

# Package ‘specieshindex’

September 30, 2021

**Type** Package

**Title** How (scientifically) popular is a given species?

**Version** 0.3.1

**Date** 2021-01-19

**Author** Jessica Tam

**Maintainer** Jessica Tam <tamtinying@gmail.com>

**Description** Finds the h-index of a species.

**Depends** R ( $\geq$  3.5.0)

**LazyData** true

**Imports** rscopus,

wosr,  
rbace,  
taxize,  
dplyr,  
tidyr,  
data.table,  
httr,  
XML,  
jsonlite,  
ggplot2,  
ggpubr

**Suggests** testthat,

knitr,  
roxygen2,  
devtools,  
rmarkdown,  
RefManageR

**License** MIT + file LICENSE

**URL** <https://github.com/jessicatytam/specieshindex>

**RoxygenNote** 7.1.2

**VignetteBuilder** knitr

**Encoding** UTF-8

**R topics documented:**

Allindices . . . . .	3
CountGenusT . . . . .	3
CountGenusTAK . . . . .	4
CountGenusTAK_base . . . . .	5
CountGenusTAK_scopus . . . . .	6
CountGenusTAK_wos . . . . .	7
CountGenusT_base . . . . .	8
CountGenusT_scopus . . . . .	9
CountGenusT_wos . . . . .	10
CountSpT . . . . .	11
CountSpTAK . . . . .	12
CountSpTAK_base . . . . .	12
CountSpTAK_scopus . . . . .	13
CountSpTAK_wos . . . . .	15
CountSpT_base . . . . .	16
CountSpT_scopus . . . . .	17
CountSpT_wos . . . . .	18
FetchGenusT . . . . .	19
FetchGenusTAK . . . . .	20
FetchGenusTAK_scopus . . . . .	20
FetchGenusTAK_wos . . . . .	21
FetchGenusT_scopus . . . . .	22
FetchGenusT_wos . . . . .	23
FetchSpT . . . . .	24
FetchSpTAK . . . . .	25
FetchSpTAK_scopus . . . . .	26
FetchSpTAK_wos . . . . .	27
FetchSpT_scopus . . . . .	28
FetchSpT_wos . . . . .	29
getYear . . . . .	30
Koala . . . . .	30
languages . . . . .	31
Platypus . . . . .	31
plotAllindices . . . . .	32
plotPub . . . . .	32
Quokka . . . . .	33
SourceType . . . . .	33
SpH5 . . . . .	34
SpHAfterdate . . . . .	34
SpHindex . . . . .	35
Spi10 . . . . .	36
SpMindex . . . . .	36
TotalCite . . . . .	37
TotalJournals . . . . .	38
TotalPub . . . . .	38
Woylie . . . . .	39
YearsPublishing . . . . .	39

---

Allindices	<i>Index summary</i>
------------	----------------------

---

**Description**

This function returns a dataframe of the summary of all of the indices.

**Usage**

```
Allindices(data, genus, species, sourcetype = 0)
```

**Arguments**

- data           The dataframe generated from [FetchSpT](#) or [FetchSpTAK](#).
- genus          Genus classification from the binomial name.
- species        Species classification from the binomial name.
- sourcetype     Source type; default is 0, enter 1 to add SourceType variables.

**Value**

A datarame of all of the indices in the package.

**Examples**

```
data(Woylie)
Allindices(Woylie, genus = "genus_name", species = "species_name", sourcetype = 0)
```

---

CountGenusT	<i>CountSpT wrapper</i>
-------------	-------------------------

---

**Description**

This is a wrapper function for [CountGenusT\\_scopus](#), [CountGenusT\\_wos](#), and [CountGenusT\\_base](#).

**Usage**

```
CountGenusT(db, genus, synonyms, additionalkeywords)
```

**Arguments**

- db            Literature database. Scopus ("scopus"), Web of Science ("wos"), or Base ("base").
- genus         Genus classification from the binomial name.
- synonyms      Alternate genus names.
- additionalkeywords   Optional search terms.

**Value**

Search count of the genus with the given `genus`.

**Examples**

```
## Not run:
CountGenusT("scopus", genus = "Osphranter")

## End(Not run)
## Not run:
CountGenusT("scopus", genus = "Osphranter", synonyms = "Macropus", additionalkeywords = "conserv*")

## End(Not run)
```

---

CountGenusTAK	<i>CountSpTAK wrapper</i>
---------------	---------------------------

---

**Description**

This is a wrapper function for [CountGenusTAK\\_scopus](#), [CountGenusTAK\\_wos](#), and [CountGenusTAK\\_base](#).

**Usage**

```
CountGenusTAK(db, genus, synonyms, additionalkeywords)
```

**Arguments**

<code>db</code>	Literature database. Scopus ("scopus"), Web of Science ("wos"), or Base ("base").
<code>genus</code>	Genus classification from the binomial name.
<code>synonyms</code>	Alternate genus names.
<code>additionalkeywords</code>	Optional search terms.

**Value**

Search count of the genus with the given `genus`.

**Examples**

```
## Not run:
CountGenusTAK("scopus", genus = "Osphranter")

## End(Not run)
## Not run:
CountGenusTAK("scopus", genus = "Osphranter", synonyms = "Macropus", additionalkeywords = "conserv*")

## End(Not run)
```

---

CountGenusTAK_base	<i>Search count from BASE - title, abstract and author keywords</i>
--------------------	---

---

## Description

This function counts the total number of search results. It counts the publications with the genus name in the title, abstract and author keywords. A check will be conducted via [gnr\\_resolve](#) to validate the genus name.

## Usage

```
CountGenusTAK_base(genus, synonyms, additionalkeywords)
```

## Arguments

genus	Genus classification from the binomial name.
synonyms	Alternate genus names.
additionalkeywords	Optional search terms.

## Value

Search count of the genus with the given genus.

## References

Chamberlain, S. & Szocs, E. (2013). taxize - taxonomic search and retrieval in R. *F1000Research*, 2, 191.

## Examples

```
## Not run:
CountGenusTAK_base("Bettongia")

#lower case letter in genus is also accepted and will return identical results

CountGenusTAK_base("bettongia")

## End(Not run)
## Not run:
CountGenusTAK_base("Bettongia", "conserv*")

#lower case letter in genus is also accepted and will return identical results

CountGenusTAK_base("bettongia", "conserv*")

## End(Not run)
```

---

CountGenusTAK\_scopus    *Search count from Scopus - title, abstract, and keywords*

---

## Description

This function counts the total number of search results. It counts the publications with the genus name in the title, abstract, and keywords. A check will be conducted via [gnr.resolve](#) to validate the genus name.

## Usage

```
CountGenusTAK_scopus(
  genus,
  synonyms,
  additionalkeywords,
  datatype = "application/xml"
)
```

## Arguments

genus	Genus classification from the binomial name.
synonyms	Alternate genus names.
additionalkeywords	Optional search terms.
datatype	Formats the URL to be sent to the API. The default is "application/xml".

## Value

Search count of the genus with the given genus.

## References

Chamberlain, S. & Szocs, E. (2013). taxize - taxonomic search and retrieval in R. *F1000Research*, 2, 191.

## Examples

```
## Not run:
CountGenusTAK_scopus("Bettongia")

#lower case letter in genus is also accepted and will return identical results

CountGenusTAK_scopus("bettongia")

## End(Not run)
## Not run:
CountGenusTAK_scopus("Bettongia")
```

```
#lower case letter in genus is also accepted and will return identical results

CountGenusTAK_scopus("bettongia")

## End(Not run)
```

---

CountGenusTAK_wos	<i>Search count from Web of Science - title, abstract and author keywords</i>
-------------------	---

---

## Description

This function counts the total number of search results. It counts the publications with the genus name in the title, abstract and author keywords. A check will be conducted via [gnr\\_resolve](#) to validate the genus name.

## Usage

```
CountGenusTAK_wos(genus, synonyms, additionalkeywords)
```

## Arguments

genus	Genus classification from the binomial name.
synonyms	Alternate genus names.
additionalkeywords	Optional search terms.

## Value

Search count of the genus with the given genus.

## References

Chamberlain, S. & Szocs, E. (2013). taxize - taxonomic search and retrieval in R. *F1000Research*, 2, 191.

## Examples

```
## Not run:
CountGenusTAK_wos("Bettongia")

#lower case letter in genus is also accepted and will return identical results

CountGenusTAK_wos("bettongia")

## End(Not run)
## Not run:
CountGenusTAK_wos("Bettongia", "conserv*")
```

```
#lower case letter in genus is also accepted and will return identical results

CountGenusTAK_wos("bettongia", "conserv*")

## End(Not run)
```

---

CountGenusT_base	<i>Search count from BASE - title only</i>
------------------	--

---

## Description

This function counts the total number of search results. It counts the publications with the genus name in the title only. A check will be conducted via [gnr\\_resolve](#) to validate the genus name.

## Usage

```
CountGenusT_base(genus, synonyms, additionalkeywords)
```

## Arguments

genus	Genus classification from the binomial name.
synonyms	Alternate genus names.
additionalkeywords	Optional search terms.

## Value

Search count of the genus with the given genus.

## References

Chamberlain, S. & Szocs, E. (2013). taxize - taxonomic search and retrieval in R. *F1000Research*, 2, 191.

## Examples

```
## Not run:
CountGenusT_base("Bettongia")

#lower case letter in genus is also accepted and will return identical results

CountGenusT_base("bettongia")

## End(Not run)
## Not run:
CountGenusT_base("Bettongia", "conserv*")

#lower case letter in genus is also accepted and will return identical results
```



```
CountGenusT_base("bettongia", "conserv*")

## End(Not run)
```

---

CountGenusT_scopus	<i>Search count from Scopus - title only</i>
--------------------	--

---

## Description

This function counts the total number of search results. It counts the publications with the genus name in the title only. A check will be conducted via [gnr\\_resolve](#) to validate the genus name.

## Usage

```
CountGenusT_scopus(
  genus,
  synonyms,
  additionalkeywords,
  datatype = "application/xml"
)
```

## Arguments

genus	Genus classification from the binomial name.
synonyms	Alternate genus names.
additionalkeywords	Optional search terms.
datatype	Formats the URL to be sent to the API. The default is "application/xml".

## Value

Search count of the genus with the given genus.

## References

Chamberlain, S. & Szocs, E. (2013). taxize - taxonomic search and retrieval in R. *F1000Research*, 2, 191.

## Examples

```
## Not run:
CountGenusT_scopus("Bettongia")

#lower case letter in genus is also accepted and will return identical results

CountGenusT_scopus("bettongia")
```

```
## End(Not run)
## Not run:
CountGenusT_scopus("Bettongia", "conserv*")

#lower case letter in genus is also accepted and will return identical results

CountGenusT_scopus("bettongia", "conserv*")

## End(Not run)
```

---

CountGenusT\_wos

*Search count from Web of Science - title only*


---

## Description

This function counts the total number of search results. It counts the publications with the genus name in the title only. A check will be conducted via [gnr\\_resolve](#) to validate the genus name.

## Usage

```
CountGenusT_wos(genus, synonyms, additionalkeywords)
```

## Arguments

genus	Genus classification from the binomial name.
synonyms	Alternate genus names.
additionalkeywords	Optional search terms.

## Value

Search count of the genus with the given genus.

## References

Chamberlain, S. & Szocs, E. (2013). taxize - taxonomic search and retrieval in R. *F1000Research*, 2, 191.

## Examples

```
## Not run:
CountGenusT_wos("Bettongia")

#lower case letter in genus is also accepted and will return identical results

CountGenusT_wos("bettongia")
```

```
## End(Not run)
## Not run:
CountGenusT_wos("Bettongia", "conserv*")

#lower case letter in genus is also accepted and will return identical results

CountGenusT_wos("bettongia", "conserv*")

## End(Not run)
```

---

CountSpT	<i>CountSpT wrapper</i>
----------	-------------------------

---

## Description

This is a wrapper function for [CountSpT.scopus](#), [CountSpT.wos](#), and [CountSpT.base](#).

## Usage

```
CountSpT(db, genus, species, synonyms, additionalkeywords)
```

## Arguments

db	Literature database. Scopus ("scopus"), Web of Science ("wos"), or Base ("base").
genus	Genus classification from the binomial name.
species	Species classification from the binomial name.
synonyms	Alternate species names.
additionalkeywords	Optional search terms.

## Value

Search count of the species with the given genus and species.

## Examples

```
## Not run:
CountSpT("scopus", genus = "Osphranter", species = "rufus")

## End(Not run)
## Not run:
CountSpT("scopus", genus = "Osphranter", species = "rufus", synonyms = "Macropus rufus", additionalkeywords = "conserv*")

## End(Not run)
```

---

CountSpTAK	<i>CountSpTAK wrapper</i>
------------	---------------------------

---

**Description**

This is a wrapper function for [CountSpTAK\\_scopus](#), [CountSpTAK\\_wos](#), and [CountSpTAK\\_base](#).

**Usage**

CountSpTAK(db, genus, species, synonyms, additionalkeywords)

**Arguments**

- db Literature database. Scopus ("scopus"), Web of Science ("wos"), or Base ("base").
- genus Genus classification from the binomial name.
- species Species classification from the binomial name.
- synonyms Alternate species names.
- additionalkeywords Optional search terms.

**Value**

Search count of the species with the given genus and species.

**Examples**

```
## Not run:
CountSpTAK("scopus", genus = "Osphranter", species = "rufus")

## End(Not run)
## Not run:
CountSpTAK("scopus", genus = "Osphranter", species = "rufus", synonyms = "Macropus rufus", additionalkeywords = "c")

## End(Not run)
```

---

CountSpTAK_base	<i>Search count from BASE - title, abstract and author keywords</i>
-----------------	---

---

**Description**

This function counts the total number of search results. It counts the publications with the binomial name in the title, abstract and author keywords. A check will be conducted via [gnr\\_resolve](#) to validate the genus and species names.

**Usage**

```
CountSpTAK_base(genus, species, synonyms, additionalkeywords)
```

**Arguments**

genus	Genus classification from the binomial name.
species	Species classification from the binomial name.
synonyms	Alternate species names.
additionalkeywords	Optional search terms.

**Value**

Search count of the species with the given `genus` and `species`.

**References**

Chamberlain, S. & Szocs, E. (2013). taxize - taxonomic search and retrieval in R. *F1000Research*, 2, 191.

**Examples**

```
## Not run:
CountSpTAK_base("Bettongia", "penicillata")

#lower case letter in genus is also accepted and will return identical results

CountSpTAK_base("bettongia", "penicillata")

## End(Not run)
## Not run:
CountSpTAK_base("Bettongia", "penicillata", "conserv*")

#lower case letter in genus is also accepted and will return identical results

CountSpTAK_base("bettongia", "penicillata", "conserv*")

## End(Not run)
```

---

CountSpTAK_scopus	<i>Search count from Scopus - title, abstract, and keywords</i>
-------------------	---

---

**Description**

This function counts the total number of search results. It counts the publications with the binomial name in the title, abstract, and keywords. A check will be conducted via [gnr\\_resolve](#) to validate the genus and species names.

**Usage**

```
CountSpTAK_scopus(
  genus,
  species,
  synonyms,
  additionalkeywords,
  datatype = "application/xml"
)
```

**Arguments**

<code>genus</code>	Genus classification from the binomial name.
<code>species</code>	Species classification from the binomial name.
<code>synonyms</code>	Alternate species names.
<code>additionalkeywords</code>	Optional search terms.
<code>datatype</code>	Formats the URL to be sent to the API. The default is "application/xml".

**Value**

Search count of the species with the given `genus` and `species`.

**References**

Chamberlain, S. & Szocs, E. (2013). taxize - taxonomic search and retrieval in R. *F1000Research*, 2, 191.

**Examples**

```
## Not run:
CountSpTAK_scopus("Bettongia", "penicillata")

#lower case letter in genus is also accepted and will return identical results

CountSpTAK_scopus("bettongia", "penicillata")

## End(Not run)
## Not run:
CountSpTAK_scopus("Bettongia", "penicillata")

#lower case letter in genus is also accepted and will return identical results

CountSpTAK_scopus("bettongia", "penicillata")

## End(Not run)
```

---

CountSpTAK_wos	<i>Search count from Web of Science - title, abstract and author keywords</i>
----------------	---

---

## Description

This function counts the total number of search results. It counts the publications with the binomial name in the title, abstract and author keywords. A check will be conducted via [gnr\\_resolve](#) to validate the genus and species names.

## Usage

```
CountSpTAK_wos(genus, species, synonyms, additionalkeywords)
```

## Arguments

genus	Genus classification from the binomial name.
species	Species classification from the binomial name.
synonyms	Alternate species names.
additionalkeywords	Optional search terms.

## Value

Search count of the species with the given genus and species.

## References

Chamberlain, S. & Szocs, E. (2013). taxize - taxonomic search and retrieval in R. *F1000Research*, 2, 191.

## Examples

```
## Not run:
CountSpTAK_wos("Bettongia", "penicillata")

#lower case letter in genus is also accepted and will return identical results

CountSpTAK_wos("bettongia", "penicillata")

## End(Not run)
## Not run:
CountSpTAK_wos("Bettongia", "penicillata", "conserv*")

#lower case letter in genus is also accepted and will return identical results

CountSpTAK_wos("bettongia", "penicillata", "conserv*")

## End(Not run)
```

---

CountSpT_base	<i>Search count from BASE - title only</i>
---------------	--

---

## Description

This function counts the total number of search results. It counts the publications with the binomial name in the title only. A check will be conducted via [gnr.resolve](#) to validate the genus and species names.

## Usage

```
CountSpT_base(genus, species, synonyms, additionalkeywords)
```

## Arguments

genus	Genus classification from the binomial name.
species	Species classification from the binomial name.
synonyms	Alternate species names.
additionalkeywords	Optional search terms.

## Value

Search count of the species with the given genus and species.

## References

Chamberlain, S. & Szocs, E. (2013). taxize - taxonomic search and retrieval in R. *F1000Research*, 2, 191.

## Examples

```
## Not run:
CountSpT_base("Bettongia", "penicillata")

#lower case letter in genus is also accepted and will return identical results

CountSpT_base("bettongia", "penicillata")

## End(Not run)
## Not run:
CountSpT_base("Bettongia", "penicillata", "conserv*")

#lower case letter in genus is also accepted and will return identical results

CountSpT_base("bettongia", "penicillata", "conserv*")

## End(Not run)
```



---

CountSpT_scopus	<i>Search count from Scopus - title only</i>
-----------------	--

---

## Description

This function counts the total number of search results. It counts the publications with the binomial name in the title only. A check will be conducted via [gnr.resolve](#) to validate the genus and species names.

## Usage

```
CountSpT_scopus(
  genus,
  species,
  synonyms,
  additionalkeywords,
  datatype = "application/xml"
)
```

## Arguments

genus	Genus classification from the binomial name.
species	Species classification from the binomial name.
synonyms	Alternate species names.
additionalkeywords	Optional search terms.
datatype	Formats the URL to be sent to the API. The default is "application/xml".

## Value

Search count of the species with the given `genus` and `species`.

## References

Chamberlain, S. & Szocs, E. (2013). taxize - taxonomic search and retrieval in R. *F1000Research*, 2, 191.

## Examples

```
## Not run:
CountSpT_scopus("Bettongia", "penicillata")

#lower case letter in genus is also accepted and will return identical results

CountSpT_scopus("bettongia", "penicillata")

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

```
CountSpT_scopus("Bettongia", "penicillata", "conserv*")

#lower case letter in genus is also accepted and will return identical results

CountSpT_scopus("bettongia", "penicillata", "conserv*")

## End(Not run)
```

---

CountSpT\_wos

*Search count from Web of Science - title only*


---

## Description

This function counts the total number of search results. It counts the publications with the binomial name in the title only. A check will be conducted via [gnr\\_resolve](#) to validate the genus and species names.

## Usage

```
CountSpT_wos(genus, species, synonyms, additionalkeywords)
```

## Arguments

genus	Genus classification from the binomial name.
species	Species classification from the binomial name.
synonyms	Alternate species names.
additionalkeywords	Optional search terms.

## Value

Search count of the species with the given genus and species.

## References

Chamberlain, S. & Szocs, E. (2013). taxize - taxonomic search and retrieval in R. *F1000Research*, 2, 191.

## Examples

```
## Not run:
CountSpT_wos("Bettongia", "penicillata")

#lower case letter in genus is also accepted and will return identical results

CountSpT_wos("bettongia", "penicillata")

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

```

CountSpT_wos("Bettongia", "penicillata", "conserv*")

#lower case letter in genus is also accepted and will return identical results

CountSpT_wos("bettongia", "penicillata", "conserv*")

## End(Not run)

```

---

FetchGenusT

*FetchGenusT wrapper*


---

## Description

This is a wrapper function for [FetchSpT\\_scopus](#), and [FetchSpT\\_wos](#).

## Usage

```
FetchGenusT(db, genus, synonyms, additionalkeywords, language = 0)
```

## Arguments

db	Literature database. Scopus ("scopus") or Web of Science ("wos").
genus	Genus classification from the binomial name.
synonyms	Alternate genus names.
additionalkeywords	Optional search terms.
language	Language of the paper; default is 0, enter 1 to retrieve the variable. Scopus only.

## Value

A dataframe of the genus' citation records with the given genus.

## Examples

```

## Not run:
FetchGenusT("scopus", genus = "Osphranter")

## End(Not run)
## Not run:
FetchGenusT("scopus", genus = "Osphranter", synonyms = "Macropus", additionalkeywords = "conserv*")

## End(Not run)

```

---

FetchGenusTAK	<i>FetchGenusTAK wrapper</i>
---------------	------------------------------

---

## Description

This is a wrapper function for [FetchGenusTAK\\_scopus](#), and [FetchGenusTAK\\_wos](#).

## Usage

```
FetchGenusTAK(db, genus, synonyms, additionalkeywords, language = 0)
```

## Arguments

db	Literature database. Scopus ("scopus") or Web of Science ("wos").
genus	Genus classification from the binomial name.
synonyms	Alternate genus names.
additionalkeywords	Optional search terms.
language	Language of the paper; default is 0, enter 1 to retrieve the variable. Scopus only.

## Value

A dataframe of the genus' citation records with the given genus.

## Examples

```
## Not run:
FetchGenusTAK("scopus", genus = "Osphranter")

## End(Not run)
## Not run:
FetchGenusTAK("scopus", genus = "Osphranter", synonyms = "Macropus", additionalkeywords = "conserv*")

## End(Not run)
```

---

FetchGenusTAK_scopus	<i>Fetch data from Scopus - title, abstract and keywords</i>
----------------------	--

---

## Description

This function fetches citation information from Scopus using genus name found in the title, abstract and keywords of the publications. Duplicates are to be removed by the user after fetching the data.

**Usage**

```
FetchGenusTAK_scopus(genus, synonyms, additionalkeywords, language = 0)
```

**Arguments**

genus	Genus classification from the binomial name.
synonyms	Alternate genus names.
additionalkeywords	Optional search terms.
language	Language of the paper; default is 0, enter 1 to retrieve the variable.

**Value**

A dataframe of the genus' citation records with the given genus.

**Examples**

```
## Not run:
FetchGenusTAK_scopus("Bettongia")

#lower case letter in genus is also accepted and will return identical results

FetchGenusTAK_scopus("bettongia")

## End(Not run)
## Not run:
FetchGenusTAK_scopus("Bettongia", "conserv*")

#lower case letter in genus is also accepted and will return identical results

FetchGenusTAK_scopus("bettongia", "conserv*")

## End(Not run)
```

---

FetchGenusTAK_wos	<i>Fetch data from Web of Science - title, abstract and author keywords.</i>
-------------------	--

---

**Description**

This function fetches citation information from Web of Science using genus name found in the title, abstract and author keywords of the publications. Duplicates are to be removed by the user after fetching the data.

**Usage**

```
FetchGenusTAK_wos(genus, synonyms, additionalkeywords)
```

**Arguments**

genus	Genus classification from the binomial name.
synonyms	Alternate genus names.
additionalkeywords	Optional search terms.

**Value**

A dataframe of the genus' citation records with the given genus.

**Examples**

```
## Not run:
FetchGenusTAK_wos("Bettongia")

#lower case letter in genus is also accepted and will return identical results

FetchGenusTAK_wos("bettongia")

## End(Not run)
## Not run:
FetchGenusTAK_wos("Bettongia", "conserv*")

#lower case letter in genus is also accepted and will return identical results

FetchGenusTAK_wos("bettongia", "conserv*")

## End(Not run)
```

---

FetchGenusT_scopus	<i>Fetch data from Scopus - title only</i>
--------------------	--

---

**Description**

This function fetches citation information from Scopus using genus name found in the title of the publications. Duplicates are to be removed by the user after fetching the data.

**Usage**

```
FetchGenusT_scopus(genus, synonyms, additionalkeywords, language = 0)
```

**Arguments**

genus	Genus classification from the binomial name.
synonyms	Alternate genus names.
additionalkeywords	Optional search terms.
language	Language of the paper; default is 0, enter 1 to retrieve the variable.

**Value**

A dataframe of the genus' citation records with the given genus.

**Examples**

```
## Not run:
FetchGenusT_scopus("Bettongia")

#lower case letter in genus is also accepted and will return identical results

FetchGenusT_scopus("bettongia")

## End(Not run)
## Not run:
FetchGenusT_scopus("Bettongia", "conserv*")

#lower case letter in genus is also accepted and will return identical results

FetchGenusT_scopus("bettongia", "conserv*")

## End(Not run)
```

---

FetchGenusT_wos	<i>Fetch data from Web of Science - title only</i>
-----------------	--

---

**Description**

This function fetches citation information from Web of Science using genus name found in the title of the publications. Duplicates are to be removed by the user after fetching the data.

**Usage**

```
FetchGenusT_wos(genus, synonyms, additionalkeywords)
```

**Arguments**

genus	Genus classification from the binomial name.
synonyms	Alternate genus names.
additionalkeywords	Optional search terms.

**Value**

A dataframe of the genus' citation records with the given genus.

## Examples

```
## Not run:
FetchGenusT_wos("Bettongia")

#lower case letter in genus is also accepted and will return identical results

FetchGenusT_wos("bettongia")

## End(Not run)
## Not run:
FetchGenusT_wos("Bettongia", "conserv*")

#lower case letter in genus is also accepted and will return identical results

FetchGenusT_wos("bettongia", "conserv*")

## End(Not run)
```

---

FetchSpT

*FetchSpT wrapper*


---

## Description

This is a wrapper function for [FetchSpT.scopus](#), and [FetchSpT.wos](#).

## Usage

```
FetchSpT(db, genus, species, synonyms, additionalkeywords, language = 0)
```

## Arguments

db	Literature database. Scopus ("scopus") or Web of Science ("wos").
genus	Genus classification from the binomial name.
species	Species classification from the binomial name.
synonyms	Alternate species names.
additionalkeywords	Optional search terms.
language	Language of the paper; default is 0, enter 1 to retrieve the variable. Scopus only.

## Value

A dataframe of the species' citation records with the given `genus` and `species`.



Examples

```
## Not run:
FetchSpT("scopus", genus = "Osphranter", species = "rufus")

## End(Not run)
## Not run:
FetchSpT("scopus", genus = "Osphranter", species = "rufus", synonyms = "Macropus rufus", additionalkeywords = "conservation")

## End(Not run)
```

---

FetchSpTAK	<i>FetchSpTAK wrapper</i>
------------	---------------------------

---

Description

This is a wrapper function for [FetchSpTAK\\_scopus](#), and [FetchSpTAK\\_wos](#).

Usage

```
FetchSpTAK(db, genus, species, synonyms, additionalkeywords, language = 0)
```

Arguments

- db Literature database. Scopus ("scopus") or Web of Science ("wos").
- genus Genus classification from the binomial name.
- species Species classification from the binomial name.
- synonyms Alternate species names.
- additionalkeywords Optional search terms.
- language Language of the paper; default is 0, enter 1 to retrieve the variable. Scopus only.

Value

A dataframe of the species' citation records with the given genus and species.

Examples

```
## Not run:
FetchSpTAK("scopus", genus = "Osphranter", species = "rufus")

## End(Not run)
## Not run:
FetchSpTAK("scopus", genus = "Osphranter", species = "rufus", synonyms = "Macropus rufus", additionalkeywords = "conservation")

## End(Not run)
```

---

FetchSpTAK\_scopus

*Fetch data from Scopus - title, abstract and keywords*


---

## Description

This function fetches citation information from Scopus using genus and species name found in the title, abstract and keywords of the publications. Duplicates are to be removed by the user after fetching the data.

## Usage

```
FetchSpTAK_scopus(genus, species, synonyms, additionalkeywords, language = 0)
```

## Arguments

genus	Genus classification from the binomial name.
species	Species classification from the binomial name.
synonyms	Alternate species names.
additionalkeywords	Optional search terms.
language	Language of the paper; default is 0, enter 1 to retrieve the variable.

## Value

A dataframe of the species' citation records with the given genus and species.

## Examples

```
## Not run:
FetchSpTAK_scopus("Bettongia", "penicillata")

#lower case letter in genus is also accepted and will return identical results

FetchSpTAK_scopus("bettongia", "penicillata")

## End(Not run)
## Not run:
FetchSpTAK_scopus("Bettongia", "penicillata", "conserv*")

#lower case letter in genus is also accepted and will return identical results

FetchSpTAK_scopus("bettongia", "penicillata", "conserv*")

## End(Not run)
```

---

FetchSpTAK_wos	<i>Fetch data from Web of Science - title, abstract and author keywords.</i>
----------------	--

---

## Description

This function fetches citation information from Web of Science using genus and species name found in the title, abstract and author keywords of the publications. Duplicates are to be removed by the user after fetching the data.

## Usage

```
FetchSpTAK_wos(genus, species, synonyms, additionalkeywords)
```

## Arguments

genus	Genus classification from the binomial name.
species	Species classification from the binomial name.
synonyms	Alternate species names.
additionalkeywords	Optional search terms.

## Value

A dataframe of the species' citation records with the given `genus` and `species`.

## Examples

```
## Not run:
FetchSpTAK_wos("Bettongia", "penicillata")

#lower case letter in genus is also accepted and will return identical results

FetchSpTAK_wos("bettongia", "penicillata")

## End(Not run)
## Not run:
FetchSpTAK_wos("Bettongia", "penicillata", "conserv*")

#lower case letter in genus is also accepted and will return identical results

FetchSpTAK_wos("bettongia", "penicillata", "conserv*")

## End(Not run)
```

---

FetchSpT\_scopus

*Fetch data from Scopus - title only*


---

## Description

This function fetches citation information from Scopus using genus and species name found in the title of the publications. Duplicates are to be removed by the user after fetching the data.

## Usage

```
FetchSpT_scopus(genus, species, synonyms, additionalkeywords, language = 0)
```

## Arguments

genus	Genus classification from the binomial name.
species	Species classification from the binomial name.
synonyms	Alternate species names.
additionalkeywords	Optional search terms.
language	Language of the paper; default is 0, enter 1 to retrieve the variable.

## Value

A dataframe of the species' citation records with the given genus and species.

## Examples

```
## Not run:
FetchSpT_scopus("Bettongia", "penicillata")

#lower case letter in genus is also accepted and will return identical results

FetchSpT_scopus("bettongia", "penicillata")

## End(Not run)
## Not run:
FetchSpT_scopus("Bettongia", "penicillata", "conserv*")

#lower case letter in genus is also accepted and will return identical results

FetchSpT_scopus("bettongia", "penicillata", "conserv*")

## End(Not run)
```

---

FetchSpT\_wos*Fetch data from Web of Science - title only*

---

## Description

This function fetches citation information from Web of Science using genus and species name found in the title of the publications. Duplicates are to be removed by the user after fetching the data.

## Usage

```
FetchSpT_wos(genus, species, synonyms, additionalkeywords)
```

## Arguments

genus	Genus classification from the binomial name.
species	Species classification from the binomial name.
synonyms	Alternate species names.
additionalkeywords	Optional search terms.

## Value

A dataframe of the species' citation records with the given genus and species.

## Examples

```
## Not run:
FetchSpT_wos("Bettongia", "penicillata")

#lower case letter in genus is also accepted and will return identical results

FetchSpT_wos("bettongia", "penicillata")

## End(Not run)
## Not run:
FetchSpT_wos("Bettongia", "penicillata", "conserv*")

#lower case letter in genus is also accepted and will return identical results

FetchSpT_wos("bettongia", "penicillata", "conserv*")

## End(Not run)
```

---

getYear	<i>Extract year</i>
---------	---------------------

---

**Description**

Extracts the year of each publication of the output from any of the Fetch functions and counts the number of publications each year.

**Usage**

```
getYear(data, genus, species)
```

**Arguments**

data	Output from any of the fetch function.
genus	Genus classification from the binomial name.
species	Species classification from the binomial name.

**Value**

A dataframe with the year and frequency of the publications

**Examples**

```
getYear(data = Woylie, genus = "Bettongia", species = "penicillata")
```

---

Koala	<i>Koala dataset</i>
-------	----------------------

---

**Description**

Citation records of koala (*Phascolarctos cinereus*) from Scopus. Data was retrieved on 10 July 2020.

**Usage**

```
Koala
```

**Format**

A data frame with 773 rows and 20 variables

**Source**

<http://api.elsevier.com/content/search/scopus>

---

languages	<i>Languages</i>
-----------	------------------

---

**Description**

List of languages of documents found on Scopus. Data was retrieved on 9 March 2021.

**Usage**

languages

**Format**

A csv file with the complete list of languages of documents found on Scopus.

**Source**

<https://www.elsevier.com/solutions/scopus/how-scopus-works/content>

---

Platypus	<i>Platypus dataset</i>
----------	-------------------------

---

**Description**

Citation records of platypus (*Ornithorhynchus anatinus*) from Scopus. Data was retrieved on 10 July 2020.

**Usage**

Platypus

**Format**

A data frame with 321 rows and 20 variables

**Source**

<http://api.elsevier.com/content/search/scopus>

---

plotAllindices	<i>Index plot</i>
----------------	-------------------

---

**Description**

Plots the indices of a single species or combined.

**Usage**

```
plotAllindices(data)
```

**Arguments**

data                      The dataframe generated from [Allindices](#).

**Value**

ggplot

**Examples**

```
W <- Allindices(Woylie, genus = "Bettongia", species = "penicillata")
Q <- Allindices(Quokka, genus = "Setonix", species = "brachyurus")
P <- Allindices(Platypus, genus = "Ornithorhynchus", species = "anatinus")
K <- Allindices(Koala, genus = "Phascolarctos", species = "cinereus")
CombineSp <- dplyr::bind_rows(W, Q, P, K)
plotAllindices(CombineSp)
```

---

plotPub	<i>Publication plot</i>
---------	-------------------------

---

**Description**

Plots the publication by year of a single species or combined.

**Usage**

```
plotPub(data)
```

**Arguments**

data                      The dataframe generated from [getYear](#).

**Value**

ggplot



**Examples**

```
extract_year_W <- getYear(data = Woylie, genus = "Bettongia", species = "penicillata")
extract_year_Q <- getYear(data = Quokka, genus = "Setonix", species = "brachyurus")
extract_year_P <- getYear(data = Platypus, genus = "Ornithorhynchus", species = "anatinus")
extract_year_K <- getYear(data = Koala, genus = "Phascolarctos", species = "cinereus")
Combine_pub <- rbind(extract_year_W, extract_year_Q, extract_year_P, extract_year_K)
plotPub(Combine_pub)
```

---

Quokka	<i>Quokka dataset</i>
--------	-----------------------

---

**Description**

Citation records of quokka (*Setonix brachyurus*) from Scopus. Data was retrieved on 10 July 2020.

**Usage**

```
Quokka
```

**Format**

A data frame with 242 rows and 20 variables

**Source**

<http://api.elsevier.com/content/search/scopus>

---

SourceType	<i>Source type</i>
------------	--------------------

---

**Description**

This function calculates the total number of items for each document type.

**Usage**

```
SourceType(data)
```

**Arguments**

**data** The dataframe generated from [FetchSpT](#) or [FetchSpTAK](#).

**Value**

A dataframe with each document and their counts.

**Examples**

```
data(Woylie)
SourceType(Woylie)
```

---

SpH5

*Species h5 index*


---

**Description**

This function calculates the h-index of a species in the past 5 years.

**Usage**

```
SpH5(data)
```

**Arguments**

`data`                      The dataframe generated from [FetchSpT](#) or [FetchSpTAK](#).

**Value**

H5 index.

**References**

Suzuki, H. (2012). *Google Scholar Metrics for Publications*. Retrieved from <https://scholar.googleblog.com/2012/04/google-scholar-metrics-for-publications.html>.

**Examples**

```
data(Woylie)
SpH5(Woylie)
```

---

SpHAfterdate

*Species h-index with a given time frame*


---

**Description**

This function calculates the h-index using a given date up till the newest record.

**Usage**

```
SpHAfterdate(data, date)
```

**Arguments**

**data**                    The dataframe generated from [FetchSpT](#) or [FetchSpTAK](#).  
**date**                    The lower limit of the timeframe.

**Value**

H-index of the given time period.

**Examples**

```
data(Woylie)
SpHAfterdate(Woylie, "2000-01-01")
```

---

SpHindex	<i>Species h-index</i>
----------	------------------------

---

**Description**

This function calculates the h-index of a species.

**Usage**

```
SpHindex(data)
```

**Arguments**

**data**                    The dataframe generated from [FetchSpT](#) or [FetchSpTAK](#).

**Value**

H-index.

**References**

Bertoli-Barsotti, L. & Lando, T. (2015). On a formula for the h-index. *Journal of Informetrics*, 9(4), 762-776.  
Hirsch, J. (2005). An index to quantify an individual's scientific research output. *Proceedings of the National Academy of Sciences of the United States of America*, 102(46), 16569-16572.

**Examples**

```
data(Woylie)
SpHindex(Woylie)
```

Spi10

*Species i10 index*

---

**Description**

This function calculates the i10 index of a species. i10 index counts all of the publications with 10 or more citations.

**Usage**

```
Spi10(data)
```

**Arguments**

data                      The dataframe generated from [FetchSpT](#) or [FetchSpTAK](#).

**Value**

i10 index.

**References**

Cornell University (2019). *i10-index*. Retrieved from <https://guides.library.cornell.edu/c.php?g=32272&p=203393>.

**Examples**

```
data(Woylie)
Spi10(Woylie)
```

---

SpMindex*Species m-index*

---

**Description**

This function calculates the m-index of species. M-index uses the h-index and divides it by the number of years of activity.

**Usage**

```
SpMindex(data)
```

**Arguments**

data                      The dataframe generated from [FetchSpT](#) or [FetchSpTAK](#).

**Value**

M-index.

**References**

University of Pittsburgh (2019). *Research Impact and Metrics: Author metrics*. Retrieved from <https://pitt.libguides.com/bibliometricIndicators/AuthorMetrics>.

**Examples**

```
data(Woylie)
SpMindex(Woylie)
```

---

TotalCite	<i>Total citations</i>
-----------	------------------------

---

**Description**

This function calculates the total number of citations.

**Usage**

```
TotalCite(data)
```

**Arguments**

**data**                      The dataframe generated from [FetchSpT](#) or [FetchSpTAK](#).

**Value**

A numerical value of the total number of citations.

**Examples**

```
data(Woylie)
TotalCite(Woylie)
```

---

TotalJournals	<i>Total journals</i>
---------------	-----------------------

---

**Description**

This function calculates the total number of journals.

**Usage**

```
TotalJournals(data)
```

**Arguments**

data                      The dataframe generated from [FetchSpT](#) or [FetchSpTAK](#).

**Value**

An integer of the total number of journals.

**Examples**

```
data(Woylie)
TotalJournals(Woylie)
```

---

TotalPub	<i>Total publications</i>
----------	---------------------------

---

**Description**

This function calculates the total number of publications.

**Usage**

```
TotalPub(data)
```

**Arguments**

data                      The dataframe generated from [FetchSpT](#) or [FetchSpTAK](#).

**Value**

An integer of the total number of publications.

**Examples**

```
data(Woylie)
TotalPub(Woylie)
```

---

Woylie

*Woylie dataset*

---

**Description**

Citation records of woylie (*Bettongia penicillata*) from Scopus. Data was retrieved on 10 July 2020.

**Usage**

Woylie

**Format**

A data frame with 113 rows and 20 variables

**Source**

<http://api.elsevier.com/content/search/scopus>

---

YearsPublishing

*Years since first publication*

---

**Description**

The number of years since the first publication in relation to the species.

**Usage**

YearsPublishing(data)

**Arguments**

data                      The dataframe generated from [FetchSpT](#) or [FetchSpTAK](#).

**Value**

Number of years.

**Examples**

```
data(Woylie)
YearsPublishing(Woylie)
```

# Index

## \* datasets

- Koala, [30](#)
- languages, [31](#)
- Platypus, [31](#)
- Quokka, [33](#)
- Woylie, [39](#)

Allindices, [3](#), [32](#)

CountGenusT, [3](#)  
CountGenusT\_base, [3](#), [8](#)  
CountGenusT\_scopus, [3](#), [9](#)  
CountGenusT\_wos, [3](#), [10](#)  
CountGenusTAK, [4](#)  
CountGenusTAK\_base, [4](#), [5](#)  
CountGenusTAK\_scopus, [4](#), [6](#)  
CountGenusTAK\_wos, [4](#), [7](#)  
CountSpT, [11](#)  
CountSpT\_base, [11](#), [16](#)  
CountSpT\_scopus, [11](#), [17](#)  
CountSpT\_wos, [11](#), [18](#)  
CountSpTAK, [12](#)  
CountSpTAK\_base, [12](#), [12](#)  
CountSpTAK\_scopus, [12](#), [13](#)  
CountSpTAK\_wos, [12](#), [15](#)

FetchGenusT, [19](#)  
FetchGenusT\_scopus, [22](#)  
FetchGenusT\_wos, [23](#)  
FetchGenusTAK, [20](#)  
FetchGenusTAK\_scopus, [20](#), [20](#)  
FetchGenusTAK\_wos, [20](#), [21](#)  
FetchSpT, [3](#), [24](#), [33–39](#)  
FetchSpT\_scopus, [19](#), [24](#), [28](#)  
FetchSpT\_wos, [19](#), [24](#), [29](#)  
FetchSpTAK, [3](#), [25](#), [33–39](#)  
FetchSpTAK\_scopus, [25](#), [26](#)  
FetchSpTAK\_wos, [25](#), [27](#)

getYear, [30](#), [32](#)

gnr\_resolve, [5–10](#), [12](#), [13](#), [15–18](#)

Koala, [30](#)

languages, [31](#)

Platypus, [31](#)  
plotAllindices, [32](#)  
plotPub, [32](#)

Quokka, [33](#)

SourceType, [33](#)  
SpH5, [34](#)  
SpHAfterdate, [34](#)  
SpHindex, [35](#)  
Spi10, [36](#)  
SpMindex, [36](#)

TotalCite, [37](#)  
TotalJournals, [38](#)  
TotalPub, [38](#)

Woylie, [39](#)

YearsPublishing, [39](#)