . . . .	17
set_entrez_key . . . . .	18
<b>Index</b>	<b>19</b>

---

entrez_citmatch	<i>Fetch pubmed ids matching specially formatted citation strings</i>
-----------------	---

---

### Description

Fetch pubmed ids matching specially formatted citation strings

### Usage

```
entrez_citmatch(bdata, db = "pubmed", retmode = "xml", config = NULL)
```

### Arguments

bdata	character, containing citation data. Each citation must be represented in a pipe-delimited format <code>journal_title year volume first_page author_name your_key </code> . The final field "your_key" is arbitrary, and can used as you see fit. Fields can be left empty, but be sure to keep 6 pipes.
db	character, the database to search. Defaults to pubmed, the only database currently available
retmode	character, file format to retrieve. Defaults to xml, as per the API documentation, though note the API only returns plain text
config	vector configuration options passed to <code>httr::GET</code>

### Value

A character vector containing PMIDs

**See Also**

[config](#) for available configs

**Examples**

```
## Not run:
ex_cites <- c("proc natl acad sci u s a|1991|88|3248|mann bj|test1|",
             "science|1987|235|182|palmenberg ac|test2|")
entrez_citmatch(ex_cites)

## End(Not run)
```

---

entrez_dbs	<i>List databases available from the NCBI</i>
------------	---

---

**Description**

Retrieves the names of databases available through the EUtils API

**Usage**

```
entrez_dbs(config = NULL)
```

**Arguments**

config            config vector passed to `httr::GET`

**Value**

character vector listing available dbs

**See Also**

Other info: [entrez\\_db\\_links\(\)](#), [entrez\\_db\\_searchable\(\)](#), [entrez\\_db\\_summary\(\)](#), [entrez\\_info\(\)](#)

**Examples**

```
## Not run:
entrez_dbs()

## End(Not run)
```

---

entrez_db_links	<i>List available links for records from a given NCBI database</i>
-----------------	--

---

### Description

For a given database, fetch a list of other databases that contain cross-referenced records. The names of these records can be used as the db argument in [entrez\\_link](#)

### Usage

```
entrez_db_links(db, config = NULL)
```

### Arguments

db	character, name of database to search
config	config vector passed to <code>httr::GET</code>

### Value

An eInfoLink object (sub-classed from list) summarizing linked-databases. Can be coerced to a data-frame with `as.data.frame`. Printing the object the name of each element (which is the correct name for `entrez_link`, and can be used to get (a little) more information about each linked database (see example below).

### See Also

[entrez\\_link](#)

Other einfo: [entrez\\_db\\_searchable\(\)](#), [entrez\\_db\\_summary\(\)](#), [entrez\\_dbs\(\)](#), [entrez\\_info\(\)](#)

### Examples

```
## Not run:
taxid <- entrez_search(db="taxonomy", term="Osmeriformes")$ids
tax_links <- entrez_db_links("taxonomy")
tax_links
entrez_link(dbfrom="taxonomy", db="pmc", id=taxid)

sra_links <- entrez_db_links("sra")
as.data.frame(sra_links)

## End(Not run)
```

---

entrez\_db\_searchable *List available search fields for a given database*

---

## Description

Fetch a list of search fields that can be used with a given database. Fields can be used as part of the term argument to [entrez\\_search](#)

## Usage

```
entrez_db_searchable(db, config = NULL)
```

## Arguments

db	character, name of database to get search field from
config	config vector passed to <code>httr::GET</code>

## Value

An `eInfoSearch` object (subclassed from `list`) summarizing linked-databases. Can be coerced to a data-frame with `as.data.frame`. Printing the object shows only the names of each available search field.

## See Also

[entrez\\_search](#)

Other `einfo`: [entrez\\_db\\_links\(\)](#), [entrez\\_db\\_summary\(\)](#), [entrez\\_dbs\(\)](#), [entrez\\_info\(\)](#)

## Examples

```
## Not run:
pmc_fields <- entrez_db_searchable("pmc")
pmc_fields[["AFFL"]]
entrez_search(db="pmc", term="Otago[AFFL]", retmax=0)
entrez_search(db="pmc", term="Auckland[AFFL]", retmax=0)

sra_fields <- entrez_db_searchable("sra")
as.data.frame(sra_fields)

## End(Not run)
```

---

entrez\_db\_summary      *Retrieve summary information about an NCBI database*

---

### **Description**

Retrieve summary information about an NCBI database

### **Usage**

```
entrez_db_summary(db, config = NULL)
```

### **Arguments**

db                    character, name of database to summaries  
config                config vector passed to `httr::GET`

### **Value**

Character vector with the following data

DbName Name of database

Description Brief description of the database

Count Number of records contained in the database

MenuName Name in web-interface to EUtils

DbBuild Unique ID for current build of database

LastUpdate Date of most recent update to database

### **See Also**

Other einfo: [entrez\\_db\\_links\(\)](#), [entrez\\_db\\_searchable\(\)](#), [entrez\\_dbs\(\)](#), [entrez\\_info\(\)](#)

### **Examples**

```
## Not run:  
entrez_db_summary("pubmed")  
  
## End(Not run)
```

entrez\_fetch

*Download data from NCBI databases***Description**

Pass unique identifiers to an NCBI database and receive data files in a variety of formats. A set of unique identifiers must be specified with either the `db` argument (which directly specifies the IDs as a numeric or character vector) or a `web_history` object as returned by [entrez\\_link](#), [entrez\\_search](#) or [entrez\\_post](#).

**Usage**

```
entrez_fetch(
  db,
  id = NULL,
  web_history = NULL,
  rettype,
  retmode = "",
  parsed = FALSE,
  config = NULL,
  ...
)
```

**Arguments**

<code>db</code>	character, name of the database to use
<code>id</code>	vector (numeric or character), unique ID(s) for records in database <code>db</code> . In the case of sequence databases these IDs can take form of an NCBI accession followed by a version number (eg AF123456.1 or AF123456.2).
<code>web_history</code>	a <code>web_history</code> object
<code>rettype</code>	character, format in which to get data (eg, fasta, xml...)
<code>retmode</code>	character, mode in which to receive data, defaults to an empty string (corresponding to the default mode for <code>rettype</code> ).
<code>parsed</code>	boolean should <code>entrez_fetch</code> attempt to parse the resulting file. Only works with xml records (including those with <code>rettypes</code> other than "xml") at present
<code>config</code>	vector, http configuration options passed to <code>httr::GET</code>
<code>...</code>	character, additional terms to add to the request, see NCBI documentation linked to in references for a complete list

**Details**

The format for returned records is set by that arguments `rettype` (for a particular format) and `retmode` for a general format (JSON, XML text etc). See [Table 1](#) in the linked reference for the set of formats available for each database. In particular, note that sequence databases (nuccore, protein and their relatives) use specific format names (eg "native", "ipg") for different flavours of xml.

For the most part, this function returns a character vector containing the fetched records. For XML records (including 'native', 'ipg', 'gbc' sequence records), setting `parsed` to `TRUE` will return an `XMLInternalDocument`,

### Value

character string containing the file created

`XMLInternalDocument` a parsed XML document if `parsed=TRUE` and `rettype` is a flavour of XML.

### References

[https://www.ncbi.nlm.nih.gov/books/NBK25499/#\\_chapter4\\_EFetch\\_](https://www.ncbi.nlm.nih.gov/books/NBK25499/#_chapter4_EFetch_)

### See Also

`config` for available 'httr' configs

### Examples

```
## Not run:
katipo <- "Latrodectus katipo[Organism]"
katipo_search <- entrez_search(db="nucore", term=katipo)
kaipto_seqs <- entrez_fetch(db="nucore", id=katipo_search$ids, rettype="fasta")
#xml
kaipto_seqs <- entrez_fetch(db="nucore", id=katipo_search$ids, rettype="native")

## End(Not run)
```

---

`entrez_global_query`     *Find the number of records that match a given term across all NCBI Entrez databases*

---

### Description

Find the number of records that match a given term across all NCBI Entrez databases

### Usage

```
entrez_global_query(term, config = NULL, ...)
```

### Arguments

<code>term</code>	the search term to use
<code>config</code>	vector configuration options passed to <code>httr::GET</code>
<code>...</code>	additional arguments to add to the query

### Value

a named vector with counts for each a database



**See Also**

[config](#) for available configs

**Examples**

```
## Not run:
NCBI_data_on_best_butterflies_ever <- entrez_global_query(term="Heliconius")

## End(Not run)
```

---

entrez\_info

*Get information about EUtils databases*


---

**Description**

Gather information about EUtils generally, or a given Eutils database. Note: The most common uses-cases for the einfo util are finding the list of search fields available for a given database or the other NCBI databases to which records in a given database might be linked. Both these use cases are implemented in higher-level functions that return just this information ([entrez\\_db\\_searchable](#) and [entrez\\_db\\_links](#) respectively). Consequently most users will not have a reason to use this function (though it is exported by [rentrez](#) for the sake of completeness).

**Usage**

```
entrez_info(db = NULL, config = NULL)
```

**Arguments**

db	character database about which to retrieve information (optional)
config	config vector passed on to <code>httr::GET</code>

**Value**

XMLInternalDocument with information describing either all the databases available in Eutils (if db is not set) or one particular database (set by 'db')

**See Also**

[config](#) for available httr configurations

Other einfo: [entrez\\_db\\_links\(\)](#), [entrez\\_db\\_searchable\(\)](#), [entrez\\_db\\_summary\(\)](#), [entrez\\_dbs\(\)](#)

**Examples**

```
## Not run:
all_the_data <- entrez_info()
XML::xpathSApply(all_the_data, "//DbName", xmlValue)
entrez_dbs()

## End(Not run)
```

---

entrez\_link

*Get links to datasets related to records from an NCBI database*


---

## Description

Discover records related to a set of unique identifiers from an NCBI database. The object returned by this function depends on the value set for the `cmd` argument. Printing the returned object lists the names, and provides a brief description, of the elements included in the object.

## Usage

```
entrez_link(
  dbfrom,
  web_history = NULL,
  id = NULL,
  db = NULL,
  cmd = "neighbor",
  by_id = FALSE,
  config = NULL,
  ...
)
```

## Arguments

<code>dbfrom</code>	character Name of database from which the Id(s) originate
<code>web_history</code>	a <code>web_history</code> object
<code>id</code>	vector with unique ID(s) for records in database <code>db</code> .
<code>db</code>	character Name of the database to search for links (or use "all" to search all databases available for <code>db</code> . <code>entrez_db_links</code> allows you to discover databases that might have linked information (see examples).
<code>cmd</code>	link function to use. Allowed values include <ul style="list-style-type: none"> <li><code>neighbor</code> (default). Returns a set of IDs in <code>db</code> linked to the input IDs in <code>dbfrom</code>.</li> <li><code>neighbor_score</code>. As 'neighbor', but additionally returns similarity scores.</li> <li><code>neighbor_history</code>. As 'neighbor', but returns web history objects.</li> <li><code>acheck</code>. Returns a list of linked databases available from NCBI for a set of IDs.</li> <li><code>ncheck</code>. Checks for the existence of links within a single database.</li> <li><code>lcheck</code>. Checks for external (i.e. outside NCBI) links.</li> <li><code>llinks</code>. Returns a list of external links for each ID, excluding links provided by libraries.</li> <li><code>llinkslib</code>. As 'llinks' but additionally includes links provided by libraries.</li> <li><code>prlinks</code>. As 'llinks' but returns only the primary external link for each ID.</li> </ul>

by_id	logical If FALSE (default) return a single eLink objects containing links for all of the provided ids. Alternatively, if TRUE return a list of eLink objects, one for each ID in id.
config	vector configuration options passed to httr::GET
...	character Additional terms to add to the request, see NCBI documentation linked to in references for a complete list

**Value**

An eLink object containing the data defined by the cmd argument (if by\_id=FALSE) or a list of such object (if by\_id=TRUE).

file XMLInternalDocument xml file resulting from search, parsed with [xmlTreeParse](#)

**References**

[https://www.ncbi.nlm.nih.gov/books/NBK25499/#\\_chapter4\\_ELink\\_](https://www.ncbi.nlm.nih.gov/books/NBK25499/#_chapter4_ELink_)

**See Also**

[config](#) for available configs

[entrez\\_db\\_links](#)

**Examples**

```
## Not run:
pubmed_search <- entrez_search(db = "pubmed", term = "10.1016/j.ympev.2010.07.013[doi]")
linked_dbs <- entrez_db_links("pubmed")
linked_dbs
nucleotide_data <- entrez_link(dbfrom = "pubmed", id = pubmed_search$ids, db = "nuccore")
#Sources for the full text of the paper
res <- entrez_link(dbfrom="pubmed", db="", cmd="llinks", id=pubmed_search$ids)
linkout_urls(res)

## End(Not run)
```

---

entrez\_post

*Post IDs to Eutils for later use*


---

**Description**

Post IDs to Eutils for later use

**Usage**

```
entrez_post(db, id = NULL, web_history = NULL, config = NULL, ...)
```

**Arguments**

db	character Name of the database from which the IDs were taken
id	vector with unique ID(s) for records in database db.
web_history	A web_history object. Can be used to add to additional identifiers to an existing web environment on the NCBI
config	vector of configuration options passed to httr::GET
...	character Additional terms to add to the request, see NCBI documentation linked to in references for a complete list

**References**

[https://www.ncbi.nlm.nih.gov/books/NBK25499/#\\_chapter4\\_EPost\\_](https://www.ncbi.nlm.nih.gov/books/NBK25499/#_chapter4_EPost_)

**See Also**

[config](#) for available httr configurations

**Examples**

```
## Not run:
so_many_snails <- entrez_search(db="nucore",
                               "Gastropoda[Organism] AND COI[Gene]", retmax=200)
upload <- entrez_post(db="nucore", id=so_many_snails$ids)
first <- entrez_fetch(db="nucore", rettype="fasta", web_history=upload,
                    retmax=10)
second <- entrez_fetch(db="nucore", file_format="fasta", web_history=upload,
                    retstart=10, retmax=10)

## End(Not run)
```

---

entrez\_search

*Search the NCBI databases using EUtils*


---

**Description**

Search a given NCBI database with a particular query.

**Usage**

```
entrez_search(
  db,
  term,
  config = NULL,
  retmode = "xml",
  use_history = FALSE,
  ...
)
```

**Arguments**

db	character, name of the database to search for.
term	character, the search term. The syntax used in making these searches is described in the Details of this help message, the package vignette and reference given below.
config	vector configuration options passed to httr::GET
retmode	character, one of json (default) or xml. This will make no difference in most cases.
use_history	logical. If TRUE return a web_history object for use in later calls to the NCBI
...	character, additional terms to add to the request, see NCBI documentation linked to in references for a complete list

**Details**

The NCBI uses a search term syntax where search terms can be associated with a specific search field with square brackets. So, for instance “Homo[ORGN]” denotes a search for Homo in the “Organism” field. The names and definitions of these fields can be identified using [entrez\\_db\\_searchable](#).

Searches can make use of several fields by combining them via the boolean operators AND, OR and NOT. So, using the search term “((Homo[ORGN] AND APP[GENE]) NOT Review[PTYP])” in PubMed would identify articles matching the gene APP in humans, and exclude review articles. More examples of the use of these search terms, and the more specific MeSH terms for precise searching, is given in the package vignette. `rentrez` handles special characters and URL encoding (e.g. replacing spaces with plus signs) on the client side, so there is no need to include these in search term

The `entrez` tutorial provides some tips on how to make the most of searches to the NCBI. In particular, the sections on uses of the "Filter" field and MeSH terms may in formulating precise searches.

**Value**

ids integer Unique IDS returned by the search  
count integer Total number of hits for the search  
retmax integer Maximum number of hits returned by the search  
web\_history A web\_history object for use in subsequent calls to NCBI  
QueryTranslation character, search term as the NCBI interpreted it  
file either and XMLInternalDocument xml file resulting from search, parsed with `xmlTreeParse` or, if retmode was set to json a list resulting from the returned JSON file being parsed with `fromJSON`.

**References**

[https://www.ncbi.nlm.nih.gov/books/NBK25499/#\\_chapter4\\_ESearch\\_](https://www.ncbi.nlm.nih.gov/books/NBK25499/#_chapter4_ESearch_)

**See Also**

`config` for available httr configurations  
[entrez\\_db\\_searchable](#) to get a set of search fields that can be used in term for any database

**Examples**

```
## Not run:
query <- "Gastropoda[Organism] AND COI[Gene]"
web_env_search <- entrez_search(db="nucore", query, use_history=TRUE)
cookie <- web_env_search$WebEnv
qk <- web_env_search$QueryKey
snail_coi <- entrez_fetch(db = "nucore", WebEnv = cookie, query_key = qk,
                        file_format = "fasta", retmax = 10)

## End(Not run)
## Not run:

fly_id <- entrez_search(db="taxonomy", term="Drosophila")
#Oh, right. There is a genus and a subgenus name Drosophila...
#how can we limit this search
(tax_fields <- entrez_db_searchable("taxonomy"))
#"RANK" loots promising
tax_fields$RANK
entrez_search(db="taxonomy", term="Drosophila & Genus[RANK]")

## End(Not run)
```

---

entrez\_summary

*Get summaries of objects in NCBI datasets from a unique ID*


---

**Description**

The NCBI offer two distinct formats for summary documents. Version 1.0 is a relatively limited summary of a database record based on a shared Document Type Definition. Version 1.0 summaries are only available as XML and are not available for some newer databases. Version 2.0 summaries generally contain more information about a given record, but each database has its own distinct format. 2.0 summaries are available for records in all databases and as JSON and XML files. As of version 0.4, `entrez` fetches version 2.0 summaries by default and uses JSON as the exchange format (as JSON object can be more easily converted into native R types). Existing scripts which relied on the structure and naming of the "Version 1.0" summary files can be updated by setting the new version argument to "1.0".

**Usage**

```
entrez_summary(
  db,
  id = NULL,
  web_history = NULL,
  version = c("2.0", "1.0"),
  always_return_list = FALSE,
  retmode = NULL,
  config = NULL,
  ...
)
```

**Arguments**

db	character Name of the database to search for
id	vector with unique ID(s) for records in database db. In the case of sequence databases these IDs can take form of an NCBI accession followed by a version number (eg AF123456.1 or AF123456.2)
web_history	A web_history object
version	either 1.0 or 2.0 see above for description
always_return_list	logical, return a list of esummary objects even when only one ID is provided (see description for a note about this option)
retmode	either "xml" or "json". By default, xml will be used for version 1.0 records, json for version 2.0.
config	vector configuration options passed to <code>httr::GET</code>
...	character Additional terms to add to the request, see NCBI documentation linked to in references for a complete list

**Details**

By default, `entrez_summary` returns a single record when only one ID is passed and a list of such records when multiple IDs are passed. This can lead to unexpected behaviour when the results of a variable number of IDs (perhaps the result of `entrez_search`) are processed with an `apply` family function or in a for-loop. If you use this function as part of a function or script that generates a variably-sized vector of IDs setting `always_return_list` to `TRUE` will avoid these problems. The function `extract_from_esummary` is provided for the specific case of extracting named elements from a list of esummary objects, and is designed to work on single objects as well as lists.

**Value**

A list of esummary records (if multiple IDs are passed and `always_return_list` if `FALSE`) or a single record.

file `XMLInternalDocument` xml file containing the entire record returned by the NCBI.

**References**

[https://www.ncbi.nlm.nih.gov/books/NBK25499/#\\_chapter4\\_ESummary\\_](https://www.ncbi.nlm.nih.gov/books/NBK25499/#_chapter4_ESummary_)

**See Also**

[config](#) for available configs

[extract\\_from\\_esummary](#) which can be used to extract elements from a list of esummary records

**Examples**

```
## Not run:
pop_ids = c("307082412", "307075396", "307075338", "307075274")
pop_summ <- entrez_summary(db="popset", id=pop_ids)
extract_from_esummary(pop_summ, "title")
```

```
# clinvar example
res <- entrez_search(db = "clinvar", term = "BRCA1", retmax=10)
cv <- entrez_summary(db="clinvar", id=res$ids)
cv
extract_from_esummary(cv, "title", simplify=FALSE)
extract_from_esummary(cv, "trait_set")[1:2]
extract_from_esummary(cv, "gene_sort")

## End(Not run)
```

---

extract\_from\_esummary *Extract elements from a list of esummary records*

---

### Description

Extract elements from a list of esummary records

### Usage

```
extract_from_esummary(esummaries, elements, simplify = TRUE)
```

### Arguments

esummaries	Either an esummary or an esummary_list (as returned by entrez_summary).
elements	the names of the element to extract
simplify	logical, if possible return a vector

### Value

List or vector containing requested elements

### See Also

[entrez\\_summary](#) for examples of this function in action.



---

linkout_urls	<i>Extract URLs from an elink object</i>
--------------	--

---

**Description**

Extract URLs from an elink object

**Usage**

```
linkout_urls(elink)
```

**Arguments**

elink                    elink object (returned by `entrez_link`) containing Urls

**Value**

list of character vectors, one per ID each containing of URLs for that ID.

**See Also**

`entrez_link`

---

parse_pubmed_xml	<i>Summarize an XML record from pubmed.</i>
------------------	---

---

**Description**

Note: this function assumes all records are of the type "PubmedArticle" and will return an empty record for any other type (including books).

**Usage**

```
parse_pubmed_xml(record)
```

**Arguments**

record                    Either and XMLInternalDocument or character the record to be parsed ( expected to come from `entrez_fetch`)

**Value**

Either a single `pubmed_record` object, or a list of several

**Examples**

```
hox_paper <- entrez_search(db="pubmed", term="10.1038/nature08789[doi]")
hox_rel <- entrez_link(db="pubmed", dbfrom="pubmed", id=hox_paper$ids)
recs <- entrez_fetch(db="pubmed",
                    id=hox_rel$links$pubmed_pubmed[1:3],
                    rettype="xml")
parse_pubmed_xml(recs)
```

---

set_entrez_key	<i>Set the ENTREZ_KEY variable to be used by all rentrez functions</i>
----------------	--

---

**Description**

The NCBI allows users to access more records (10 per second) if they register for and use an API key. This function allows users to set this key for all calls to rentrez functions during a particular R session. See the vignette section "Using API keys" for a detailed description.

**Usage**

```
set_entrez_key(key)
```

**Arguments**

key                    character. Value to set ENTREZ\_KEY to (i.e. your API key).

**Value**

A logical of length one, TRUE is the value was set FALSE if not. value is returned inside invisible(), i.e. it is not printed to screen when the function is called.

# Index

## \* **info**

- entrez\_db\_links, [4](#)
- entrez\_db\_searchable, [5](#)
- entrez\_db\_summary, [6](#)
- entrez\_dbs, [3](#)
- entrez\_info, [9](#)

config, [3](#), [8](#), [9](#), [11–13](#), [15](#)

entrez\_citmatch, [2](#)  
entrez\_db\_links, [3](#), [4](#), [5](#), [6](#), [9](#)  
entrez\_db\_searchable, [3](#), [4](#), [5](#), [6](#), [9](#), [13](#)  
entrez\_db\_summary, [3–5](#), [6](#), [9](#)  
entrez\_dbs, [3](#), [4–6](#), [9](#)  
entrez\_fetch, [7](#), [17](#)  
entrez\_global\_query, [8](#)  
entrez\_info, [3–6](#), [9](#)  
entrez\_link, [4](#), [7](#), [10](#)  
entrez\_post, [7](#), [11](#)  
entrez\_search, [5](#), [7](#), [12](#)  
entrez\_summary, [14](#), [16](#)  
extract\_from\_esummary, [15](#), [16](#)

fromJSON, [13](#)

linkout\_urls, [17](#)

parse\_pubmed\_xml, [17](#)

set\_entrez\_key, [18](#)

xmlTreeParse, [11](#), [13](#)