

# Package ‘rvertnet’

December 2, 2015

**Title** Search 'Vertnet', a 'Database' of Vertebrate Specimen Records

**Description** Retrieve, map and summarize data from the 'VertNet.org' archives. Functions allow searching by many parameters, including 'taxonomic' names, places, and dates. In addition, there is an interface for conducting spatially delimited searches, and another for requesting large 'datasets' via email.

**Version** 0.4.1

**Author** Scott Chamberlain [aut, cre], Chris Ray [aut], Vijay Barve [aut]

**Maintainer** Scott Chamberlain <myrmecocystus@gmail.com>

**License** MIT + file LICENSE

**LazyData** true

**URL** <https://github.com/ropensci/rvertnet>

**BugReports** <https://github.com/ropensci/rvertnet/issues>

**VignetteBuilder** knitr

**Imports** methods, stats, utils, plyr, jsonlite, httr, dplyr, ggplot2, maps

**Suggests** testthat, knitr, RSQLite

**RoxygenNote** 5.0.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2015-12-02 09:17:21

## R topics documented:

rvertnet-package	2
bigsearch	3
dump	5
rvertnet-defunct	6
searchbyterm	6
spatialsearch	9
vertmap	10

vertsearch . . . . .	11
vertsummary . . . . .	13
vert_id . . . . .	13
<b>Index</b>	<b>15</b>

---

rvertnet-package	<i>Search VertNet archives using R</i>
------------------	--

---

## Description

There are a variety of ways to search VertNet

### Search by term

Search for `_Aves_` in the state of `_California_`, limit to 10 records, e.g.:

```
searchbyterm(class = "Aves", state = "California", lim = 10, verbose = FALSE)
```

Search for `_Mustela nigripes_` in the states of `_Wyoming_` or `_South Dakota_`, limit to 20 records, e.g.:

```
searchbyterm(genus = "Mustela", specific epithet = "nigripes", state = "(wyoming OR south dakota)", limit = 20, verbose = FALSE)
```

### Big data

Specifies a termwise search (like `'searchbyterm()'`), but requests that all available records be made available for download as a tab-delimited text file.

```
bigsearch(genus = "ochotona", rf = "pikaRecords", email = "big@search.luv")
```

### Spatial search

```
spatialsearch(lat = 33.529, lon = -105.694, radius = 2000, limit = 10, verbose = FALSE)
```

### Full text search

Find records using a global full-text search of VertNet archives.

```
vertsearch(taxon = "aves", state = "california")
```

bigsearch

*Request to download a large number of VertNet records.***Description**

Specifies a termwise search (like [searchbyterm](#)) and requests that all available records be made available for download as a tab-delimited text file.

**Usage**

```
bigsearch(specificepithet = NULL, genus = NULL, family = NULL,
  order = NULL, class = NULL, compact = FALSE, year = NULL,
  date = NULL, mappable = NULL, error = NULL, continent = NULL,
  cntry = NULL, stateprovince = NULL, county = NULL, island = NULL,
  igroup = NULL, inst = NULL, id = NULL, catalognumber = NULL,
  collector = NULL, type = NULL, hastypestatus = NULL, media = NULL,
  rank = NULL, tissue = NULL, resource = NULL, rfile, email,
  verbose = TRUE, ...)
```

**Arguments**

specificepithet	Taxonomic specific epithet, e.g. (sapiens in Homo sapiens) (character)
genus	Taxonomic genus (character)
family	Taxonomic family (character)
order	Taxonomic order (character)
class	Taxonomic class (character)
compact	Return a compact data frame (logical)
year	Year (numeric) or range of years designated by comparison operators "<", ">", "<=" or ">=". You can pass in more than one of these queries, in a vector. See example below. (character)
date	Event date associated with this occurrence record; yyyy-mm-dd or the range yyyy-mm-dd/yyyy-mm-dd (character)
mappable	Record includes valid coordinates in decimal latitude and decimal longitude (logical)
error	Coordinate uncertainty in meters (numeric) or range of uncertainty values designated by comparison operators "<", ">", "<=" or ">=" (character)
continent	Continent to search for occurrence (character)
cntry	Country to search for occurrence (character)
stateprovince	State or province to search for occurrence (character)
county	County to search for occurrence (character)
island	Island to search for occurrence (character)

igroup	Island group to search for occurrence (character)
inst	Code name for the provider/institution of record (character)
id	Provider's unique identifier for this occurrence record (character)
catalognumber	Provider's catalog number or other ID for this record (character)
collector	Collector name (character)
type	Type of record; "specimen" or "observation" (character)
hastypestatus	Specimen associated with this record is identified as a holotype, paratype, neotype, etc. (character)
media	Record also references associated media, such as a film or video (logical)
rank	TBD (numeric)
tissue	Record is likely to reference tissues (logical)
resource	Identifier for the resource/dataset from which the record was indexed (character)
rfile	A name for the results file that you will download (character). Required.
email	An email address where you can be contacted when your records are ready for download (character). Required.
verbose	Print progress and information messages. Default: TRUE
...	Curl arguments passed on to <a href="#">GET</a>

### Details

[bigsearch](#) allows you to request records as a tab-delimited text file. This is the best way to access a large number of records, such as when your search results indicate that >1000 records are available. You will be notified by email when your records are ready for download.

### Value

Prints messages on progress, but returns NULL

### Reading data

We suggest reading data in with [fread](#) - as it's very fast for the sometimes large datasets you will get from using this function, and is usually robust to formatting issues.

### References

<https://github.com/VertNet/webapp/wiki/The-API-search-function>

### Examples

```
## Not run:
# replace "big@search.luv" with your own email address
bigsearch(genus = "ochotona", rf = "pikaRecords", email = "big@search.luv")

# Pass in curl options for curl debugging
library("httr")
bigsearch(genus = "ochotona", rfile = "pikaRecords", email = "big@search.luv", config=verbose())
```

```
# Use more than one year query
bigsearch(class = "aves", year = c(">=1976", "<=1986"),
          rfile = "test-bigsearch1", email = "myrmecocystus@gmail.com")

## End(Not run)
```

---

 dump

*Use Vertnet taxon specific dump from KNB*


---

## Description

Use Vertnet taxon specific dump from KNB

## Usage

```
dump_init(path, group = "amphibians", table = NULL)

dump_tbl(x)

dump_links()
```

## Arguments

path	(character) Path to a sqlite file on your machine.
group	(character) One of mammals, reptiles, amphibians, fishes, or birds
table	(character) sqlite table name, you can use anything you like, but code defaults to the values in the group parameter
x	An object of class src_sqlite

## Details

dump\_init creates a src\_sqlite class that you can use to query the data either using SQL syntax or dplyr's R syntax. dump\_tbl is just a wrapper around tbl to create a tbl class object that you can use to feed directly into dplyr's verbs, like [select](#) and [filter](#)

## References

<http://blog.vertnet.org/post/115875718156/the-data-one-thing-about-vertnet-and-big-data>

## Examples

```
## Not run:
# You first need to create your SQLite databases, e.g, for amphibians:
## In the terminal
# wget
# https://knb.ecoinformatics.org/knb/d1/mn/v1/object/urn:uuid:afc58110-b9c1-4cf7-b46c-837bdc930a21
# mv urn\:uuid\:afc58110-b9c1-4cf7-b46c-837bdc930a21 vertnet_amphib.gz
```

```

# gunzip vertnet_amphib.gz
# sqlite3 amphibians.sqlite

## In SQLite
# sqlite> .separator ','
# sqlite> .import vertnet_amphib amphibians

# After you have a SQLite database, do
# library("dplyr")
# x <- dump_init(path = "~/github/sac/vertnetdumps/amphibians.sqlite")

# use SQL syntax
# tbl(x, sql("SELECT scientificname,title FROM amphibians LIMIT 10"))

# use R syntax
# tab <- x %>% dump_tbl()
# tab %>%
# filter(year > 2010) %>%
# select(scificname, title)

## End(Not run)

```

---

rvertnet-defunct

*Defunct functions in rvertnet*


---

### Description

- [vertavailablemaps](#): Function is now defunct, i.e., not available anymore.
- [vertlocations](#): Function is now defunct, i.e., not available anymore.
- [vertoccurrence](#): Function is now defunct, i.e., not available anymore.
- [vertoccurrencecount](#): Function is now defunct, i.e., not available anymore.
- [vertproviders](#): Function is now defunct, i.e., not available anymore.
- [verttaxa](#): Function is now defunct, i.e., not available anymore.

---

searchbyterm

*Search for records using keywords/terms to control how your query is interpreted.*


---

### Description

Returns only those records in which the targeted input is found in association with the specified search terms.

**Usage**

```
searchbyterm(specificepithet = NULL, genus = NULL, family = NULL,
  order = NULL, class = NULL, limit = 1000, compact = TRUE,
  year = NULL, date = NULL, mappable = NULL, error = NULL,
  continent = NULL, cntry = NULL, stateprovince = NULL, county = NULL,
  island = NULL, igroup = NULL, inst = NULL, id = NULL,
  catalognumber = NULL, collector = NULL, type = NULL,
  hastypestatus = NULL, media = NULL, rank = NULL, tissue = NULL,
  resource = NULL, verbose = TRUE, ...)
```

**Arguments**

specificepithet	Taxonomic specific epithet, e.g. (sapiens in Homo sapiens) (character)
genus	Taxonomic genus (character)
family	Taxonomic family (character)
order	Taxonomic order (character)
class	Taxonomic class (character)
limit	(numeric) Limit on the number of records returned. Up to 1000. We'll incorporate paging later for higher limits. See <a href="#">bigsearch</a> to get larger result sets in a text file via email.
compact	Return a compact data frame (logical)
year	Year (numeric) or range of years designated by comparison operators "<", ">", "<=" or ">=". You can pass in more than one of these queries, in a vector. See example below. (character)
date	Event date associated with this occurrence record; yyyy-mm-dd or the range yyyy-mm-dd/yyyy-mm-dd (character)
mappable	Record includes valid coordinates in decimal latitude and decimal longitude (logical)
error	Coordinate uncertainty in meters (numeric) or range of uncertainty values designated by comparison operators "<", ">", "<=" or ">=" (character)
continent	Continent to search for occurrence (character)
cntry	Country to search for occurrence (character)
stateprovince	State or province to search for occurrence (character)
county	County to search for occurrence (character)
island	Island to search for occurrence (character)
igroup	Island group to search for occurrence (character)
inst	Code name for the provider/institution of record (character)
id	Provider's unique identifier for this occurrence record (character)
catalognumber	Provider's catalog number or other ID for this record (character)
collector	Collector name (character)
type	Type of record; "specimen" or "observation" (character)

hastypestatus	Specimen associated with this record is identified as a holotype, paratype, neotype, etc. (character)
media	Record also references associated media, such as a film or video (logical)
rank	TBD (numeric)
tissue	Record is likely to reference tissues (logical)
resource	Identifier for the resource/dataset from which the record was indexed (character)
verbose	Print progress and information messages. Default: TRUE
...	Curl arguments passed on to <a href="#">GET</a>

### Details

`searchbyterm` builds a query from input parameters based on Darwin Core (dwc) terms (for the full list of terms, see <https://code.google.com/p/darwincore/wiki/DarwinCoreTerms>). The query string is appended to the base URL for VertNet search requests. View the query string for specification of dwc terms used in the search.

### Value

A data frame of search results

### References

<https://github.com/VertNet/webapp/wiki/The-API-search-function>

### Examples

```
## Not run:
# Limit the number of records returned to <1000; use bigsearch() for >1000 records
(out <- searchbyterm(class = "aves", state = "california", limit = 10))

# Find multiple species
(out <- searchbyterm(genus = "ochotona", specific epithet = "(princeps OR collaris)", limit=10))

# Specifying a single year (no quotes) or range of years (use quotes)
(out <- searchbyterm(class = "aves", state = "california", year = 1976, limit = 10))
(out <- searchbyterm(class = "aves", state = "california", year = ">=1976", limit = 10))

# Specifying a range (in meters) for uncertainty in spatial location (use quotes)
out <- searchbyterm(class = "aves", state = "nevada", error = "<25")
out <- searchbyterm(class = "aves", state = "california", year = 1976, error = "<=1000")

# Specifying records by event date (use quotes)
out <- searchbyterm(class = "aves", state = "california", date = "2009-03-25")
# ...but specifying a date range may not work
out <- searchbyterm(specific epithet = "nigripes", date = "1935-09-01/1935-09-30")

# Pass in curl options for curl debugging
library("httr")
out <- searchbyterm(class = "aves", limit = 10, config=verbose())
# out <- searchbyterm(class = "aves", limit = 500, config=timeout(1))
```



```
# Use more than one year query
searchbyterm(class = "aves", year = c(">=1976", "<=1986"))

## End(Not run)
```

---

spatialsearch	<i>Find records within some distance of a point given latitude and longitude.</i>
---------------	---

---

### Description

Searches by decimal latitude and longitude to return any occurrence record within the input distance (radius) of the input point.

### Usage

```
spatialsearch(lat, long, radius, limit = 1000, compact = TRUE,
  verbose = TRUE, ...)
```

### Arguments

lat	Latitude of the central point, in decimal degrees (numeric) Required.
long	Longitude of the central point, in decimal degrees (numeric) Required.
radius	Radius to search, in meters (numeric). There is no default value for this parameter. Required.
limit	(numeric) Limit on the number of records returned. Up to 1000. We'll incorporate paging later for higher limits. See <a href="#">bigsearch</a> to get larger result sets in a text file via email.
compact	Return a compact data frame (logical)
verbose	Print progress and information messages. Default: TRUE
...	Curl arguments passed on to <a href="#">GET</a>

### Details

[spatialsearch](#) finds all records of any taxa having decimal lat/long coordinates within a given radius (in meters) of your coordinates.

### Value

A data frame of search results

### References

<https://github.com/VertNet/webapp/wiki/The-API-search-function>

## Examples

```
## Not run:
res <- spatialssearch(lat = 33.529, long = -105.694, radius = 2000, limit = 10)

# Pass in curl options for curl debugging
library("httr")
out <- spatialssearch(lat = 33.529, long = -105.694, radius = 2000, limit = 10, config=verbose())

## End(Not run)
```

---

 vertmap

*Make a simple map to visualize VertNet data.*


---

## Description

Plots record locations on a world or regional map using latitude/longitude data returned by a VertNet search.

## Usage

```
vertmap(input = NULL, mapdatabase = "world", region = ".",
        geom = geom_point, jitter = NULL)
```

## Arguments

input	Output from <a href="#">vertsearch</a> , <a href="#">searchbyterm</a> , or <a href="#">spatialssearch</a> . Must include columns "decimallatitude" and "decimallongitude"
mapdatabase	The base map on which your data are displayed; what you choose here determines what you can choose in the region parameter; one of: county, state, usa, world, world2, france, italy, or nz
region	The region in which your data are displayed; to see region names for the "world" database layer, run <code>sort(unique(map_data("world")\$region))</code> after loading packages maps and ggplot2; to see region names for the US "state" layer, run <code>sort(unique(map_data("state")\$region))</code>
geom	Specifies the type of object being plotted; one of: <code>geom_point</code> or <code>geom_jitter</code> (do not use quotes)
jitter	If <code>geom = geom_jitter</code> , the amount by which to jitter points in width, height, or both. Default

## Details

vertmap uses decimal latitude and longitude data in records generated by an rvertnet search to display returned records on a specified base map. Taxa are color-coded by scientific name, if available. Adapt the vertmap code to construct maps according to your own specifications.

**Value**

Map of record locations displayed on the selected base map

**Examples**

```
## Not run:
out <- vertsearch("Junco hyemalis") # get occurrence records
vertmap(out)                        # map occurrence records

# Records are color coded by dwc term "scientificname" - sometimes unavailable
out <- vertsearch("mustela nigripes")
vertmap(input = out, mapdatabase = "state")

# Use searchbyterm() to match records with mapped region
spec <- searchbyterm(genus = "ochotona", specific epithet = "princeps", state = "california",
  limit = 200)
vertmap(input = spec, mapdatabase = "state", region = "california")

# Many species
splist <- c("Accipiter erythronemius", "Aix sponsa", "Haliaeetus leucocephalus",
  "Corvus corone", "Threskiornis molucca", "Merops malimbicus")
out <- lapply(splist, function(x) vertsearch(t=x, lim=100))
library("plyr")
out <- ldply(lapply(out, "[", "data"))
vertmap(out)
## jitter points
library("ggplot2")
vertmap(out, geom = geom_jitter, jitter = position_jitter(1, 6))

## End(Not run)
```

---

 vertsearch

---

*Find records using a global full-text search of VertNet archives.*


---

**Description**

Returns any record containing your target text in any field of the record.

**Usage**

```
vertsearch(taxon = NULL, ..., limit = 1000, compact = TRUE,
  verbose = TRUE)
```

**Arguments**

taxon (character) Taxonomic identifier or other text to search for  
 ... (character) Additional search terms

limit	(numeric) Limit on the number of records returned. Up to 1000. We'll incorporate paging later for higher limits. See <a href="#">bigsearch</a> to get larger result sets in a text file via email.
compact	Return a compact data frame (boolean)
verbose	Print progress and information messages. Default: TRUE

## Details

[vertsearch](#) performs a nonspecific search for your input within every record and field of the Vert-Net archives. For a more specific search, try [searchbyterm](#)

## Value

A data frame of search results

## References

<https://github.com/VertNet/webapp/wiki/The-API-search-function>

## Examples

```
## Not run:
out <- vertsearch(taxon = "aves", state = "california", limit=10)

# Limit the number of records returned (under 1000)
out <- vertsearch("(kansas state OR KSU)", limit = 200)
# Use bigsearch() to retrieve >1000 records

# Find multiple species using searchbyterm():
# a) returns a specific result
out <- searchbyterm(genus = "mustela", species = "(nivalis OR erminea)")
vertmap(out)

# b) returns a non-specific result
out <- vertsearch(taxon = "(mustela nivalis OR mustela erminea)")
vertmap(out)

# c) returns a non-specific result
splist <- c("mustela nivalis", "mustela erminea")
out <- lapply(splist, function(x) vertsearch(taxon = x, lim = 500))
library("plyr")
out <- ldply(lapply(out, "[[", "data"))
vertmap(out)

## End(Not run)
```

---

vertsummary	<i>Summarize a set of records downloaded from VertNet.</i>
-------------	--

---

**Description**

Creates a simple summary of data returned by a VertNet search.

**Usage**

```
vertsummary(input, verbose = TRUE)
```

**Arguments**

input	Output from <a href="#">vertsearch</a> , <a href="#">searchbyterm</a> , or <a href="#">spatialsearch</a> . Required.
verbose	Print progress and information messages. Default: TRUE

**Details**

[vertsummary](#) provides information on the sources, types and extent of data returned by a VertNet search.

**Value**

A list of summary statistics

**Examples**

```
## Not run:
recs <- vertsearch("Junco hyemalis") # get occurrence records
vertsummary(recs)                   # summarize occurrence records

vertsummary(vertsearch("Oncorhynchus clarki henshawi"))

## End(Not run)
```

---

vert_id	<i>Search by Vertnet occurrence ID</i>
---------	--

---

**Description**

Search by Vertnet occurrence ID

**Usage**

```
vert_id(ids, compact = TRUE, verbose = TRUE, ...)
```

**Arguments**

ids	(character) VertNet IDs, one or more. Required.
compact	(logical) Return a compact data frame. That is, remove empty columns. Default: TRUE
verbose	(logical) Print progress and information messages. Default: TRUE
...	Curl arguments passed on to <a href="#">GET</a>

**Details**

VertNet IDs can be a variety of things, some URIs (i.e., with `http://...`), while others start with `urn`.

**Value**

A list, with data frame of search results, and list of metadata

**References**

<http://bit.ly/vertnet-wiki>

**Examples**

```
## Not run:
vert_id(ids = "urn:catalog:AUM:Fish:13271")
ids <- c("http://arctos.database.museum/guid/MSB:Mamm:56979?seid=1643089",
        "urn:catalog:CM:Herps:116520",
        "urn:catalog:AUM:Fish:13271")
res <- vert_id(ids)
res$data$occurrenceid
identical(sort(res$data$occurrenceid), sort(ids))

# Lots
out <- vertsearch(taxon = "aves", state = "california", limit = 20)
ids <- out$data$occurrenceid
res <- vert_id(ids)
identical(sort(res$data$occurrenceid), sort(ids))

## End(Not run)
```

# Index

## \*Topic **package**

rvertnet-package, 2

bigsearch, 3, 4, 7, 9, 12

dump, 5

dump\_init (dump), 5

dump\_links (dump), 5

dump\_tbl (dump), 5

filter, 5

fread, 4

GET, 4, 8, 9, 14

rvertnet (rvertnet-package), 2

rvertnet-defunct, 6

rvertnet-package, 2

searchbyterm, 3, 6, 8, 10, 12, 13

select, 5

spatialsearch, 9, 9, 10, 13

tbl, 5

vert\_id, 13

vertavailablemaps, 6

vertlocations, 6

vertmap, 10

vertoccurrence, 6

vertoccurrencecount, 6

vertproviders, 6

vertsearch, 10, 11, 12, 13

vertsummary, 13, 13

verttaxa, 6