Package 'rvertnet'

May 9, 2026

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

Description Retrieve, map and summarize data from the 'VertNet.org' archives (<https://vertnet.org/>). 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 0.8.4

License MIT + file LICENSE

URL https://github.com/ropensci/rvertnet, https://docs.ropensci.org/rvertnet/

BugReports https://github.com/ropensci/rvertnet/issues

Encoding UTF-8

Depends R (>= 2.10)

Imports jsonlite (>= 1.5), crul (>= 0.5.2), dplyr (>= 0.5.0), tibble, ggplot2, maps

Suggests knitr, rmarkdown, testthat, vcr, withr

RoxygenNote 7.3.1

X-schema.org-applicationCategory Data Access

X-schema.org-keywords species, occurrences, biodiversity, maps, vertnet, mammals, mammalia, specimens

X-schema.org-isPartOf ``https://ropensci.org"

VignetteBuilder knitr

LazyData true

NeedsCompilation no

Author Scott Chamberlain [aut, cph] (0000-0003-1444-9135), Chris Ray [ctb], Vijay Barve [ctb], Dave Slager [aut, cre, cph] (0000-0003-2525-2039)

Maintainer Dave Slager <slager@uw.edu>

Repository CRAN

Date/Publication 2024-02-15 23:10:02 UTC

Contents

`rvertnet-package` *Search VertNet archives using R*

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", specificepithet = "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")
```

No results?

It's possible to get no results when requesting data from VertNet, then run the same function again 10 seconds later, and you do get a result. I'm not sure why this is, something having to do with Vertnet's infrastucture that I'm not aware of. Point is, if you are sure you haven't made any mistakes with the parameters, etc., then simply run the function call again.

Author(s)

Maintainer: Dave Slager <slager@uw.edu> (0000-0003-2525-2039) [copyright holder]

Authors:

- Scott Chamberlain <myrmecocystus@gmail.com> (0000-0003-1444-9135) [copyright holder]

Other contributors:

- Chris Ray [contributor]
- Vijay Barve [contributor]

See Also

Useful links:

- https://github.com/ropensci/rvertnet
- https://docs.ropensci.org/rvertnet/
- Report bugs at https://github.com/ropensci/rvertnet/issues

---

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

Description

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

Usage

```
bigsearch(..., rfile, email, messages = TRUE, callopts = list())
```

Arguments

...	arguments, must be named, see searchbyterm() for details
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.
messages	(logical) Print progress and information messages. Default: TRUE
callopts	(named list) Curl arguments passed on to crul::verb-GET

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 `data.table::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", rfile = "pikaRecords", email = "big@search.luv")

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

# Use more than one year query
bigsearch(class = "aves", year = c(">=1976", "<=1986"),
          rfile = "test-bigsearch1", email = "big@search.luv")

## 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 by term*

Description

Flexible search for records using keywords/terms

Usage

```
searchbyterm(
  ...,
  limit = 1000,
  compact = TRUE,
  messages = TRUE,
  only_dwc = TRUE,
  callopts = list()
)
```

Arguments

`...`	arguments, must be named, see section `Parameters` for details. Multiple inputs to a single parameter are supported, but you have to construct that string yourself with `AND` or `OR` operators; see examples below.
`limit`	(numeric) Limit on the number of records returned. If >1000 results, we use a cursor internally, but you should still get up to the results you asked for. See also `bigsearch()` to get larger result sets in a text file via email.
`compact`	(logical) Return a compact data frame
`messages`	(logical) Print progress and information messages. Default: `TRUE`
`only_dwc`	(logical) whether or not to return only Darwin Core term fields. Default: `TRUE`
`callopts`	(named list) Curl arguments passed on to crul::verb-GET

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).

Value

A list with two slots:

- meta: a named list of metadata for the search results
- data: a data frame of search results, columns vary

Parameters

All these parameters can be passed in to `searchbyterm()`. All others will be silently dropped.

See https://github.com/VertNet/webapp/wiki/The-API-search-function for more details

taxon

- kingdom (character) Taxonomic kingdom
- phylum (character) Taxonomic phylum
- class (character) Taxonomic class
- order (character) Taxonomic order
- family (character) Taxonomic family
- genus (character) Taxonomic genus
- specificepithet (character) Taxonomic specific epithet, e.g. (sapiens in Homo sapiens)
- infraspecificepithet (character) Taxonomic infraspecific epithet
- scientificname (character) scientific name
- vernacularname (character) a verncular name

event

- year (numeric) Year 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
- month (numeric) month or range of months designated by comparison operators "<", ">", "<=" or ">=". You can pass in more than one of these queries, in a vector. See example below
- day (numeric) day or range of days designated by comparison operators "<", ">", "<=" or ">=". You can pass in more than one of these queries, in a vector. See example below
- eventdate Event date associated with this occurrence record; yyyy-mm-dd or the range yyyy-mm-dd/yyyy-mm-dd (character)

record level

- institutioncode (character) Code name for the provider/institution of record
- occurrenceid (character) Provider's unique identifier for this occurrence record
- catalognumber (character) Provider's catalog number or other ID for this record
- collectioncode (character) collection code
- license (dcterms:license) license string
- iptlicense (eml:intellectualRights) license string
- basisofrecord (character) one of PreservedSpecimen, FossilSpecimen, MaterialSample, Occurrence, MachineObservation, HumanObservation
- hasmedia (logical) Record also references associated media, such as a film or video
- isfossil (logical) `dwc:basisOfRecord` is FossilSpecimen or collection is a paleo collection
- haslicense (logical) `dcterms:license` or `eml:intellectualRights` has a license designated

identification

- typestatus (character) a type status
- hastypestatus (logical) type status known or not

occurrence

- iptrecordid (character) (same as `dwc:occurrenceID`)
- recordedby (character) Collector name
- recordnumber (character) record number
- fieldnumber (character) field number
- establishmentmeans (character) establishment means
- wascaptive (logical) (`dwc:establishmentMeans` or occurrenceRemarks suggests it was captive)
- wasinvasive (logical) (was the organism recorded to be invasive where and when it occurred)
- sex (character) standardized sex from original sex field or extracted from elsewhere in the record
- lifestage (character) lifeStage from original sex field or extracted from elsewhere in the record
- preparations (not sure what this means)
- hastissue (logical) Record is likely to reference tissues
- reproductivecondition (not sure what this means)

location

- continent (character) Continent to search for occurrence
- country (character) Country to search for occurrence
- stateprovince (character) State or province to search for occurrence
- county (character) County to search for occurrence
- island (character) Island to search for occurrence
- igroup (character) Island group to search for occurrence
- municipality (character)
- waterbody (character)
- geodeticdatum (character)
- georeferencedby (character)
- georeferenceverificationstatus (character)
- location a Google GeoField of the `dwc:decimalLatitude`, `dwc:decimalLongitude`
- mappable (logical) Record includes valid coordinates in decimal latitude and decimal longitude

geological context

- bed (character) geological bed
- formation (character) geological formation
- group (character) geological group

- member (character) geological member

traits

- haslength (logical) (was a value for length extracted?)
- hasmass (logical) (was a value for mass extraccted?)
- hassex (logical) (does the record have sex?)
- haslifestage (logical) (does the record have life stage?)
- lengthtype (character) type of length measurement extracted from the record, can refer to a number or to a range) ('total length', 'standard length', 'snout-vent length','head-body length', 'fork length', 'total length range', 'standard length range', 'snout-vent length range', 'head-body length range', 'fork length range'
- lengthinmm (numeric) length measurement extracted from the record
- massing (numeric) mass measurement extracted from the record (For detailed information about trait extraction and aggregation and querying via the VertNet portal, see http://vertnet.org/resources/traitsguide.htm

data set

- gbifdatasetid (character) GBIF identifier for the data set
- gbifpublisherid (character) GBIF identifier for the data publishing organization
- lastindexed (character) date (YYYY-MM-DD) the record was most recently indexed into VertNet
- networks (character) one of MaNIS, ORNIS, HerpNET, FishNet, VertNet, Arctos, Paleo
- migrator (character) the version of the migrator used to process the data set, a date of form (YYYY-MM-DD)
- orgcountry (character) the country where the organization is located
- orgstateprovince (character) the first-level administrative unit where the organization is located

index

- rank (character) a higher number means the record is more complete with respect to georeferences, scientific names, and event dates
- vntype (character) Type of record; "specimen" or "observation"
- hashid (integer) a value to distribute records in 10k bins; 0-9998

other

- coordinateuncertaintyinmeters (character) Coordinate uncertainty in meters (numeric) or range of uncertainty values designated by comparison operators "<", ">", "<=", or ">="

No results?

It's possible to get no results with a call to `searchbyterm()`, then run it again 10 seconds later, and you do get a result. I'm not sure why this is, something having to do with Vertnet's infrastucture that I'm not aware of. Point is, if you are sure you haven't made any mistakes with the parameters, etc., then simply run the function call again.

References

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

Examples

```
## Not run:
# Find multiple species
out <- searchbyterm(genus = "ochotona",
  specificepithet = "(princeps OR collaris)", limit=10)

# iptrecordid
searchbyterm(iptrecordid = "7108667e-1483-4d04-b204-6a44a73a5219")

# you can pass more than one, as above, in a single string in parens
records <- "(7108667e-1483-4d04-b204-6a44a73a5219 OR 1efe900e-bde2-45e7-9747-2b2c3e5f36c3)"
searchbyterm(iptrecordid = records, callopts = list(verbose = TRUE))

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

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

# Pass in curl options for curl debugging
out <- searchbyterm(class = "aves", limit = 10,
 callopts = list(verbose = TRUE))

# Use more than one year query
searchbyterm(genus = "mustela", specificepithet = "nigripes",
   year = c('>=1900', '<=1940'))

searchbyterm(sex  = "male", limit = 30)$data$sex
searchbyterm(lifestage  = "juvenile", limit = 30)$data$lifestage

## End(Not run)
```

`simple_dwc_terms` *Darwin core terms*

Description

Used internally by `vert_id()` to filter data to Darwin core terms only. Get in touch with us if these terms need correcting or are out of date.

Usage

```
simple_dwc_terms
```

Format

simple_dwc_terms:
A character vector

Source

https://raw.githubusercontent.com/tdwg/dwc/master/dist/simple_dwc_vertical.csv

---

`spatialsearch` *Find records within some distance of a point given latitude and longitude.*

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,
  messages = TRUE,
  ...
)
```

Arguments

`lat`	(numeric) Latitude of the central point, in decimal degrees required.
`long`	(numeric) Longitude of the central point, in decimal degrees required.
`radius`	(numeric) Radius to search, in meters. There is no default value for this parameter. required.
`limit`	(integer) Limit on the number of records returned. If >1000 results, we use a cursor internally, but you should still get up to the results you asked for. See also `bigsearch()` to get larger result sets in a text file via email.
`compact`	(logical) Return a compact data frame. default: `TRUE`
`messages`	(logical) Print progress and information messages. Default: `TRUE`
`...`	Curl arguments passed on to crul::HttpClient

Details

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

Value

A list with two slots:

- meta: a named list of metadata for the search results
- data: a data frame of search results, columns vary

References

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

Examples

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

# Pass in curl options for curl debugging
out <- spatialsearch(lat = 33.529, long = -105.694, radius = 2000,
  limit = 10, verbose = TRUE)

## End(Not run)
```

---

traitsearch *Trait focused search*

Description

Trait focused search

Usage

```
traitsearch(
  taxon = NULL,
  has_mass = FALSE,
  has_length = FALSE,
  has_sex = FALSE,
  has_lifestage = FALSE,
  length_type = NULL,
  length = NULL,
  mass = NULL,
  limit = 1000,
  compact = TRUE,
  messages = TRUE,
```

```
  callopts = list(),
  ...
)
```

Arguments

taxon	(character) Taxonomic identifier or other text to search for
has_mass	(logical) limit to records that have mass data (stored in massing). Default: FALSE
has_length	(logical) limit to records that have length data (stored in lengthinmm). Default: FALSE
has_sex	(logical) limit to records that have sex data (stored in sex). Default: FALSE
has_lifestage	(logical) limit to records that have lifestage data (stored in lifestage). Default: FALSE
length_type	(character) length type, one of 'total length', 'standard length', 'snout-vent length', 'head-body length', 'fork length', 'total length range', 'standard length range', 'snout-vent length range', 'head-body length range', 'fork length range'. (stored in lengthtype) Default: NULL
length	(list) list of query terms for length, e.g., "< 100"
mass	(list) list of query terms for mass, e.g., "< 100"
limit	(numeric) Limit on the number of records returned. If >1000 results, we use a cursor internally, but you should still get up to the results you asked for. See also bigsearch to get larger result sets in a text file via email.
compact	Return a compact data frame (boolean)
messages	Print progress and information messages. Default: TRUE
callopts	curl options in a list passed on to HttpClient, see examples
...	(character) Additional search terms. These must be unnamed

Details

Wraps vertsearch, with some of the same parameters, but with additional parameters added to make querying for traits easy.

Value

a list, same as returned by vertsearch, with data in the data slot

Examples

```
## Not run:
traitsearch(has_mass = TRUE, limit = 3)
traitsearch(has_lifestage = TRUE)
traitsearch(has_mass = TRUE, has_length = TRUE)
res <- traitsearch(length_type = "total length",
  length = list(">= 300", "<= 1000"))
summary(as.numeric(res$data$lengthinmm))
```

```
res <- traitsearch(has_mass = TRUE, mass = list(">= 20", "<= 500"))
summary(as.numeric(res$data$massing))

traitsearch(taxon = "aves", has_mass = TRUE, limit = 100)

## 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 vertsearch, searchbyterm, or spatialsearch. 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 `sort(unique(map_data("world")$region))` after loading packages maps and ggplot2; to see region names for the US "state" layer, run `sort(unique(map_data("state")$region))`
geom	Specifies the type of object being plotted; one of: `geom_point` or `geom_jitter` (do not use quotes)
jitter	If `geom = geom_jitter`, 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 unavailble
out <- vertsearch("mustela nigripes")
vertmap(input = out, mapdatabase = "state")

# Use searchbyterm() to match records with mapped region
spec <- searchbyterm(genus = "ochotona", specificepithet = "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))
out <- dplyr::bind_rows(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,
  messages = TRUE,
  only_dwc = TRUE,
  callopts = list()
)
```

Arguments

taxon	(character) Taxonomic identifier or other text to search for
...	(character) Additional search terms. These must be unnamed
limit	(numeric) Limit on the number of records returned. If >1000 results, we use a cursor internally, but you should still get up to the results you asked for. See also bigsearch to get larger result sets in a text file via email.
compact	Return a compact data frame (boolean)
messages	Print progress and information messages. Default: TRUE
only_dwc	(logical) whether or not to return only Darwin Core term fields. Default: TRUE
callopts	curl options in a list passed on to HttpClient, see examples

Details

vertsearch performs a nonspecific search for your input within every record and field of the VertNet 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", "california", limit=3)

# 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))
out <- dplyr::bind_rows(lapply(out, "[[", "data"))
vertmap(out)

# curl options
```

```
vertsearch(taxon = "Aves", limit = 10, callopts = list(verbose = TRUE))
# vertsearch(taxon = "Aves", limit = 10, callopts = list(timeout_ms = 10))

## End(Not run)
```

---

vertsummary *Summarize a set of records downloaded from VertNet.*

Description

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

Usage

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

Arguments

input	Output from vertsearch, searchbyterm, or spatialsearch. 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:
# get occurrence records
recs <- vertsearch("Junco hyemalis", limit = 10)

# summarize occurrence records
vertsummary(recs)

vertsummary(vertsearch("Oncorhynchus clarki henshawi"))

## End(Not run)
```

vert_id *Search by Vertnet occurrence ID*

Description

Search by Vertnet occurrence ID

Usage

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

Arguments

ids	(character) VertNet IDs, one or more. Required.
compact	(logical) Return a compact data frame. That is, remove empty columns. Default: TRUE
messages	(logical) Print progress and information messages. Default: TRUE
...	Curl arguments passed on to crul::HttpClient

Details

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

Internally in this function we filter data to darwin core terms only. To see what terms we use, see: print(simple_dwc_terms).

See documentation for more information: ?simple_dwc_terms

Value

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

Examples

```
## Not run:
vert_id(ids = "urn:catalog:CM:Herps:116520")
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

out <- vertsearch(taxon = "aves", state = "california", limit = 5)
(ids <- out$data$occurrenceid)
res <- vert_id(ids)
identical(sort(res$data$occurrenceid), sort(ids))

## End(Not run)
```

Index