Package: taxonlookup (via r-universe)

May 7, 2026

Title A dynamically-updating versioned taxonomic resource for vascular plants

Version 1.1.5

Date 2016-01-29

Description This is a taxon lookup table for land plants. It is built from a set of scripts that dynamically build a versioned genus-family-order-higher taxa lookup table from canonical sources on the web. It uses semantic versioning to keep track of changes.

BugReports https://github.com/traitecoevo/taxonlookup/issues

Depends R (>= 3.1.0)

License MIT + file LICENSE

LazyData true

Maintainer Will Cornwell <wcornwell@gmail.com>

Imports datastorr (>= 0.0.3)

Suggests testthat

Remotes github::ropenscilabs/datastorr

RoxygenNote 6.0.1

Config/pak/sysreqs libicu-dev libssl-dev

Repository https://traitecoevo.r-universe.dev

Date/Publication 2019-02-20 01:04:58 UTC

RemoteUrl https://github.com/traitecoevo/taxonlookup

RemoteRef HEAD

RemoteSha 38e0ef646cc90991f781d40629c71322ad35c159

Contents

`add_higher_order` *Add higher order taxonomy*

Description

Augment the genus-family-order lookup table with plant higher taxa data, from http://datadryad.org/resource/doi:10.5061/dry

Usage

```
add_higher_order(lookup = plant_lookup(include_counts = TRUE),
  order_column = "order")
```

Arguments

`lookup`	A lookup table (by default `plant_lookup()`)
`order_column`	The column within `lookup` that corresponds to taxonomic order.

Details

Data for the higher-level taxonomy lookup (http://datadryad.org/resource/doi:10.5061/dryad.63q27.2/1.1) compiled by Dave Tank and colleagues

Because of the currently non-nested structure of higher clade information for plants, the format of the lookup is also not nested. In the lookup the higher nodes are columns within the data.frame. If a genus is a descendent of a particular node, the row for that genus repeats the node name (which is also the column name). If a genus is not a descendent of that particular node, the cell is left blank.

Examples

```
##' # see the format of the resource
head(add_higher_order())
#
# load the data.frame into memory
ho<-add_higher_order()
#
# return Rosid orders
unique(ho$order[ho$Rosidae=="Rosidae"])
#
# find the number of Conifer species in the world
sum(ho$number.of.accepted.species[ho$Coniferae=="Coniferae"])
```

lookup_table *Build a family and order lookup table for a set of species*

Description

Build a lookup table for a set of species, connecting the species names to plant genera, families and orders

Usage

```
lookup_table(species_list, lookup_table = NULL, genus_column = "genus",
  missing_action = c("drop", "NA", "error"), by_species = FALSE,
  family.tax = "apweb", include_counts = FALSE, ...)
```

Arguments

species_list	Character vector of species bionomials Genus and species may be seperated by " " or "_"
lookup_table	Any higher taxonomy lookup table, but by default plant_lookup
genus_column	The column within lookup_table that corresponds to genus. By default this is "genus", which is the correct name for plant_lookup.
missing_action	How to behave when there are genera in the species_list that are not found in the lookup table. "drop" (the default) generates a table without these genera, "NA" will leave the non-genus taxonomic levels as missing values and error will throw an error.
by_species	If TRUE, then return a larger data frame with one row per species and with species as row names.
family.tax	There are two available family taxonomies–"apweb" and "plantlist". the default is "apweb"
include_counts	This will include the number of species in the genus (data from the plant list) in the output
...	variables passed on to plant_lookup, including version number.

Details

The data within this lookup table comes from two sources:

1. The Plant List v1.1. (http://www.theplantlist.org/) for accepted genera to families and species richness within each genera. Note that we do not consider hybrids (e.g. Genus X species) as distinct species for this count while the plant list summary statistics do, so the the counts from this package will not line up exactly with the ones on the TPL website.

2. APWeb (http://www.mobot.org/MOBOT/research/APweb/) for family-level synonymies and family-to-order for all vascular plant families. Note that there is not currently order-level information available for Bryophytes.

Examples

```
lookup_table("Pinus_ponderosa")
#
# or with a space
#
lookup_table("Pinus ponderosa")
#
#
# control how you want the function to handle non-matches
#
lookup_table(c("Pinus ponderosa","Neitheragenus noraspecies"))
#
lookup_table(c("Pinus ponderosa","Neitheragenus noraspecies"),missing_action="NA")
```

`plant_lookup` *Plant taxonomy lookup table*

Description

Lookup table relating plant genera, families and orders along with number of species in each genus. Data persists across package installations.

Usage

```
plant_lookup(version = NULL, include_counts = FALSE, family.tax = "apweb",
  path = NULL)

plant_lookup_versions(local = TRUE, path = NULL)

plant_lookup_version_current_local(path = NULL)

plant_lookup_version_current_github(path = NULL)

get_most_recent_plant_lookup(path = NULL)

plant_lookup_del(version, path = NULL)
```

Arguments

`version`	Version number. The default will load the most recent version on your computer or the most recent version known to the package if you have never downloaded the data before. With `plant_lookup_del`, specifying `version=NULL` will delete *all* data sets.
`include_counts`	Logical: Include a column of number of "accepted" species within each genus counts as `number.of.species`.

family.tax	the value "ap.web" will return the family names from apweb otherwise the lookup will include the family names from the plant list. Currently there are 8 family names that differ between the two sources (e.g., Compositae in the plant list versus Asteraceae in ap.web)
path	Path to store the data at. If not given, datastorr will use rappdirs to find the best place to put persistent application data on your system. You can delete the persistent data at any time by running mydata_del(NULL) (or mydata_del(NULL, path) if you use a different path).
local	Logical indicating if local or github versions should be pulled. With any luck, local=FALSE is a superset of local=TRUE. For mydata_version_current, if TRUE, but there are no local versions, then we do check for the most recent github version.

Details

The data within this lookup table comes from two sources:

1. The Plant List v1.1. (http://www.theplantlist.org/) for accepted genera to families and species richness within each genera. Note that we do not consider hybrids (e.g. Genus X species) as distinct species for this count while the plant list summary statistics do, so the the counts from this package will not line up exactly with the ones on the TPL website.

2. APWeb (http://www.mobot.org/MOBOT/research/APweb/) for family-level synonymies and family-to-order for all vascular plant families. Note that there is not currently order-level information available for Bryophytes.

These data are then currated–we correct some spelling errors, special character issues, genera listed in multiple families, family-level synonomy, and other issues that arise in assembling a resources like this. Details of the curation are at https://github.com/traitecoevo/taxonlookup

Examples

```
#
# see the format of the resource
#
head(plant_lookup())
#
# or with number of species in each genus.
#
head(plant_lookup(include_counts = TRUE))
#
# load the data.frame into memory
#
pl<-plant_lookup(include_counts = TRUE)
#
# return family, order, and number of species for the genus Eucalyptus
#
pl$family[pl$genus=="Eucalyptus"]
pl$order[pl$genus=="Eucalyptus"]
pl$number.of.species[pl$genus=="Eucalyptus"]
#
# find the number of accepted species within the Myrtaceae
```

```
#
sum(pl$number.of.species[pl$family=="Myrtaceae"])
```

Index