# taxize vignette - a taxonomic toolbelt for R

## About the package

`taxize` is a taxonomic toolbelt for R. `taxize` wraps APIs for a large suite of taxonomic databases availab on the web.

---

## Quick start

**First, install `taxize`** First, install and load `taxize` into the R session.

```
install.packages("taxize")
```

```
library(taxize)
```

```
##
##
## New to taxize? Tutorial at http://ropensci.org/tutorials/taxize_tutorial.html
## citation(package='taxize') for the citation for this package
## API key names have changed. Use tropicosApiKey, eolApiKey, ubioApiKey, and pmApiKey in your .Rprofil
## Use suppressPackageStartupMessages() to suppress these startup messages in the future
```

Advanced users can also download and install the latest development copy from GitHub.

**Resolve taxonomic name** This is a common task in biology. We often have a list of species names and we want to know a) if we have the most up to date names, b) if our names are spelled correctly, and c) the scientific name for a common name. One way to resolve names is via the Global Names Resolver (GNR) service provided by the Encyclopedia of Life. Here, we are searching for two misspelled names:

```
temp <- gnr_resolve(names = c("Helianthos annus", "Homo saapiens"))
```

```
## Loading required package: rjson
```

```
temp[, -c(1, 4)]
```

```
##                  matched_name data_source_title
## 1        Helianthus annuus L. Catalogue of Life
## 2         Helianthus annus L.               EOL
## 3            Helianthus annus               EOL
## 4            Helianthus annus     uBio NameBank
## 5 Homo sapiens Linnaeus, 1758 Catalogue of Life
```

The correct spellings are *Helianthus annuus* and *Homo sapiens*. Another approach uses the Taxonomic Name Resolution Service via the Taxosaurus API developed by iPLant and the Phylotastic organization. In this example, we provide a list of species names, some of which are misspelled, and we'll call the API with the *tnrs* function.

```
mynames <- c("Helianthus annuus", "Pinus contort", "Poa anua", "Abis magnifica",
    "Rosa california", "Festuca arundinace", "Sorbus occidentalos", "Madia sateva")
tnrs(query = mynames, source = "iPlant_TNRS")[, -c(5:7)]
```

```
## Calling http://taxosaurus.org/retrieve/7909fdff76c3d4c5ce9195a1dc783bcf
```

```
##         submittedname         acceptedname    sourceid score
## 7   Helianthus annuus   Helianthus annuus iPlant_TNRS     1
## 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 turns out there are a few corrections: e.g., *Madia sateva* should be *Madia sativa*, and *Rosa california* should be *Rosa californica*. Note that this search worked because fuzzy matching was employed to retrieve names that were close, but not exact matches. Fuzzy matching is only available for plants in the TNRS service, so we advise using EOL's Global Names Resolver if you need to resolve animal names.

taxize takes the approach that the user should be able to make decisions about what resource to trust, rather than 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. In the future, we may provide the ability for taxize to suggest the best match from a variety of sources.

Another common use case is when there are many synonyms for a species. In this example, we have three synonyms of the currently accepted name for a species.

```
mynames <- c("Helianthus annuus ssp. jaegeri", "Helianthus annuus ssp. lenticularis",
    "Helianthus annuus ssp. texanus")
tsn <- get_tsn(mynames)
library(plyr)
ldply(tsn, itis_acceptname)
```

```
  submittedtsn      acceptedname acceptedtsn
1       525928 Helianthus annuus       36616
2       525929 Helianthus annuus       36616
3       525930 Helianthus annuus       36616
```

**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. For example, you may find out that species A and species B are in Family C, which may lead to some interesting insight, 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 databases).

A number of data sources in taxize provide the capability to retrieve higher taxonomic names, but we will highlight two of the more useful ones: Integrated Taxonomic Information System (ITIS) and National Center for Biotechnology Information (NCBI). First, we'll search for two species, *Abies procera} and* Pinus contorta* within ITIS.

```
specieslist <- c("Abies procera", "Pinus contorta")
classification(specieslist, db = "itis")
```

```
##
## Retrieving data for taxon 'Abies procera'
##
##
## Retrieving data for taxon 'Pinus contorta'
```

```
## $`Abies procera`
##                 name         rank
## 1            Plantae      Kingdom
## 2    Viridaeplantae    Subkingdom
## 3      Streptophyta  Infrakingdom
## 4      Tracheophyta      Division
## 5   Spermatophytina   Subdivision
## 6      Gymnospermae Infradivision
## 7         Pinopsida         Class
## 8           Pinales         Order
## 9          Pinaceae        Family
## 10            Abies         Genus
## 11    Abies procera       Species
##
## $`Pinus contorta`
##                 name         rank
## 1            Plantae      Kingdom
## 2    Viridaeplantae    Subkingdom
## 3      Streptophyta  Infrakingdom
## 4      Tracheophyta      Division
## 5   Spermatophytina   Subdivision
## 6      Gymnospermae Infradivision
## 7         Pinopsida         Class
## 8           Pinales         Order
## 9          Pinaceae        Family
## 10            Pinus         Genus
## 11   Pinus contorta       Species
##
## attr(,"class")
## [1] "classification"
## attr(,"db")
## [1] "itis"
```

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

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. As with the `classification` function you can specify the data source with the `db` argument, either ITIS or NCBI.

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

```
##
## Retrieving data for taxon 'Helianthus annuus'
```

```
##       family
## 1 Asteraceae
```

I may happen that a data source does not provide information on the queried species, than one could take the result from another source and union the results from the different sources.

**Interactive name selection** As mentioned most databases use a numeric code to reference a species. A general workflow in taxize is: Retrieve Code for the queried species and then use this code to query more data/information.

Below are a few examples. When you run these examples in R, you are presented with a command prompt asking for the row that contains the name you would like back; that output is not printed below for brevity. In this example, the search term has many matches. The function returns a data.frame of the matches, and asks for the user to input what row number to accept.

```
get_tsn(searchterm = "Heliastes", searchtype = "sciname")
```

```
##
## Retrieving data for taxon 'Heliastes'
```

```
##                   target    tsn
## 1     Heliastes bicolor 615238
## 2   Heliastes chrysurus 615250
## 3     Heliastes cinctus 615573
## 4  Heliastes dimidiatus 615257
## 5  Heliastes hypsilepis 615273
## 6 Heliastes immaculatus 615639
## 7 Heliastes opercularis 615300
## 8      Heliastes ovalis 615301
```

```
##
## More than one TSN found for taxon 'Heliastes'!
##
##              Enter rownumber of taxon (other inputs will return 'NA'):
##
## Input accepted, took taxon 'Heliastes bicolor'.
```

```
## [1] "615238"
## attr(,"match")
## [1] "found"
## attr(,"class")
## [1] "tsn"
```

In another example, you can pass in a long character vector of taxonomic names:

```
splist <- c("annona cherimola", "annona muricata", "quercus robur")
get_tsn(searchterm = splist, searchtype = "sciname")
```

```
##
## Retrieving data for taxon 'annona cherimola'
##
##
## Retrieving data for taxon 'annona muricata'
##
##
## Retrieving data for taxon 'quercus robur'
## [1] "506198" "18098"  "19405"
## attr(,"match")
## [1] "found" "found" "found"
## attr(,"class")
## [1] "tsn"
```

**What taxa are the children of my taxon of interest?** If someone is not a taxonomic specialist on a particular taxon he likely does not know what children taxa are within a family, or within a genus. This task becomes especially unwieldy when there are a large number of taxa downstream. You can of course go to a website like Wikispecies or Encyclopedia of Life to get downstream names. However, taxize provides an easy way to programatically search for downstream taxa, both for the Catalogue of Life (CoL) and the Integrated Taxonomic Information System. 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 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
```

The result from the above call to `col_downstream()` is a data.frame that gives a number of columns of different information.

**Matching species tables with different taxonomic resolution** Biologist often need to match different sets of data tied to species. For example, trait-based approaches are a promising tool in ecology. One problem is that abundance data must be matched with trait databases. These two data tables may contain species information on different taxonomic levels and possibly data must be aggregated to a joint taxonomic level, so that the data can be merged. taxize can help in this data-cleaning step, providing a reproducible workflow:

We can use the mentioned `classification`-function to retrieve the taxonomic hierarchy and then search the hierarchies up- and downwards for matches. Here is an example to match a species with names on three different taxonomic levels.

```
A <- "gammarus roeseli"

B1 <- "gammarus roeseli"
B2 <- "gammarus"
B3 <- "gammaridae"

A_clas <- classification(A, db = 'ncbi')
```

```
##
## Retrieving data for taxon 'gammarus roeseli'
```

```
B1_clas <- classification(B1, db = 'ncbi')
```

```
##
## Retrieving data for taxon 'gammarus roeseli'
```

```
B2_clas <- classification(B2, db = 'ncbi')
```

```
##
## Retrieving data for taxon 'gammarus'
```

```
B3_clas <- classification(B3, db = 'ncbi')
```

```
##
## Retrieving data for taxon 'gammaridae'
```

```
B1[match(A, B1)]
```

```
## [1] "gammarus roeseli"
```

```
A_clas[[1]]$Rank[tolower(A_clas[[1]]$ScientificName) %in% B2]
```

```
## NULL
```

```
A_clas[[1]]$Rank[tolower(A_clas[[1]]$ScientificName) %in% B3]
```

```
## NULL
```

If we find a direct match (here *Gammarus roeseli*), we are lucky. But we can also match Gammaridae with *Gammarus roeseli*, but on a lower taxonomic level. A more comprehensive and realistic example (matching a trait table with an abundance table) is given in the vignette on matching.