taxize: taxonomic search and retrieval in R

Scott A. Chamberlain[1] **and Eduard Szöcs**[2]

[1]**Biology Department, Simon Fraser University, 8888 University Dr, Burnaby, BC, Canada, V5A 1S6, myrmecocystus@gmail.com**
[2]**Institute for Environmental Sciences, University Koblenz-Landau, Fortstr. 7, 76829 Landau, Germany, szoe8822@uni-landau.de**

Corresponding author: Scott A Chamberlain (myrmecocystus@gmail.com)

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 increasingly available on the web, but scientists need a way to access it in a programmatic fashion that's easy 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 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 facilitates open and reproducible science by allowing taxonomic data collection to be done in the open-source R platform.

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 objects into taxonomic context, allowing them to know close and distant relatives, find relevant literature, and more.

The use of taxonomic names is, unfortunately, not straightforward. Taxonomic names often vary due to name revisions at the generic or specific levels, lumping or splitting lower taxa (genera, species) among higher taxa (families), and name spelling changes. For example, a study found that a compilation of 308,000 plant observations from 51 digitized herbarium records had 22,100 unique taxon names, of which only 13,000 were accepted names [1, 2]. In addition, there is no one authoritative taxonomic names source for all taxa - although, there are taxon specific sources that are used by many scientists. Different sources (e.g., uBio, Tropicos, ITIS) may use different accepted names for the same taxon. For example, while the Integrated Taxonomic Information Service (ITIS) has *Helianthus x glaucus* as an accepted name, The Plant List [3] ~~has~~ that name as unresolved. But *Helianthus glaucus* is an accepted name in The Plant List, while ITIS does not list this name.

One attempt to help inconsistencies in taxonomy is the use of numeric codes. For example, ITIS assigns a Taxonomic Serial Number (TSN) to each taxon, while the Universal Biological Indexer and Organizer (uBio) assigns each taxon a NameBank identifier (namebankID), and Tropicos assigns their own identifier to each taxon. Codes are helpful within a database as they can easily refer to, for example, *Helianthus annuus* with a code like 123456 instead of its whole name. However, each database uses their own code; in this case for *Helianthus annuus*, ITIS uses 36616, uBio uses 2658020, and Tropicos uses 40022652. ~~Yet, there are no universal codes for taxa across databases, leading to additional confusion.~~ Last, name comparisons across databases have to be done with the actual names, not the codes.

Taxonomic data is getting easier to obtain through the web (e.g., [4]). 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. Programatically getting taxonomic names solves the problem by looping over a list of names. In addition, doing taxonomic searching, etc. becomes reproducible. With increasing reports of irreproducibility in science [5, 6], it is extremely important to make science workflows repeatable. Science workflows can now easily incorporate text, code, and images in a single executable document [7]. Reproducible documents should become mainstream in biology to avoid mistakes, and make collaboration easier.

The R language is ~~a~~ widely used ~~language~~ by biologists, and now has over 5,000 packages on the Comprehensive R Archive Network (CRAN) to extend R. R is great for manipulating, visualizing and fitting statistical models to data. [8] gives a detailed discussion of advantages of R in computational biology. Getting data from the web will be increasingly common as more and more data gets moved to the cloud. Therefore there is a need to get data from the web directly into R. Increasingly, data is available from the web via application programming interfaces (API). These allow computers to talk to one another using code that is not human readable, but is machine readable. Web APIs often define a number of methods that allow users to search for a species name, or retrieve the synonyms for a species name, for example. A further advantage of APIs is that they are language agnostic, meaning that data can be consumed in almost any computing context, allowing users to interact with the web API without having to know the details of the code. Moreover data can be accessed from every computer, whereas for example an Excel file can only be opened in a few programs.

The goal of taxize, an R package in development, is to make many use cases ~~having to do with~~ retrieving and resolving taxonomic names easy and reproducible. In taxize, we have written a suite of R functions that interact with many taxonomic data sources via their web APIs (Table 1). The interface to each function is usually a simple list of species names, just as a user would ~~do~~ when interacting with a website. Therefore, we hope moving from a web to R interface for taxonomic names will be relatively seamless (if one is already nominally familiar with R).

Here, we justify the need for programmatic taxonomic resolution tools like taxize, discuss our data sources, and run through a suite of use cases to demonstrate the variety of ways that users can use taxize.

Why we need 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 [22] provides the ability to search for taxonomic names in documents and normalize those names found. In addition, there are many web interfaces to search for and normalize names such as Encyclopedia of Life's Global Names Resolver [15], uBio tools [21], and iPlant's Taxonomic Name Resolution Service [19].

All of these data repositories provide ways to search for taxonomic names and resolve them in some cases. However, scientists ideally need a tool that is free and can be used programmatically, thereby facilitating reproducible research. 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

Table 1. Some key functions in taxize, what they do, and their data sources

Function name	What it does	Source
apg_lookup	Changes names to match the APGIII list	Angiosperm Phylogeny Group [9]
classification	Upstream classification	Various
col_children	Direct children	Catalogue of Life [10]
col_downstream	Downstream taxa to specified rank	Catalogue of Life [10]
eol_hierarchy	Upstream classification	Encyclopedia of Life [4]
eol_search	Search EOL taxon information	Encyclopedia of Life [4]
get_seqs	Get NCBI sequences	National Center for Biotechnology Information [11]
get_tsn	Get ITIS TSN	Integrated Taxonomic Information System [12]
get_uid	Get NCBI UID	National Center for Biotechnology Information [11]
searchbycommonname	Search ITIS by common name	Integrated Taxonomic Information System [12]
searchbyscientificname	Search ITIS by scientific name	Integrated Taxonomic Information System [12]
gisd_isinvasive	Invasiveness status	Global Invasive Species Database [13]
gni_parse	Parse scientific names into components	Global Names Index [4, 14]
gni_search	Search EOL's global names index	Global Names Index [4, 14]
gnr_resolve	Resolve names using EOL's global names index	Global Names Resolver [4, 15]
itis_downstream	Downstream taxa to specified rank	Integrated Taxonomic Information System [12]
iucn_status	IUCN status	IUCN Red List [16]
phylomatic_tree	Get a plant Phylogeny	Phylomatic [17]
plantminer	Search Plantminer	Plantminer [18]
tax_name	Get taxonomic name for specific rank	Various
tax_rank	Get rank of a taxonomic name	Various
tnrs	Resolve names using iPlant	iPlant Taxonomic Name Resolution Service [19]
tp_acceptednames	Check for accepted names using Tropicos	Tropicos [20]
tpl_search	Search the Plant List	The Plant List [3]
ubio_namebank	Search uBio	uBio [21]

with species.

Data sources and package details

taxize uses many data sources (Table 1), 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 (Phylomatic) returns phylogenies, while another (NCBI) returns genetic sequence data. However, there are other R packages that are focused solely on sequence data, such as rsnps [23], rentrez [24], BoSSA [25], and ape [26], so taxize does not venture deeply into these other domains.

Some of the data sources taxize interacts with require authentication. That is, in addition to the search terms the user provides (e.g., *Homo sapiens*), the data provider requires an alphanumeric identification key so that they can better manage their servers, collect analytics, and shut down users that abuse the API. The services that require an API key in taxize are: Encyclopedia of Life (EOL) [4], the Universal Biological Indexer and Organizer (uBio) [21], Tropicos [20], and Plantminer [18]. One can easily obtain API keys by visiting the website of each service (see Table 1 for links to each site). There are two typical ways of using API keys. First, you can pass in your API key in a function call (e.g., *ubio_namebank(srchName='Ursus americanus', key='your_alphanumeric_key')*). Second, you can store your key in the .Rprofile file, which is a common place to store settings. We recommend the sec-

ond option as it simplifies function calls as taxize detects the stored keys.

One available data source in taxize is The Plant List [3]. The connection in taxize is done via the *taxonstand* package [27] that solely interacts with ~~that~~ The Plant List. We provide a few convenience functions that wrap *taxonstand* into taxize.

taxize would not have been possible without the work of others. taxize uses httr [28] and RCurl [29] for doing calls to web APIs, XML [30] for parsing XML, RJSONIO [31] for parsing JSON, and stringr [32] and plyr [33] for manipulating data.

New data sources can be added; for example, we plan to add the following sources: Wikispecies and The Tree of Life. A connection to `freshwaterecology.info` [34] (a database with autecological characteristics, ecological preferences and biological traits as well as distribution patterns of more than 12,000 European freshwater organisms belonging to fish, macro-invertebrates, macrophytes, diatoms and phytoplankton) will be finished when their new API will be released. In addition, the authors may be contacted for further suggestions of data sources to be added.

Use cases

First, install taxize

First, one must install and load taxize into the R session.

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

Advanced users can also download and install the latest development copy from GitHub [35].

Resolve taxonomic names

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 [15]. Here, we are searching for two misspelled names:

```
temp <- gnr_resolve(names = c("Helianthos annus",
"Homo saapiens"))
temp[, -c(1, 4)]
```

```
                matched_name       data_source_title
1        Helianthus annuus L.       Catalogue of Life
2           Helianthus annus GBIF Taxonomic Backbone
3           Helianthus annus                     EOL
4         Helianthus annus L.                     EOL
5           Helianthus annus           uBio NameBank
6 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 [36] 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)[, -c(5:7)]
```

```
         submittedName        acceptedName    sourceId score
9    Helianthus annuus   Helianthus annuus iPlant_TNRS     1
10   Helianthus annuus   Helianthus annuus        NCBI     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
7      Rosa california    Rosa californica iPlant_TNRS  0.99
8      Rosa california          California        NCBI     1
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 californa* 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",
"Helianthus annuus var. lenticularis",
"Helianthus annuus var. macrocarpus",
"Helianthus annuus var. texanus")
tsn <- get_tsn(mynames)
ldply(tsn, itis_acceptname)
```

```
  submittedTsn      acceptedName acceptedTsn
1       525928 Helianthus annuus       36616
2       525929 Helianthus annuus       36616
3       525930 Helianthus annuus       36616
4       536095 Helianthus annuus       36616
5       536096 Helianthus annuus       36616
6       536097 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) [12] and National Center for Biotechnology Information (NCBI) [11]. First, we'll search for two species, *Abies procera* and *Pinus contorta* within ITIS.

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

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

$`Pinus contorta`
        rankName       taxonName    tsn
1        Kingdom         Plantae 202422
2     Subkingdom  Viridaeplantae 846492
3   Infrakingdom    Streptophyta 846494
4       Division    Tracheophyta 846496
5    Subdivision Spermatophytina 846504
6  Infradivision    Gymnospermae 846506
7          Class       Pinopsida 500009
8          Order         Pinales 500028
9         Family        Pinaceae  18030
10         Genus           Pinus  18035
11       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(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 = "itis")
```

```
      family
1 Asteraceae
```

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

```
      family
1 Asteraceae
```

It may happen that a data source does not provide information on the queried species, then 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")
```

```
           combinedname    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
 1
NA
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", "shorea robusta", "pandanus patina",
"oryza sativa", "durio zibethinus")
get_tsn(searchterm = splist, searchtype = "sciname")
```

```
[1] "506198" "18098"  "19405"  "506787" "507376" "41976"  "506099"
attr(,"class")
[1] "tsn"
```

In another example, note that no match at all returns an NA:

```
get_uid(sciname = c("Chironomus riparius", "aaa vva"))
```

```
[1] "315576" NA
attr(,"class")
[1] "uid"
```

Retrieve a phylogeny

Ecologists are increasingly taking a phylogenetic approach to ecology, applying phylogenies to topics such as the study of community structure [37], ecological networks [38], functional trait ecology [39]. Yet, ~~M~~many biologists are not adequately trained in reconstructing phylogenies. Fortunately, there are some sources for getting a phylogeny without having to know how to build one; one of these is for angiosperms, called Phylomatic [17]. We have created a workflow in taxize that accepts a species list, and taxize works behind the scenes to get higher taxonomic names, which are required by Phylomatic to get a phylogeny. Here is a short example, producing the tree in Fig. 1.

```
taxa <- c("Poa annua", "Abies procera", "Helianthus annuus")
tree <- phylomatic_tree(taxa = taxa)
tree$tip.label <- capwords(tree$tip.label)
plot(tree, cex = 1)
```

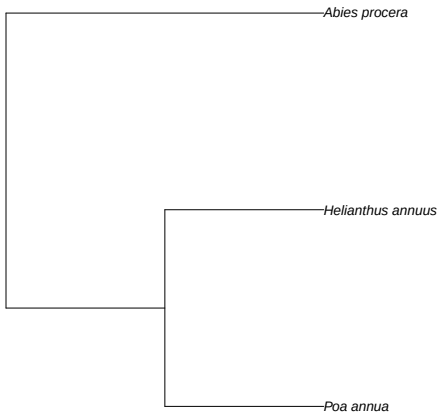

Figure 1. ~~a~~ A phylogeny for three species. This phylogeny was produced using the *phylomatic_tree* function, which queries the Phylomatic database, and prunes a previously created phylogeny of plants.

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'). These strings are submitted to the Phylomatic API, and if no errors occur, a phylogeny in newick format is returned. The *phylomatic_tree()* function also cleans up the newick string and converts it to a **ape** *phylo* object. The output from *phylomatic_tree()* is a *phylo* object, which can be used for plotting and phylogenetic analyses. Be aware that Phylomatic has certain limitations - refer to the paper describing Phylomatic [17] and the website `http://phylodiversity.net/phylomatic/`.

There are currently no resources for getting a phylogeny of animals simply from species names. However, a few projects are working on this problem, including the Open Tree of Life [40]. We will incorporate these resources when the appropriate APIs are available.

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 [41] or Encyclopedia of Life [4] to get downstream names. However, taxize provides an easy way to programatically search for downstream taxa, both for the Catalogue of Life (CoL) [10] and the Integrated Taxonomic Information System [12]. 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.

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 (IUCN Red List of Threatened Species [16]). 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 ~~P~~anther and ~~L~~ynx.

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

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

It turns out that the panther has a status of endangered (EN) and the lynx has a status of least concern (LC).

Search for available genes in GenBank

Another use case available in taxize deals with genetic sequences. taxize has three functions to interact with GenBank to search for available genes (*get_genes_avail*), download genes by GenBank ID (*get_genes*), and download genes via taxonomic name search, including retrieving a congeneric if the searched taxon does not exist in the database (*get_seqs*). In this example, we search for gene sequences for *Umbra limi*.

```
out <- get_genes_avail(taxon_name = "Umbra limi",
seqrange = "1:2000", getrelated = FALSE)
```

Then we can ask if 'RAG1' exists in any of the gene names.

```
out[grep("RAG1", out$genesavail, ignore.case = TRUE), -3]
```

```
        spused length access_num    ids
413 Umbra limi    732 JX190826  394772608
427 Umbra limi    959 AY459526   45479841
434 Umbra limi   1631 AY380548   38858304
```

It turns out that there are 430 different unique records found. However, this doesn't mean that there are 430 different genes found as the API does not provide metadata

to classify genes. However, you can use regular expressions (e.g., *grep*) to search for the gene of interest.

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 [42]. One problem is that abundance data must be matched with trait databases ~~like~~ [43]. 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')
B1_clas <- classification(B1, db = 'ncbi')
B2_clas <- classification(B2, db = 'ncbi')
B3_clas <- classification(B3, db = 'ncbi')


B1[match(A, B1)]


[1] "gammarus roeseli"


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


[1] "genus"


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


[1] "family"
```

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 Appendix B.

Aggregating data to a specific taxonomic rank

In biology, one can asks questions at varying taxonomic levels. One may perform analyses on different taxonomic levels. This use case is easily handled in taxize. A function called *tax_agg* will aggregate community data to a specific taxonomic level. In this example, we take data of 5 species and aggregate them to family level. Again we can specify if we want to use data from ITIS or NCBI.

```
data(dune, package = 'vegan')
df <- dune[ , 1:5]
colnames(df) <- c("Bellis perennis", "Empetrum nigrum",
"Juncus bufonius", "Juncus articulatus", "xxx")
head(df)
```

```
   Bellis perennis Empetrum nigrum Juncus bufonius Juncus articulatus xxx
2                3               0               0                   0   0
13               0               0               3                   0   0
4                2               0               0                   0   0
16               0               0               0                   3   0
6                0               0               0                   0   0
1                0               0               0                   0   0


agg <- tax_agg(df, rank = 'family', db = 'ncbi')
agg


Aggregated community data

Level of Aggregation: FAMILY
No. taxa before aggregation: 5
No. taxa after aggregation: 4
No. taxa not found: 1


head(agg$x)


   Asteraceae Ericaceae Juncaceae xxx
2           3         0         0   0
13          0         0         3   0
4           2         0         0   0
16          0         0         3   0
6           0         0         0   0
1           0         0         0   0
```

We see that the two *Juncus* species are aggregated to Juncaceae and their abundances are summed up. If a taxon is one lower taxonomic resolution ~~then~~ the queried or the taxon is not found in the database then these are not aggregated and returned as is.

Conclusions

Taxonomic information is increasingly sought out by biologists as we take phylogenetic and taxonomic approaches to science. Taxonomic data ~~is~~ becoming more widely available on the web, yet scientists require programmatic access to this data for developing reproducible workflows. taxize was created to bridge this gap - to bring taxonomic data on the web into R, where the data can be easily manipulated, visualized, and analyzed in a reproducible workflow.

We have outlined a suite of use cases in taxize that will likely fit real use cases ~~of~~ many biologists. Of course we have not thought of all possible use cases, so we hope that the biology community can give us feedback on what use cases they want to see available in taxize. One thing we could change in the future is to make functions that fit use cases, and then allow users to select the data source as a parameter in the function. This could possibly make the user interface easier to understand.

taxize is currently under development and will be for some time given the large number of data sources ~~put~~ together in the package, and the fact that APIs for each data source can change, requiring changes in taxize code. Contributions to taxize are strongly encouraged, and can be easily done using GitHub ~~here~~ [35]. We hope taxize will be ~~taking~~ up by the community and developed collaboratively, making it progressively better through time as new use cases arise, bug reports squashed, and contri-

butions merged.

Author contributions

In order to give appropriate credit to each author of an article, the individual contributions of each author to the manuscript should be detailed in this section. We recommend using author initials and then stating briefly how they contributed.

Competing interests

No competing interests.

Grant information

~~No funding was recieved for this work.~~

Acknowledgements

The taxize package is part of the rOpenSci project `http://ropensci.org/`. We thank Carl Boettiger, Karthik Ram, Owen Jones, Naim Matasci, and Ralf Schäfer for comments on previous versions of this manuscript. We thank all API maintainers for their work making their databases open to the public.

References

[1] Michael D. Weiser, Brian J. Enquist, Brad Boyle, Timothy J. Killeen, Peter M. Jørgensen, Gustavo Fonseca, Michael D. Jennings, Andrew J. Kerkhoff, Thomas E. Lacher Jr, Abel Monteagudo, and et al. Latitudinal patterns of range size and species richness of new world woody plants. *Global Ecology and Biogeography*, 16(5):679–688, sep 2007.

[2] Brad Boyle, Nicole Hopkins, Zhenyuan Lu, Juan Antonio Raygoza Garay, Dmitry Mozzherin, Tony Rees, Naim Matasci, Martha L Narro, William H Piel, Sheldon J Mckay, and et al. The taxonomic name resolution service: an online tool for automated standardization of plant names. *BMC Bioinformatics*, 14(1):16, 2013.

[3] The Plant List. A working list of all plant species. Available:`http://www.theplantlist.org`, 2013. Accessed May 27 2013.

[4] Encylopedia of Life. Available: `http://eol.org/`, 2013. Accessed May 27 2013.

[5] Victoria C Stodden. Reproducible research: Addressing the need for data and code sharing in computational science. *Computing in Science & Engineering*, 12(5):8–12, 2010.

[6] Carl Zimmer. A sharp rise in retractions prompts calls for reform. *New York Times*, 2012.

[7] Yihui Xie. *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, 2013.

[8] Robert C. Gentleman, Vincent J. Carey, Douglas M. Bates, Ben Bolstad, Marcel Dettling, Sandrine Dudoit, Byron Ellis, Laurent Gautier, Yongchao Ge, Jeff Gentry, Kurt Hornik, Torsten Hothorn, Wolfgang Huber, Stefano Iacus, Rafael Irizarry, Friedrich Leisch, Cheng Li, Martin Maechler, Anthony J. Rossini, Gunther Sawitzki, Colin Smith, Gordon Smyth, Luke Tierney, Jean YH Yang, and Jianhua Zhang. Bioconductor: open software development for computational biology and bioinformatics. *Genome Biology*, 5(10):R80, September 2004. PMID: 15461798.

[9] Angiosperm Phylogeny Group. Available: `http://www.mobot.org/MOBOT/research/APweb/`, 2013. Accessed May 27 2013.

[10] Y. Roskov, T. Kunze, L. Paglinawan, T. Orrell, D. Nicolson, A. Culham, N. Bailly, P. Kirk, T. Bourgoin, G. Baillargeon, F. Hernandez, and A. De Wever. Catalogue of Life. Available: `http://www.catalogueoflife.org/`, 2013. Accessed May 27 2013.

[11] Scott Federhen. The ncbi taxonomy database. *Nucleic Acids Research*, 40(D1):D136–D143, 2012.

[12] ITIS. Integrated taxonomic information service. Available: `http://www.itis.gov/`, 2013. Accessed May 27 2013.

[13] Global Invasive Species Database. Global invasive species database. Available: `http://www.issg.org/database/welcome/`, 2013. Accessed May 27 2013.

[14] Global Names Index. Available: `http://gni.globalnames.org/`, 2013. Accessed May 27 2013.

[15] Global Names Resolver. Available: `http://resolver.globalnames.org/`, 2013. Accessed May 27 2013.

[16] IUCN. Iucn red list of threatened species. Available: `http://www.iucnredlist.org`, 2013. Accessed May 27 2013.

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

[18] Gustavo Henrique Carvalho, Marcus Vinicius Cianciaruso, and Marco Antônio Batalha. Plantminer: a web tool for checking and gathering plant species taxonomic information. *Environmental Modelling & Software*, 25(6):815–816, 2010.

[19] TNRS. Taxonomic name resolution service. Available: `http://tnrs.iplantcollaborative.org/`, 2013. Accessed May 27 2013.

[20] Missouri Botanical Garden. Tropicos.org. Available: `http://www.tropicos.org/`, 2013. Accessed May 27 2013.

[21] uBio. Universal biological indexer and organizer. Available: `http://www.ubio.org/index.php?pagename=sample_tools`, 2013. Accessed May 27 2013.

[22] Linnaeus. Available: `http://linnaeus.sourceforge.net/`, 2013. Accessed May 27 2013.

[23] Scott Chamberlain and Kevin Ushey. *rsnps: Interface to SNP data on the web.*, 2013. R package version 0.0.4.

[24] David Winter. *rentrez: Entrez in R*, 2013. R package version 0.2.1.

[25] Pierre Lefeuvre. *BoSSA: a Bunch of Structure and Sequence Analysis*, 2010. R package version 1.2.

[26] E. Paradis, J. Claude, and K. Strimmer. APE: analyses of phylogenetics and evolution in R language. *Bioinformatics*, 20:289–290, 2004.

[27] Luis Cayuela, Íñigo Granzow-de la Cerda, Fabio S. Albuquerque, and Duncan J. Golicher. taxonstand: An r package for species names standardisation in vegetation databases. *Methods in Ecology and Evolution*, 3(6):1078–1083, 2012.

[28] Hadley Wickham. *httr: Tools for working with URLs and HTTP*, 2012. R package version 0.2.

[29] Duncan Temple Lang. *RCurl: General network (HTTP/FTP/...) client interface for R*, 2013. R package version 1.95-4.1.

[30] Duncan Temple Lang. *XML: Tools for parsing and generating XML within R and S-Plus.*, 2013. R package version 3.95-0.2.

[31] Duncan Temple Lang. *RJSONIO: Serialize R objects to JSON, JavaScript Object Notation*, 2013. R package version 1.0-3.

[32] Hadley Wickham. *stringr: Make it easier to work with strings.*, 2012. R package version 0.6.2.

[33] Hadley Wickham. The split-apply-combine strategy for data analysis. *Journal of Statistical Software*, 40(1):1–29, 2011.

[34] A. Schmidt-Kloiber and D. Hering. www.freshwaterecology.info - the taxa and autecology database for freshwater organisms, version 5.0. Available: `www.freshwaterecology.info`, May 2013.

[35] taxize. taxize on github. Available: `https://github.com/ropensci/taxize_`, 2013.

[36] Taxosaurus. The taxonomic thesaurus. Available: `http://taxosaurus.org/`, 2013. Accessed May 27 2013.

[37] Campbell O Webb, David D Ackerly, Mark A McPeek, and Michael J Donoghue. Phylogenies and community ecology. *Annual Review of Ecology and Systematics*, pages 475–505, 2002.

[38] Nicole E Rafferty and Anthony R Ives. Phylogenetic trait-based analyses of ecological networks. *Ecology*, 2013.

[39] N LeRoy Poff, Julian D Olden, Nicole KM Vieira, Debra S Finn, Mark P Simmons, and Boris C Kondratieff. Functional trait niches of north american lotic insects: traits-based ecological applications in light of phylogenetic relationships. *Journal of the North American Benthological Society*, 25(4):730–755, 2006.

[40] Open Tree of Life. Available: `http://blog.opentreeoflife.org/`, 2013. Accessed May 27 2013.

[41] Wikispecies. Available: `http://species.wikimedia.org/wiki/Main_Page`, 2013. Accessed May 27 2013.

[42] B. Statzner and L.A Bêche. Can biological invertebrate traits resolve effects of multiple stressors on running water ecosystems? *Freshwater Biology*, 55:80–119, 2010.

[43] Philippe Usseglio-Polatera, Michel Bournaud, Philippe Richoux, and Henri Tachet. Biological and ecological traits of benthic freshwater macroinvertebrates: relationships and definition of groups with similar traits. *Freshwater Biology*, 43(2):175–205, 2000.

[44] National Center for Biotechnology Information. Taxonomy database. Available: `http://www.ncbi.nlm.nih.gov/taxonomy`, 2013. Accessed May 27 2013.

[45] Donald J Baird, Christopher J O Baker, Robert B Brua, Mehrdad Hajibabaei, Kearon McNicol, Timothy J Pascoe, and Dick de Zwart. Toward a knowledge infrastructure for traits-based ecological risk assessment. *Integrated Environmental Assessment and Management*, 7(2):209–215, 2011.

[46] Michael Kleyer, Stéphane Dray, Francescode Bello, Jan Leps, Robin J. Pakeman, Barbara Strauss, Wilfried Thuiller, and Sandra Lavorel. Assessing species and community functional responses to environmental gradients: which multivariate methods? *Journal of Vegetation Science*, 23(5):805–821, 2012.

[47] Luis Cayuela. *Taxonstand: Taxonomic standardization of plant species names*, 2012. R package version 1.0.

[48] E. Paradis, J. Claude, and K. Strimmer. APE: analyses of phylogenetics and evolution in R language. *Bioinformatics*, 20:289–290, 2004.