

# Package ‘taxotools’

May 14, 2019

**Type** Package

**Title** Tools to Handle Taxonomic Lists

**Version** 0.0.29

**Date** 2019-05-13

**Maintainer** Vijay Barve <vijay.barve@gmail.com>

**Description** Some tools to work with taxonomic name lists.

**License** CC0

**Imports** taxize, wikitaxa, plyr, sqldf

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.1.1

**BugReports** <https://github.com/vijaybarve/taxotools/issues>

**NeedsCompilation** no

**Author** Vijay Barve [aut, cre] (<<https://orcid.org/0000-0002-4852-2567>>)

**Repository** CRAN

**Date/Publication** 2019-05-13 22:00:03 UTC

## R topics documented:

BuildCSList . . . . .	2
cast_canonical . . . . .	3
cast_cs_field . . . . .	4
CheckScientificName . . . . .	5
check_scientific . . . . .	6
ExpandSyn . . . . .	6
expand_name . . . . .	7
GetITISyn . . . . .	8
GetWikiSyn . . . . .	8
get_itis_syn . . . . .	9
GuessTaxoLevel . . . . .	10

guess_taxo_level . . . . .	11
ListITISyn . . . . .	11
list_higher_taxo . . . . .	12
list_itis_syn . . . . .	13
list_wiki_syn . . . . .	14
MakeCanonical . . . . .	15
make_canonical . . . . .	16
MatchLists . . . . .	17
match_lists . . . . .	18
melt_canonical . . . . .	18
melt_cs_field . . . . .	19
taxotools . . . . .	20

<b>Index</b>	<b>22</b>
--------------	-----------

---

BuildCSList	<i>Build a comma separated List</i>
-------------	-------------------------------------

---

## Description

This function is depreciated. Please use [cast\\_cs\\_field](#)

## Usage

```
BuildCSList(data, pri, sec, duplicate = FALSE, sepchar = ",")
```

## Arguments

data	data frame containing primary and secondary data columns
pri	Primary field name (repeating values)
sec	Secondary field (values would be added to same record, comma separated)
duplicate	If true, duplicate entries are allowed in secondary field
sepchar	Character separator between the data items. Default is comma

## Details

Builds a comma separated list given a data frame with two fiends, primary repeating values and secondary values to be summarized with comma separated in the same field

## Value

a data frame with two fields Primary and secondary (comma separated list)

## See Also

Other Discontinued functions: [CheckScientificName](#), [ExpandSyn](#), [GetWikiSyn](#), [GuessTaxoLevel](#), [MakeCanonical](#), [MatchLists](#), [make\\_canonical](#)

## Examples

```
## Not run:
scnames <- c("Abrothrix longipilis", "Abrothrix jelskii")
SynList <- GetITISSyn(scnames)
BuildCSList(SynList,"Name","Syn")

## End(Not run)
```

---

cast_canonical	<i>Construct canonical names</i>
----------------	----------------------------------

---

## Description

Construct canonical names using Genus, Species and Subspecies fields. At times due to spaces or NAs in the data fields, it makes it tricky to generate canonical names.

## Usage

```
cast_canonical(dat, genus = "", species = "", subspecies = "")
```

## Arguments

dat	data frame containing taxonomic list
genus	field name for Genus field
species	field name for Species field
subspecies	field name for Subspecies field

## Value

a data frame containing Canonical names field added or repopulated using filed names for Genus, Species and Subspecies specified in parameters

## See Also

Other Name functions: [check\\_scientific](#), [expand\\_name](#), [guess\\_taxo\\_level](#), [list\\_higher\\_taxo](#), [melt\\_canonical](#)

## Examples

```
## Not run:
cast_canonical(mylist,"genus","species","subspecies")

## End(Not run)
```

---

cast_cs_field	<i>Build a character (comma) separated List within field</i>
---------------	--

---

### Description

Builds a character (comma) separated list within a field given a data frame with primary field repeating values and secondary field with values to be character separated in the same field (secondary)

### Usage

```
cast_cs_field(data, pri, sec, duplicate = FALSE, sepchar = ",")
```

### Arguments

data	data frame containing primary and secondary data columns
pri	Primary field name (repeating values)
sec	Secondary field (values would be added to same record, comma separated)
duplicate	If true, duplicate entries are allowed in secondary field
sepchar	Character separator between the data items. Default is comma

### Value

a data frame with two fields Primary and secondary (comma separated list)

### See Also

Other List functions: [melt\\_cs\\_field](#)

### Examples

```
## Not run:
scnames <- c("Abrothrix longipilis", "Abrothrix jelskii")
SynList <- list_itis_syn(scnames)
cast_cs_field(SynList,"Name","Syn")

## End(Not run)
```

---

CheckScientificName    *Parse and resolve a scientific name string*

---

### Description

This function is deprecated. Please use [check\\_scientific](#)

### Usage

```
CheckScientificName(name)
```

### Arguments

name	scientific name string to be checked
------	--------------------------------------

### Details

Parse the names using GBIF parse API to make sure the name passed is an scientific name

### Value

Resolved Canonical name and NULL is not matched

### See Also

Other Discontinued functions: [BuildCSList](#), [ExpandSyn](#), [GetWikiSyn](#), [GuessTaxoLevel](#), [MakeCanonical](#), [MatchLists](#), [make\\_canonical](#)

### Examples

```
## Not run:  
CheckScientificName("Akodon longipilis (Waterhouse, 1837)")  
CheckScientificName("Mus longipilis Waterhouse, 1837")  
CheckScientificName("Akodon hershkovitzi Patterson, Gallardo, and Freas, 1984")  
  
## End(Not run)
```

---

check_scientific	<i>Parse and resolve a scientific name string</i>
------------------	---

---

**Description**

Parse the name using GNR and GBIF parse API to make sure the name is scientific name

**Usage**

```
check_scientific(name)
```

**Arguments**

name	scientific name string to be checked
------	--------------------------------------

**Value**

Resolved canonical name (NULL if not matched)

**See Also**

Other Name functions: [cast\\_canonical](#), [expand\\_name](#), [guess\\_taxo\\_level](#), [list\\_higher\\_taxo](#), [melt\\_canonical](#)

**Examples**

```
check_scientific("Akodon longipilis (Waterhouse, 1837)")
check_scientific("Mus longipilis Waterhouse, 1837")
check_scientific("Akodon hershkovitzi Patterson, Gallardo, and Freas, 1984")
```

---

ExpandSyn	<i>Expands Scientific name if the genus is used with '.'</i>
-----------	--

---

**Description**

This function is deprecated. Please use [expand\\_name](#)

**Usage**

```
ExpandSyn(name, syn)
```

**Arguments**

name	scientific name
syn	synonym with short form genus name to expand the Genus

**Value**

Synonym with Genus expanded using either name or previous names in the syn list

**See Also**

Other Discontinued functions: [BuildCSList](#), [CheckScientificName](#), [GetWikiSyn](#), [GuessTaxoLevel](#), [MakeCanonical](#), [MatchLists](#), [make\\_canonical](#)

**Examples**

```
## Not run:
ExpandSyn("Addax gibbosa", "A. mytilopes")
ExpandSyn("Oryx addax", "O. nasomaculatus")

## End(Not run)
```

---

expand_name	<i>Expands Scientific name</i>
-------------	--------------------------------

---

**Description**

At times the genus is specified with first character and '.' rather than repeating genus names every time. These are either synonyms or species of the same genus listed one below another. To convert these names to canonical names, we need to expand the genus name (typically) using previous entry in the list.

**Usage**

```
expand_name(fullname, shortname)
```

**Arguments**

fullname	full scientific name
shortname	scientific name with short form genus name to expand the Genus

**Value**

scientific name with Genus expanded using reference name provided as parameter

**See Also**

Other Name functions: [cast\\_canonical](#), [check\\_scientific](#), [guess\\_taxo\\_level](#), [list\\_higher\\_taxo](#), [melt\\_canonical](#)

**Examples**

```
expand_name("Addax gibbosa", "A. mytilopes")
expand_name("Oryx addax", "O. nasomaculatus")
```

GetITISSyn                      *Get ITIS Synonyms for list of names*

---

**Description**

This function is deprecated. Please use [list\\_itis\\_syn](#)

**Usage**

```
GetITISSyn(namelist)
```

**Arguments**

namelist                      list of scientific names

**Details**

Fetch Synonyms from ITIS

**Value**

a data frame containing names (passed) and synonyms

**See Also**

Other ITIS functions: [ListITISSyn](#), [get\\_itis\\_syn](#), [list\\_itis\\_syn](#)

**Examples**

```
#GetITISSyn("Abrothrix longipilis")
#GetITISSyn(c("Abditomys latidens", "Abeomelomys sevia", "Abrothrix jelskii" ))
```

---

GetWikiSyn                      *Get Wikipedia Synonyms for list of names*

---

**Description**

This function is deprecated. Please use [list\\_wiki\\_syn](#)

**Usage**

```
GetWikiSyn(namelist)
```

**Arguments**

namelist                      list of scientific names



**Details**

Fetch Synonyms from Wikipedia and clean them for use

**Value**

a data frame containing names, synonyms and Canonical synonyms matched with GBIF backbone taxonomy

- Name : Scientific name
- OrigSyn : Original synonym returned by Wikipedia
- Syn : Synonym in canonical form, matched with GBIF

**See Also**

Other Discontinued functions: [BuildCSList](#), [CheckScientificName](#), [ExpandSyn](#), [GuessTaxoLevel](#), [MakeCanonical](#), [MatchLists](#), [make\\_canonical](#)

**Examples**

```
#GetWikiSyn("Abrothrix illutea")  
#GetWikiSyn(c("Abditomys latidens", "Abeomelomys sevia", "Abrocoma schistacea"))
```

---

get\_itis\_syn

*Get ITIS Synonyms for a Scientific Name*

---

**Description**

Fetch Synonyms using ITIS web service

**Usage**

```
get_itis_syn(scname)
```

**Arguments**

scname            Scientific Name

**Value**

a list containing synonyms

**See Also**

Other ITIS functions: [GetITISSyn](#), [ListITISSyn](#), [list\\_itis\\_syn](#)

**Examples**

```
## Not run:
get_itis_syn("Abrothrix longipilis")
get_itis_syn("Abditomys latidens")

## End(Not run)
```

---

GuessTaxoLevel      *Guess the level of Scientific Name*

---

**Description**

#' This function is deprecated. Please use [match\\_lists](#)

**Usage**

```
GuessTaxoLevel(name)
```

**Arguments**

name                      scientific name string to be checked

**Value**

Guess on level of Taxon name and NULL if not sure

**See Also**

Other Discontinued functions: [BuildCSList](#), [CheckScientificName](#), [ExpandSyn](#), [GetWikiSyn](#), [MakeCanonical](#), [MatchLists](#), [make\\_canonical](#)

**Examples**

```
## Not run:
GuessTaxoLevel("Akodon longipilis")
GuessTaxoLevel("Akodon")
GuessTaxoLevel("Abrocoma cinerea shistacea")

## End(Not run)
```

---

guess_taxo_level	<i>Guess the taxonomic level of Scientific Name</i>
------------------	---

---

**Description**

Guesses the taxonomic level i.e. Genus, Species or Subspecies based on number of words

**Usage**

```
guess_taxo_level(name)
```

**Arguments**

name	scientific name string to be checked
------	--------------------------------------

**See Also**

Other Name functions: [cast\\_canonical](#), [check\\_scientific](#), [expand\\_name](#), [list\\_higher\\_taxo](#), [melt\\_canonical](#)

**Examples**

```
guess_taxo_level("Akodon longipilis")
guess_taxo_level("Akodon")
guess_taxo_level("Abrocoma cinerea shistacea")
```

---

ListITISSyn	<i>Get list ITIS Synonyms for a Scientific Name</i>
-------------	---

---

**Description**

This function is deprecated. Please use [get\\_itis\\_syn](#)

**Usage**

```
ListITISSyn(scname)
```

**Arguments**

scname	Scientific Name
--------	-----------------

**Details**

Fetch Synonyms using ITIS web service

**Value**

a list containing synonyms

**See Also**

Other ITIS functions: [GetITISSyn](#), [get\\_itis\\_syn](#), [list\\_itis\\_syn](#)

**Examples**

```
#ListITISSyn("Abrothrix longipilis")
#ListITISSyn("Abditomys latidens")
```

---

list_higher_taxo	<i>Get higher taxonomy data for list of names</i>
------------------	---

---

**Description**

Retrieve higher taxonomy information (like Family and Order) for each record from the "Encyclopedia of Life" web API.

**Usage**

```
list_higher_taxo(indf, canonical, genus = FALSE, verbose = FALSE,
  progress = TRUE)
```

**Arguments**

indf	input data frame containing taxonomic list
canonical	field name containing scientific names
genus	If TRUE, use only genus level data to get taxonomy
verbose	If TRUE, displays each name string for which the higher taxonomy is sought
progress	If TRUE prints progress bar and messages on the console.

**Details**

This function makes use of certain functions in the [taxize](#) package. It scans and retrieves the taxonomic hierarchy for each scientific name (or just genus name) in the data set. When new data are retrieved, they are stored in a local sqlite database, `taxo.db`, for faster further access.

**Value**

data frame with added / updated columns

- "Kingdom" Kingdom of the Scientific name
- "Phylum" Phylum of the Scientific name
- "Order\_" Order of the Scientific name
- "Family" Family of the Scientific name
- "Genus" Genus of the Scientific name

and also saves a local copy of taxonomy downloaded for future use in 'taxo.db' sqlite file

**See Also**

Other Name functions: [cast\\_canonical](#), [check\\_scientific](#), [expand\\_name](#), [guess\\_taxo\\_level](#), [melt\\_canonical](#)

**Examples**

```
## Not run:  
inat <- list_higher_taxo(inat)  
  
## End(Not run)
```

---

list_itis_syn	<i>Get ITIS Synonyms for list of names</i>
---------------	--

---

**Description**

Fetch Synonyms from ITIS

**Usage**

```
list_itis_syn(namelist)
```

**Arguments**

namelist      list of scientific names

**Value**

a data frame containing names (passed) and synonyms

**See Also**

Other ITIS functions: [GetITISyn](#), [ListITISyn](#), [get\\_itis\\_syn](#)

**Examples**

```
## Not run:
list_itis_syn("Abrothrix longipilis")
#list_itis_syn(c("Abditomys latidens", "Abeomelomys sevia", "Abrothrix jelskii" ))

## End(Not run)
```

---

`list_wiki_syn`*Get Wikipedia Synonyms for list of names*

---

**Description**

Fetch Synonyms from Wikipedia and clean them for use

**Usage**

```
list_wiki_syn(namelist)
```

**Arguments**

`namelist`      list of scientific names

**Value**

a data frame containing names, synonyms and Canonical synonyms matched with GBIF backbone taxonomy

- Name : Scientific name
- OrigSyn : Original synonym returned by Wikipedia
- Syn : Synonym in canonical form, matched with GBIF

**Examples**

```
list_wiki_syn("Abrothrix illutea")
#list_wiki_syn(c("Abditomys latidens", "Abeomelomys sevia", "Abrocoma schistacea"))
```

---

MakeCanonical	<i>Construct canonical names</i>
---------------	----------------------------------

---

## Description

This function is depreciated. Please use [make\\_canonical](#)

## Usage

```
MakeCanonical(dat, genus = "", species = "", subspecies = "")
```

## Arguments

dat	data frame containing taxonomic list
genus	field name for Genus field
species	field name for Species field
subspecies	field name for Subspecies field

## Details

Canonical names using Genus, Species and Subspecies friends

## Value

a data frame containing Canonical names field added or repopulated using filed names for Genus, Species and Subspecies specified in parameters

## See Also

Other Discontinued functions: [BuildCSList](#), [CheckScientificName](#), [ExpandSyn](#), [GetWikiSyn](#), [GuessTaxoLevel](#), [MatchLists](#), [make\\_canonical](#)

## Examples

```
## Not run:  
MakeCanonical(mylist, "genus", "species", "subspecies")  
  
## End(Not run)
```

---

make_canonical	<i>Construct canonical names</i>
----------------	----------------------------------

---

### Description

Construct canonical names using Genus, Species and Subspecies fields. At times due to spaces or NAs in the data fields, it makes it tricky to generate canonical names.

### Usage

```
make_canonical(dat, genus = "", species = "", subspecies = "")
```

### Arguments

dat	data frame containing taxonomic list
genus	field name for Genus field
species	field name for Species field
subspecies	field name for Subspecies field

### Value

a data frame containing Canonical names field added or repopulated using filed names for Genus, Species and Subspecies specified in parameters

### See Also

Other Discontinued functions: [BuildCSList](#), [CheckScientificName](#), [ExpandSyn](#), [GetWikiSyn](#), [GuessTaxoLevel](#), [MakeCanonical](#), [MatchLists](#)

### Examples

```
## Not run:  
make_canonical(mylist,"genus","species","subspecies")  
  
## End(Not run)
```



---

MatchLists	<i>match two taxonomic lists</i>
------------	----------------------------------

---

### Description

This function is deprecated. Please use [match\\_lists](#)

### Usage

```
MatchLists(master, lst, masterfld, lstfld)
```

### Arguments

master	master taxonomic list
lst	match taxonomic list
masterfld	field name for canonical name in master list
lstfld	field name for canonical name in match list

### Details

match two taxonomic lists using canonical names

### Value

a list with two data frames containing matched and non-matched names from the master list

### See Also

Other Discontinued functions: [BuildCSList](#), [CheckScientificName](#), [ExpandSyn](#), [GetWikiSyn](#), [GuessTaxoLevel](#), [MakeCanonical](#), [make\\_canonical](#)

### Examples

```
## Not run:  
MatchLists(master, lst, "canonical", "canonical")  
  
## End(Not run)
```

---

match_lists	<i>match two taxonomic lists</i>
-------------	----------------------------------

---

**Description**

match two taxonomic lists using canonical names

**Usage**

```
match_lists(master, checklist, masterfld, checklistfld)
```

**Arguments**

master	master taxonomic list
checklist	match taxonomic list
masterfld	field name for canonical name in master list
checklistfld	field name for canonical name in match list

**Value**

a list with two data frames containing matched and non-matched names from the master list

**Examples**

```
## Not run:  
match_lists(master, checklist, "canonical", "canonical")  
  
## End(Not run)
```

---

melt_canonical	<i>Deconstruct canonical names</i>
----------------	------------------------------------

---

**Description**

Deconstruct canonical names into Genus, Species and Subspecies fields.

**Usage**

```
melt_canonical(dat, canonical = "", genus = "", species = "",  
subspecies = "")
```

**Arguments**

dat	data frame containing taxonomic list
canonical	field name for canonical names
genus	field name for Genus
species	field name for Species
subspecies	field name for Subspecies

**Value**

a data frame containing Genus, Species and Subspecies fields added or or repopulated using data in canonical name field.

**See Also**

Other Name functions: [cast\\_canonical](#), [check\\_scientific](#), [expand\\_name](#), [guess\\_taxo\\_level](#), [list\\_higher\\_taxo](#)

**Examples**

```
## Not run:
melt_canonical(mylist,"genus","species","subspecies")

## End(Not run)
```

---

melt_cs_field	<i>Build a list separating character (comma) separated field into separate records</i>
---------------	--

---

**Description**

Builds a list separating character (comma) separated field given a data frame with a field with repeating values

**Usage**

```
melt_cs_field(data, pri, sepchar = ",")
```

**Arguments**

data	data frame containing a data columns with character(comma) separated values
pri	Primary field name with character(comma) separated values
sepchar	Character separator between the data items. Default is comma

**Value**

a data frame with separate records for each value in primary field specified

**See Also**

Other List functions: [cast\\_cs\\_field](#)

**Examples**

```
## Not run:
scnames <- c("Abrothrix longipilis", "Abrothrix jelskii")
syn_list <- list_itis_syn(scnames)
cs_syn_list <- cast_cs_field(syn_list , "Name", "Syn")
syn_list_new <- melt_cs_field(cs_syn_list, "Syn")

## End(Not run)
```

---

taxotools

*taxotools: Tools to handle taxonomic data*

---

**Description**

Some tools to work with lists taxonomic names.

**List functions**

- [cast\\_cs\\_field](#)
- [melt\\_cs\\_field](#)
- [match\\_lists](#)

**Name functions**

- [expand\\_name](#)
- [check\\_scientific](#)
- [guess\\_taxo\\_level](#)
- [cast\\_canonical](#)
- [melt\\_canonical](#)
- [list\\_higher\\_taxo](#)

**ITIS functions**

- [get\\_itis\\_syn](#) #'
- [list\\_itis\\_syn](#)

**Wiki functions**

- [list\\_wiki\\_syn](#)

**Discontinued functions**

- [MatchLists](#)
- [BuildCSList](#)
- [ExpandSyn](#)
- [CheckScientificName](#)
- [GuessTaxoLevel](#)
- [MakeCanonical](#)
- [GetWikiSyn](#)
- [GetITISyn](#)
- [ListITISyn](#)
- [make\\_canonical](#)

**Citation**

- Barve, V., (2019). taxotools: Tools to handle taxonomic data (R package V 0.0.26). Retrieved from <https://cran.r-project.org/web/packages/taxotools/index.html>

# Index

[BuildCSList](#), [2](#), [5](#), [7](#), [9](#), [10](#), [15–17](#), [21](#)

[cast\\_canonical](#), [3](#), [6](#), [7](#), [11](#), [13](#), [19](#), [20](#)  
[cast\\_cs\\_field](#), [2](#), [4](#), [20](#)  
[check\\_scientific](#), [3](#), [5](#), [6](#), [7](#), [11](#), [13](#), [19](#), [20](#)  
[CheckScientificName](#), [2](#), [5](#), [7](#), [9](#), [10](#), [15–17](#),  
[21](#)

[expand\\_name](#), [3](#), [6](#), [7](#), [11](#), [13](#), [19](#), [20](#)  
[ExpandSyn](#), [2](#), [5](#), [6](#), [9](#), [10](#), [15–17](#), [21](#)

[get\\_itis\\_syn](#), [8](#), [9](#), [11–13](#), [20](#)  
[GetITISSyn](#), [8](#), [9](#), [12](#), [13](#), [21](#)  
[GetWikiSyn](#), [2](#), [5](#), [7](#), [8](#), [10](#), [15–17](#), [21](#)  
[guess\\_taxo\\_level](#), [3](#), [6](#), [7](#), [11](#), [13](#), [19](#), [20](#)  
[GuessTaxoLevel](#), [2](#), [5](#), [7](#), [9](#), [10](#), [15–17](#), [21](#)

[list\\_higher\\_taxo](#), [3](#), [6](#), [7](#), [11](#), [12](#), [19](#), [20](#)  
[list\\_itis\\_syn](#), [8](#), [9](#), [12](#), [13](#), [20](#)  
[list\\_wiki\\_syn](#), [8](#), [14](#), [20](#)  
[ListITISSyn](#), [8](#), [9](#), [11](#), [13](#), [21](#)

[make\\_canonical](#), [2](#), [5](#), [7](#), [9](#), [10](#), [15](#), [16](#), [17](#), [21](#)  
[MakeCanonical](#), [2](#), [5](#), [7](#), [9](#), [10](#), [15](#), [16](#), [17](#), [21](#)  
[match\\_lists](#), [10](#), [17](#), [18](#), [20](#)  
[MatchLists](#), [2](#), [5](#), [7](#), [9](#), [10](#), [15](#), [16](#), [17](#), [21](#)  
[melt\\_canonical](#), [3](#), [6](#), [7](#), [11](#), [13](#), [18](#), [20](#)  
[melt\\_cs\\_field](#), [4](#), [19](#), [20](#)

[taxize](#), [12](#)  
[taxotools](#), [20](#)  
[taxotools-package \(taxotools\)](#), [20](#)