

# 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>*Institute for Environmental Sciences, University Koblenz-Landau, Forststraße 7, 76829 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. ABSTRACT

All species are hierarchically related to one another, and we use taxonomic names to label the nodes in this hierarchy. Taxonomic data is becoming easily available on the web, but scientists need a way to access taxonomic data on the web in a programmatic fashion, which is fast and reproducible. We have developed *taxize*, an open-source software package (freely available from <http://cran.r-project.org/web/packages/taxize/index.html>) for the R language. *taxize* provides a simple, programmatic access to taxonomic data for 13 data sources around the web. We discuss the need for a taxonomic toolbelt in R, and outline a suite of use cases for which *taxize* is ideally suited (including a full workflow as an appendix). The *taxize* package will facilitate open and reproducible science by allowing taxonomic data collection to be done in the open-source R platform.

## II. INTRODUCTION

Evolution by natural selection has led to a hierarchical relationship among all living organisms. Thus, species are categorized using a taxonomic hierarchy, starting with the binomial species name (e.g., *Homo sapiens*), moving up to genus (*Homo*), then family (*Hominidae*), and on up to Domain (*Eukarya*). Biologists, whether studying organisms at the cell, organismal, or community level, can put their study taxa into taxonomic context, allowing them to know close and distant relatives, find relevant literature, and more. Discovering the correct taxonomic names is, unfortunately, not straightforward. Taxonomic names often change due to name changes at the generic or specific levels, lumping or splitting lower taxa (genera, species) among higher taxa (families), and name spelling changes. In addition, there is no one authoritative taxonomic names source. Instead, there are essentially competing sources (e.g., uBio, Tropicos, ITIS) that may have different accepted names for the same taxon. The goal of *taxize*, an R package in development, is to make all use cases having to do with taxonomy easy and replicable.

Taxonomic data is getting easier to obtain through web interfaces (e.g., <http://eol.org/>). However, there are a number of good reasons to obtain taxonomic information programatically rather than through a web interface. First, if you have more than a few names to lookup on a website, it can take quite a long time to enter each name, get data, and repeat for each species. Second, programatically getting taxonomic names solves the first problem by looping over a list of names, and is reproducible. With increasing reports of irreproducibility in science [1, 2], it is extremely important to make science workflows repeatable.

The R language is the dominant language used by biologists (reference), and now has over 4,000 packages on the main repository (CRAN) and more than 2,500 packages on other repositories to extend R. R is great for manipulating, visualizing and fitting statistical models to data. However, the key missing piece in R is the ability to get data within R. Increasingly, data is available from the web via API's, or application programming interfaces. These are bits of code that allow computers to talk to one another. Web APIs define a number of methods that allow users to search for a species name, or retrieve the synonyms for a species name, for example. One can then write functions in any programming language to interact with the API, allowing users to interact with the web API without having to know the details of the code. In *taxize*, we have written a suite of R functions that interact with many taxonomic data sources via their web APIs (Table I). The interface to each function is usually a simple list of species names, just as user would do with a web API.

## III. WHY 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.

## IV. DATA SOURCES

taxize uses many data sources (Table I), and more can easily be added. There are two common tasks provided by the data sources: name search and name resolution. Other functionality in taxize includes retrieving a classification tree for a species, or retrieving child taxa of a focal taxon. One of the data sources (Phylomat) returns phylogenies, while another (NCBI) returns genetic sequence data. However, there are other R packages that are focused solely on sequence data, such as rsnp [3], rentrez [4], BoSSA [5], and ape [6].

Some of the data sources taxize interacts with require authentication. That is, in addition to the search terms you provide (e.g., *Homo sapiens*), they require an alphanumeric identification key so that they can better manage their data. The services that do require an API key are: Encyclopedia of Life (EOL), the Universal Biological Indexer and Organizer (uBio), Tropicos, and Plantminer. You can easily obtain an API key by visiting the website of each service (see (Table I) for links to each site).

TABLE I. Data sources used in taxize, tasks available, and links to them

Source name	Name search	Name resolution	Phylogeny	Sequences	URL
Encyclopedia of Life	Yes	See GNR below	No	No	<a href="http://eol.org/">http://eol.org/</a>
Integrated Taxonomic Information System	Yes	Synonyms	No	No	<a href="http://www.itis.gov/">http://www.itis.gov/</a>
iPlant Taxonomic Name Resolution Service	Yes	Yes	No	No	<a href="http://bit.ly/16dHkBy">http://bit.ly/16dHkBy</a>
Phylomatic	No	No	Yes	No	<a href="http://bit.ly/P0pjMz">http://bit.ly/P0pjMz</a>
uBio	Yes	Yes	No	No	<a href="http://www.ubio.org/">http://www.ubio.org/</a>
Global Names Resolver	Yes	Yes	No	No	<a href="http://bit.ly/11R3Pbr">http://bit.ly/11R3Pbr</a>
Global Names Index	Yes	No	No	No	<a href="http://bit.ly/11R3RQB">http://bit.ly/11R3RQB</a>
IUCN Red List	Yes	No	No	No	<a href="http://bit.ly/11R3RQC">http://bit.ly/11R3RQC</a>
Tropicos	Yes	Yes	No	No	<a href="http://www.tropicos.org/">http://www.tropicos.org/</a>
Plantminer	Yes	No	No	No	<a href="http://www.plantminer.com/">http://www.plantminer.com/</a>
The Plant List	Yes	Yes	No	No	<a href="http://www.theplantlist.org/">http://www.theplantlist.org/</a>
Catalogue of Life	Yes	Yes	No	No	<a href="http://bit.ly/11R3S75">http://bit.ly/11R3S75</a>
National Center for Biotechnology Information	Yes	X	Yes <sup>a</sup>	Yes	<a href="http://www.ncbi.nlm.nih.gov/">http://www.ncbi.nlm.nih.gov/</a>

<sup>a</sup> Web only, no API, see <http://1.usa.gov/11R446a>

## V. 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.packages("devtools")
require(devtools)
install_github("taxize_", "ropensci")
```

Loading taxize 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 (GNR) service provided by the Encyclopedia of Life (<http://resolver.globalnames.org/>).

```
# Here, we are searching for two misspelled names
temp <- gnr_resolve(names = c("Helianthos 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)]
```

It looks like the correct spellings are *Helianthus annuus* and *Homo sapiens*. Another approach uses the Taxonomic Name Resolution Service via the Taxosaurus API (<http://taxosaurus.org/>).

```
# A list of species names, some of which are misspelled
mynames <- c("Helianthus annuus", "Pinus contort", "Poa anua", "Abis magnifica",
             "Rosa california", "Festuca arundinace", "Sorbus occidentalos", "Madia sateva")

# And we'll call the API with the tnrs function, and remove a few columns
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 annua	iPlant_TNRS	0.96
3	Abis magnifica	Abies magnifica	iPlant_TNRS	0.96
8	Rosa california	Rosa californica	iPlant_TNRS	0.99
2	Festuca arundinace	Festuca arundinacea	iPlant_TNRS	0.99
1	Sorbus occidentalos	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 california* should be *Rosa californica*.

taxize takes the approach that the user should be able to make decisions about what resource to trust, rather than taxize making the decision. Both the EOL GNR and the TNRS services provide data from a variety of data sources. The user may trust a specific data source, thus may want to use the names from that data source.

## C. Retrieve higher taxonomic names

Another task biologists often face is getting higher taxonomic names for a taxa list. Having the higher taxonomy allows you to put into context the relationships of your species list (i.e., Species A and B are in Family C), as opposed to not knowing that Species A and B are closely related. This also makes it easy to aggregate/standardize data to a specific taxonomic level (e.g., family level) or to match data to other databases with different taxonomic resolution (e.g., trait data).

Two data sources provide this type of capability: ITIS and NCBI. The principle in both is the same - first you need to get an identifier for the queried species and then can retrieve additional information with this ID.

```
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	Viridaeplantae	846492
3	Viridaeplantae	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	Viridaeplantae	846492
3	Viridaeplantae	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. Interactive name selection

In the previous section we used the function `get_tsn_to_get_a_classification_for_two_plant_species`.

```
# In this example, this search term has many matches. The function returns
# a data.frame of the matches, and asks for user input. Enter the row
# number of the name you want.
get_tsn(searchterm = "Elisabethae", searchtype = "sciname")
```

Retrieving data for species ' Elisabethae '

	combinedname	tsn
1	Bittacus elisabethae	810495
2	Caenauchochloa elisabethae	754498
3	Ceratina elisabethae	765455
4	Dolicheremaeus elisabethae	740429
5	Euphione elisabethae	64893
6	Harmothoe elisabethae	573734
7	Hydraena elisabethae	815669
8	Lizzia elisabethae	48875
9	Melinna elisabethae	67765
10	Myrmicocrypta elisabethae	580527
11	Myzomela boiei elisabethae	713955
12	Notosacantha elisabethae	838430
13	Odontomyia elisabethae	627043
14	Parabelbella elisabethae	737749
15	Peltigera elisabethae	191230
16	Polymorphanisus elisabethae	600292
17	Pseudopterogorgia elisabethae	52264
18	Rhantus elisabethae	815107
19	Scyllarides elisabethae	553004
20	Sellnickochthonius elisabethae	735137
21	Serilophus lunatus elisabethae	711636
22	Stenhelina elisabethae	87927
23	Tetramorium elisabethae	582397
24	Theridion elisabethae	868794
25	Trichoniscus elisabethae	597365
26	Trichoniscus elisabethae estoniensis	598124

More than one TSN found for species ' Elisabethae '!

Enter rownumber of species (other inputs will return 'NA'):

```
# You can pass in a long list of taxonomic names
splist <- c("annona cherimola", "annona muricata", "quercus robur", "shorea robusta",
            "pandanus patina", "oryza sativa", "durio zibethinus")
get_tsn(searchterm = splist, searchtype = "sciname")
```

Retrieving data for species ' annona cherimola '

Retrieving data for species ' annona muricata '

Retrieving data for species ' quercus robur '

Retrieving data for species ' shorea robusta '

Retrieving data for species ' pandanus patina '

Retrieving data for species ' oryza sativa '

```
Retrieving data for species ' durio zibethinus '
[1] "506198" "18098" "19405" "506787" "507376" "41976" "506099"
attr(,"class")
[1] "tsn"
```

```
# In this example, note that no match at all returns an NA
get_uid(sciname = c("Chironomus riparius", "aaa vva"))
```

```
Retrieving data for species ' Chironomus riparius '
```

```
Retrieving data for species ' aaa vva '
[1] "315576" NA
attr(,"class")
[1] "uid"
```

```
text <- "9,562445749\n37,66119516\n72,9103195\n"
scan(dec = ",", text = text)

[1] 9.562 37.661 72.910
```

### E. Retrieve a phylogeny

Many biologists are not adequately trained in reconstructing phylogenies. However, there are some sources for getting a phylogeny without having to now how to build one; one of these is for angiosperms, called *Phyloomatic* [7]. We have created a workflow in *taxize* that lets you input a simple species list, and then *taxize* works behind the scenes to get higher taxonomic names, which are required by *Phyloomatic* to get a phylogeny. Here is a short example.

```
# 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")
```

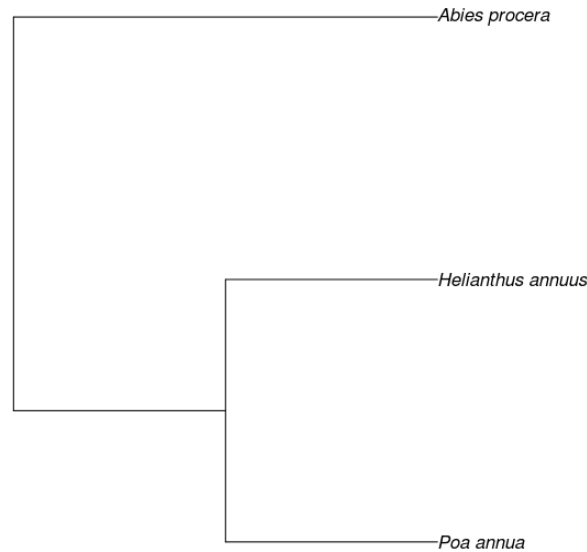
```
Retrieving data for species ' Poa annua '
```

```
Retrieving data for species ' Abies procera '
```

```
Retrieving data for species ' Helianthus annuus '
```

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

```
# plot the tree
plot(tree, cex = 1)
```



Behind the scenes the function `phylomatic_tree` retrieves a Taxonomic Serial Number (TSN) from ITIS for each species name, then a string is created for each species like this `poaceae/oryza/oryza_sativa` (with format 'family/genus/genus\_epithet').

#### F. What taxa are the 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")[[1]]
```

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

#### G. 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).

## VI. CONCLUSION

some text here

## VII. FUNDING

SAC is supported by CANPOLIN of Canada.

## VIII. ACKNOWLEDGEMENTS

The taxize package is part of the rOpenSci project <http://ropensci.org/>.

## IX. APPENDICES

- Appendix A. A complete reproducible workflow, from a species list to a phylogeny.

- 
- [1] Victoria C Stodden, “Reproducible research: Addressing the need for data and code sharing in computational science,” *Computing in Science & Engineering* **12**, 8–12 (2010).
  - [2] Carl Zimmer, “A sharp rise in retractions prompts calls for reform,” *New York Times* (2012).
  - [3] Scott Chamberlain and Kevin Ushey, *rsnps: Interface to SNP data on the web*. (2013), r package version 0.0.4.
  - [4] David Winter, *rentrez: Entrez in R* (2013), r package version 0.2.1.
  - [5] Pierre Lefeuvre, *BoSSA: a Bunch of Structure and Sequence Analysis* (2010), r package version 1.2.
  - [6] E. Paradis, J. Claude, and K. Strimmer, “APE: analyses of phylogenetics and evolution in R language,” *Bioinformatics* **20**, 289–290 (2004).
  - [7] Campbell O Webb and Michael J Donoghue, “Phyloomatic: tree assembly for applied phylogenetics,” *Molecular Ecology Notes* **5**, 181–183 (2005).