

# taxize - taxonomic search and retrieval in R

Scott Chamberlain<sup>1,\*</sup> and Eduard Szöcs<sup>2,†</sup>

<sup>1</sup>*Biology Department, Simon Fraser University, Canada.*

<sup>2</sup>*University Koblenz-Landau, Germany*

Keywords: taxonomy; R; software; data; API

---

\* E-mail: [myrmecocystus@gmail.com](mailto:myrmecocystus@gmail.com)

† E-mail: [szoe8822@uni-landau.de](mailto:szoe8822@uni-landau.de)

## I. INTRODUCTION

some text here

## II. THE CASE FOR TAXIZE

There are a large suite of applications developed around the problem of searching for, resolving, and getting higher taxonomy for species names. For example, Linnaeus <http://linnaeus.sourceforge.net/> provides ability to search for taxonomic names in documents and normalize names. In addition, there are many web interfaces to search for and normalize names such as Encyclopedia of Life's Global Names Resolver <http://resolver.globalnames.org/>, uBio tools [http://www.ubio.org/index.php?pagename=sample\\_tools](http://www.ubio.org/index.php?pagename=sample_tools), and iPlant's Taxonomic Name Resolution Service <http://tnrs.iplantcollaborative.org/>.

All of these tools provide great ways to search for taxonomic names and resolve them in some cases. However, scientists ideally need a tool that can be used programmatically, and thus be made reproducible, and highly customizable. The goal of taxize is to make it easy to create reproducible and easy to use workflows for searching for taxonomic names, resolving them, getting higher taxonomic names, and other tasks related to research dealing with species.

## III. DATA SOURCES

taxize uses many data sources, and more can easily be added.

TABLE I. Data sources used in taxize

Source name	Name search	Name resolution	Phylogeny	URL
Encyclopedia of Life	Yes	See GNR below	No	<a href="http://eol.org/">http://eol.org/</a>
Integrated Taxonomic Resolution Service	Yes	Synonyms	No	X
iPlant Taxonomic Name Resolution Service	Yes	Yes	No	X
Phylomatic	No	X	No	X
uBio	Yes	X	No	X
Global Names Resolver	Yes	X	No	X
Global Names Index	Yes	X	No	X
IUCN Red List	Yes	X	No	X
Tropicos	Yes	X	No	X
Plantminer	Yes	X	No	X
Theplantlist.org	Yes	X	No	X
Catalogue of Life	Yes	X	No	X
National Center for Biotechnology Information (NCBI)	Yes	X	No	X

## IV. USE CASES

There are a variety of use cases for which taxize is ideally suited, and few side cases in which taxize can be useful. We discuss five ideal use cases for taxize at length, and highlight the side cases in brief.

### A. Installing taxize

First, let's install taxize. There are two versions of taxize, a stable release that can be installed from the R package repository, CRAN, and from GitHub, where the code is developed.

Installing from CRAN or GitHub

```
## From CRAN
install.packages("taxize")

## From GitHub
```

```
install_github("taxize_", "ropensci")
```

Loading into your R session

```
library(taxize)
```

## B. Resolve taxonomic names

This is a common task in biology. We often have a list of species names and we want to know if a) we have the most up to date names, b) our names are spelled correctly, and c) if we have common names, we likely need the scientific names. One way to resolve names is via the Global Names Resolver service provided by the Encyclopedia of Life (<http://resolver.globalnames.org/>).

```
# Here, we are searching for two misspelled names
temp <- gnr_resolve(names = c("Helianthus annus", "Homo saapiens"), returndf = TRUE)

# let's take a peek at the data, excluding the data source ID and score
# columns
temp[, -c(1, 4)]
```

	submitted_name	name_string	title
1	Helianthus annus	Helianthus annuus L.	Catalogue of Life
3	Helianthus annus	Helianthus annus GBIF	Taxonomic Backbone
4	Helianthus annus	Helianthus annus	EOL
5	Helianthus annus	Helianthus annus L.	EOL
6	Helianthus annus	Helianthus annus	uBio NameBank
2	Homo saapiens	Homo sapiens Linnaeus, 1758	Catalogue of Life

Looks like the correct spellings are *Helianthus annuus* and *Homo sapiens*, cool!

Another approach is using the Taxonomic Name Resolution Service via the Taxosaurus API (<http://taxosaurus.org/>).

```
# Lets set our list of species names
mynames <- c("Helianthus annuus", "Pinus contort", "Poa anua", "Abis magnifica",
            "Rosa californica", "Festuca arundinacea", "Sorbus occidentalis", "Madia sateva")

# And we'll call the API with the tnrs function
tnrs(query = mynames)[, -c(5:7)]
```

	submittedName	acceptedName	sourceId	score
7	Helianthus annuus	Helianthus annuus	iPlant_TNRS	1.00
4	Pinus contort	Pinus contorta	iPlant_TNRS	0.98
5	Poa anua	Poa alta	iPlant_TNRS	0.77
3	Abis magnifica	Abies magnifica	iPlant_TNRS	0.96
8	Rosa californica	Rosa californica	iPlant_TNRS	0.99
2	Festuca arundinacea	Festuca arundinacea	iPlant_TNRS	0.99
1	Sorbus occidentalis	Sorbus occidentalis	iPlant_TNRS	0.99
6	Madia sateva	Madia sativa	iPlant_TNRS	0.97

It looks like there are a few corrections: e.g., *Madia sateva* should be *Madia sativa*, and *Rosa californica* should be *Rosa californica*.

## C. Retrieve higher taxonomic names

Another task biologists often face is wanting to get higher taxonomic names for their list of taxa. If you have the higher taxonomy you can put in to context the relationships of your list (i.e., Species A and B are in Family X), as

opposed to not knowing that Species A and B are closely related. A number of data sources provide this type of capability. First, let's take a look at the Integrated Taxonomic Information Service (ITIS).

```
specieslist <- c("Abies procera", "Pinus contorta")
classification(get_tsn(specieslist, "sciname"))
```

Retrieving data for species ' Abies procera '

Retrieving data for species ' Pinus contorta '

```
[[1]]
```

	parentName	parentTsn	rankName	taxonName	tsn
1			Kingdom	Plantae	202422
2	Plantae	202422	Subkingdom	Viridaplantae	846492
3	Viridaplantae	846492	Infrakingdom	Streptophyta	846494
4	Streptophyta	846494	Division	Tracheophyta	846496
5	Tracheophyta	846496	Subdivision	Spermatophytina	846504
6	Spermatophytina	846504	Infradivision	Gymnospermae	846506
7	Gymnospermae	846506	Class	Pinopsida	500009
8	Pinopsida	500009	Order	Pinales	500028
9	Pinales	500028	Family	Pinaceae	18030
10	Pinaceae	18030	Genus	Abies	18031
11	Abies	18031	Species	Abies procera	181835

```
[[2]]
```

	parentName	parentTsn	rankName	taxonName	tsn
1			Kingdom	Plantae	202422
2	Plantae	202422	Subkingdom	Viridaplantae	846492
3	Viridaplantae	846492	Infrakingdom	Streptophyta	846494
4	Streptophyta	846494	Division	Tracheophyta	846496
5	Tracheophyta	846496	Subdivision	Spermatophytina	846504
6	Spermatophytina	846504	Infradivision	Gymnospermae	846506
7	Gymnospermae	846506	Class	Pinopsida	500009
8	Pinopsida	500009	Order	Pinales	500028
9	Pinales	500028	Family	Pinaceae	18030
10	Pinaceae	18030	Genus	Pinus	18035
11	Pinus	18035	Species	Pinus contorta	183327

It turns out both species are in the family Pinaceae. You can also get this type of information from the NCBI by doing `classification(get_uid(specieslist))`

Instead of a full classification, you may only want a single name, say a family name for your species of interest. The function `tax_name` is built just for this purpose. And you can specify the data source you retrieve the taxonomic name from with the `db` parameter.

```
tax_name(query = "Helianthus annuus", get = "family", db = "itis")
```

Retrieving data for species ' Helianthus annuus '

```
family
1 Asteraceae
```

```
tax_name(query = "Helianthus annuus", get = "family", db = "ncbi")
```

Retrieving data for species ' Helianthus annuus '

```
family
1 Asteraceae
```

### D. Retrieve a phylogeny

There is an increasingly common use case: many biologists are not adequately trained in reconstructing phylogenies. However, so-called *taxonomic phylogenies* can be constructed from simply knowing the higher taxonomic classification of a set of taxa. There are few taxon groups for which we can get phylogenies simply based on taxonomy; one of these is for angiosperms, called Phylomatic [1]. We have created a workflow in taxize that lets you input a simple species list, and then taxize does work behind the scenes to get higher taxonomic names, which are required by Phylomatic to get a phylogeny. Here is a short example.

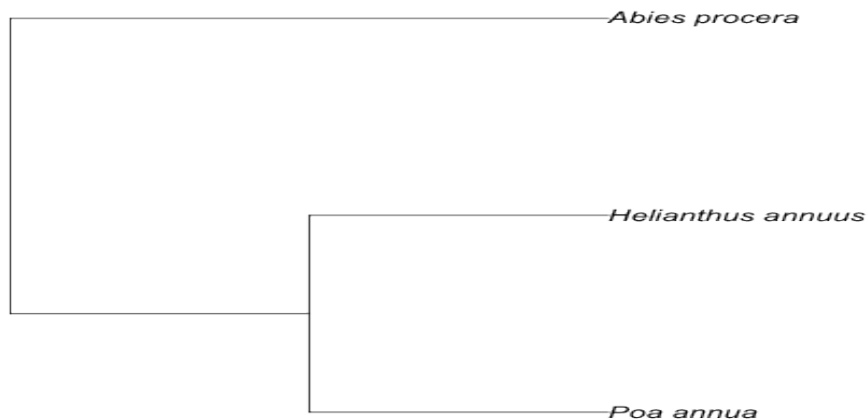
```
library(doMC)
registerDoMC(cores = 4)

# input the taxonomic names
taxa <- c("Poa annua", "Abies procera", "Helianthus annuus")

# fetch the tree - the formatting of names and higher taxonomy is done
# within the function
tree <- phylomatic_tree(taxa = taxa, get = "POST", informat = "newick", method = "phylomatic",
  storedtree = "R20120829", taxaformat = "slashpath", outformat = "newick",
  clean = "true")

tree$tip.label <- capwords(tree$tip.label)

# plot the tree
plot(tree, cex = 1.2)
```



### E. What taxa are in children of my taxon of interest?

If you aren't a taxonomic specialist on a particular taxon you likely don't know what children taxa are within a family, or within a genus. You can of course go to a website like Wikispecies ([http://species.wikimedia.org/wiki/Main\\_Page](http://species.wikimedia.org/wiki/Main_Page)) or Encyclopedia of Life (<http://eol.org/>). taxize provides an easy way for you to search for downstream taxa, both for the Catalogue of Life (CoL; <http://www.catalogueoflife.org/>) and the Integrated Taxonomic Information Database (<http://www.itis.gov/>). Here is a short example using the CoL in which we want to find all the species within the genus *Apis* (honey bees).

```
col_downstream(name = "Apis", downto = "Species")

Apis
  childtaxa_id  childtaxa_name childtaxa_rank
1      6971712 Apis andreniformis      Species
2      6971713      Apis cerana      Species
3      6971714      Apis dorsata      Species
4      6971715      Apis florea      Species
5      6971716 Apis koschevnikovi      Species
6      6845885      Apis mellifera      Species
7      6971717      Apis nigrocincta      Species
```

F. IUCN Status

There are a number of things we can do once we have the correct taxonomic names. One thing we can do is ask about the conservation status of a species. We have provided a set of functions, *iucn\_summary* and *iucn\_status*, to search for species names, and extract the status information, respectively. Here, we search for the Panther and Lynx.

```
ia <- iucn_summary(c("Panthera uncia", "Lynx lynx"))
iucn_status(ia)

Panthera uncia      Lynx lynx
      "EN"          "LC"
```

It turns out that the Panther is endangered (EN) and the Lynx is of least concern (LN).

V. CONCLUSION

some text here

VI. FUNDING

SAC is supported by CANPOLIN of Canada. EZ is supported by XXXX.

VII. ACKNOWLEDGEMENTS

some text here

- 
- [1] Campbell O Webb and Michael J Donoghue, “Phylomatic: tree assembly for applied phylogenetics,” *Molecular Ecology Notes* **5**, 181–183 (2005).