Package: neotoma (via r-universe)

March 15, 2025

Type Package

Title Access to the Neotoma Paleoecological Database Through R

Version 1.7.7

Date 2022-09-13

Encoding UTF-8

Author Simon J. Goring [aut, cre], Gavin L. Simpson [aut], Jeremiah P. Marsicek [ctb], Karthik Ram [aut], Luke Sosalla [ctb]

Maintainer Simon J. Goring <goring@wisc.edu>

Description NOTE: This package is deprecated. Please use the neotoma2 package described at https://github.com/NeotomaDB/neotoma2. Access paleoecological datasets from the Neotoma Paleoecological Database using the published API (<http://wnapi.neotomadb.org/>), only containing datasets uploaded prior to June 2020. The functions in this package access various pre-built API functions and attempt to return the results from Neotoma in a usable format for researchers and the public.

License MIT + file LICENSE

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

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

Depends R (>= 2.10)

Imports plyr, reshape2, jsonlite, httr, xml2, analogue, leaflet, dplyr, sf, assertthat, methods

Suggests testthat, knitr

RoxygenNote 7.2.1

Config/pak/sysreqs libgdal-dev gdal-bin libgeos-dev make libicu-dev libpng-dev libxml2-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev

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

RemoteUrl https://github.com/ropensci/neotoma

RemoteRef master

RemoteSha fd935c440bb5809502bb3d41c034af0ce7a45ecc

Contents

`ages` *Access proxy age data*

Description

Extracts age information from objects and returns them in a useful format.

Usage

```
ages(obj, ...)

## S3 method for class 'download'
ages(obj, ...)

## S3 method for class 'download_list'
ages(obj, ...)
```

Arguments

`obj`	an R object from which counts are to be extracted.
`...`	arguments passed to other methods.

Details

Methods are available for "download" and "download_list" objects.

Value

Either a data frame of ages or a list of such objects.

Author(s)

Simon Goring

Examples

```
## Not run:
ostracodes <- get_dataset(datasettype = 'ostracode')

ostro.dl <- get_download(ostracodes)
ostro.ages <- ages(ostro.dl)

## End(Not run)
```

`bind` *Function to bind objects together into a longer object.*

Description

From multiple `download*`s, `dataset*`s or `sites`, join them together into a single object.

Usage

```
bind(x, ...)
```

Arguments

`x`	An object returned by one of the `get_*` commands for download, site or dataset.
`...`	other objects of the same class.

Details

To support further synthesis and analysis `compile_download` works to transform a list returned by `get_download` into a large data frame with columns for site and sample attributes and also with the associated assemblage data at each sample depth. This function also does the same for single sites.

Value

This command returns a larger list.

Author(s)

Simon J. Goring <`simon.j.goring@gmail.com`>

References

Neotoma Project Website: http://www.neotomadb.org

Examples

```
## Not run:
#  Search for sites with "Thuja" pollen that are older than 8kyr BP and
#  that are on the west coast of North America:
t8kyr.poa <- get_dataset(taxonname="Thuja*",
                         loc=c(-150, 20, -100, 60), ageyoung = 8000)
t8kyr.canis <- get_dataset(taxonname="Canis*",
                           loc=c(-150, 20, -100, 60), ageyoung = 8000)

t8kyr.co_site <- bind(t8kyr.poa, t8kyr.canis)
plot(t8kyr.co_site)

####
# We want to look at four different dataset types across a forest-prairie
# boundary:
dataset_types <- c("ostracode surface sample",
                   "water chemistry",
                   "diatom surface sample",
                   "pollen surface sample")

# Run the `get_dataset` function for each of the different dataset types
dataset_lists <- lapply(dataset_types,
                          function(x) {
                            get_dataset(datasettype=x,
                                        loc = c(-100,43,-92,48))
                                        })

# Using do.call here to make sure that I don't have to split the list out.
new_datasets <- do.call(bind, dataset_lists)

# And voila!
plot(new_datasets)


## End(Not run)
```

`browse` *Open a browser window to display a Neotoma dataset within the Neotoma Explorer*

Description

Using a `download` or `dataset` object, open up a browser window in the users default browser. Passing a `download_list` or `dataset_list` will open Neotoma Explorer with the first object and return a warning.

Using a numeric value, `download`, `download_list`, `dataset` or `dataset_list` object, open up a browser window in the users default browser. Very large objects

Usage

```
browse(x)
```

Arguments

`x`	A numeric value, `download`, `download_list`, `dataset` or `dataset_list` object.

Value

Returns a NULL value, opens a browser.

Author(s)

Simon J. Goring <`simon.j.goring@gmail.com`>

References

Neotoma Project Website: http://www.neotomadb.org API Reference: http://wnapi.neotomadb.org/doc/resources/sites

Examples

```
## Not run:
# Where are the XRF data?

xrf.data <- get_dataset(datasettype='X-ray fluorescence (XRF)')
browse(xrf.data)


## End(Not run)
```

`browse.dataset` *Open a browser window to display a Neotoma dataset within the Neotoma Explorer*

Description

Using a numeric value, `download`, `download_list`, `dataset` or `dataset_list` object, open up a browser window in the users default browser. Very large objects

Usage

```
## S3 method for class 'dataset'
browse(x)
```

Arguments

`x`	A `dataset` object.

`browse.dataset_list` *Open a browser window to display a Neotoma dataset within the Neotoma Explorer*

Description

Using a numeric value, `download`, `download_list`, `dataset` or `dataset_list` object, open up a browser window in the users default browser. Very large objects

Usage

```
## S3 method for class 'dataset_list'
browse(x)
```

Arguments

`x`	A `dataset_list` object.

`browse.default` *Open a browser window to display a Neotoma dataset within the Neotoma Explorer*

Description

Using a numeric value, `download`, `download_list`, `dataset` or `dataset_list` object, open up a browser window in the users default browser. Very large objects

Usage

```
## Default S3 method:
browse(x)
```

Arguments

`x`	A numeric value with the dataset ID.

`browse.download` *Open a browser window to display a Neotoma dataset within the Neotoma Explorer*

Description

Using a numeric value, `download`, `download_list`, `dataset` or `dataset_list` object, open up a browser window in the users default browser. Very large objects

Usage

```
## S3 method for class 'download'
browse(x)
```

Arguments

`x`	A `download` object.

`browse.download_list` *Open a browser window to display a Neotoma dataset within the Neotoma Explorer*

Description

Using a numeric value, `download`, `download_list`, `dataset` or `dataset_list` object, open up a browser window in the users default browser. Very large objects

Usage

```
## S3 method for class 'download_list'
browse(x)
```

Arguments

`x`	A `download_list` object.

`compile_downloads` *Compile download objects*

Description

Function to convert multiple downloads into a single large table.

From the assemblage data for multiple cores, return a single data.frame with columns for site metadata and assemblage data.

To support further synthesis and analysis `compile_download` works to transform a list returned by `get_download` into a large data frame with columns for site and sample attributes and also with the associated assemblage data at each sample depth. This function also does the same for single sites.

Usage

```
compile_downloads(downloads)
```

Arguments

`downloads`	A download_list as returned by `get_download`, or multiple downloads joined in a list.

Value

This command returns a data frame.

Author(s)

Simon J. Goring <`simon.j.goring@gmail.com`>

References

Neotoma Project Website: http://www.neotomadb.org

Gavin DG, Oswald WW, Wahl ER, Williams JW. 2003. A statistical approach to evaluating distance metrics and analog assignments for pollen records. Quaternary Research 60: 356-367.

Whitmore J, Gajewski K, Sawada M, Williams JW, Shuman B, Bartlein PJ, Minckley T, Viau AE, Webb III T, Shafer S, Anderson P, Brubaker L. 2005. Modern pollen data from North America and Greenland for multi-scale paleoenvironmental applications. Quaternary Science Reviews 24: 1828-1848.

Williams J, Shuman B. 2008. Obtaining accurate and precise environmental reconstructions from the modern analog technique and North American surface pollen dataset. Quaternary Science Reviews. 27:669-687.

API Reference: http://wnapi.neotomadb.org/doc/resources/contacts

Examples

```
## Not run:
#  Search for sites with "Thuja" pollen that are older than 8kyr BP and
#  that are on the west coast of North America:
t8kyr.datasets <- get_dataset(taxonname='Thuja*',
                              loc=c(-150, 20, -100, 60),
                              ageyoung = 8000)

#  Returns 3 records (as of 04/04/2013), get dataset for the first record,
#  Gold Lake Bog.
thuja.sites <- get_download(t8kyr.datasets)

gold.p25 <- compile_taxa(thuja.sites, 'P25')

all.gold <- compile_downloads(gold.p25)

pollen.sums <- rowSums(all.gold[,11:ncol(all.gold)], na.rm=TRUE)

plot(x = all.gold$age,
     y = all.gold$Cupressaceae.Taxaceae / pollen.sums,
     col = all.gold$site.name,
     pch = 19)


## End(Not run)
```

`compile_taxa` *Function to convert assemblage taxa to standardized lists.*

Description

From the assemblage data for the core return assemblage data with the assemblage taxa Currently implemented only for pollen data.

Usage

```
compile_taxa(object, list.name, alt.table = NULL, cf = TRUE, type = TRUE)
```

Arguments

`object`	A pollen object returned by `get_download`.
`list.name`	The taxon compilation list, one of a set of lists from the literature (e.g., `"P25"`, `"WhitmoreFull"`). More detail in section Details.
`alt.table`	A user provided table formatted with at least two columns, one called 'taxon' and the other named as in `list.name`.
`cf`	Should taxa listed as *cf*s (*e.g.*, *cf*. *Gilia*) be considered highly resolved?
`type`	Should taxa listed as types (*e.g.*, *Iva annua*-type) be considered highly resolved?

Details

The data object uses the smaller pollen subset. As this package develops we will add the capacity to summarize data output from the translation. Currently we can return only subsets that have been defined in the literature. These lists include:

- `"P25"` This list is derived from Gavin et al., (2003), and includes 25 pollen taxa.
- `"WS64"` This list is derived from Williams and Shuman (2008).
- `"WhitmoreFull"` This is the full list associated with the Whitmore et al., (2005) North American Modern Pollen Database.
- `"WhitmoreSmall"` As above, but taxa for which both fully resolved and undifferentiated exist these taxa are summed.

Value

This command returns a list object with the same structure as the parent pollen object returned by `get_download`, or a matrix (or data frame) depending on whether `object` is one or the other. Any pollen taxon not included in the major taxa defined in the pollen gets returned as 'Other'.

Author(s)

Simon J. Goring `<simon.j.goring@gmail.com>`

References

Neotoma Project Website: http://www.neotomadb.org

Gavin DG, Oswald WW, Wahl ER, Williams JW. 2003. A statistical approach to evaluating distance metrics and analog assignments for pollen records. Quaternary Research 60: 356-367.

Whitmore J, Gajewski K, Sawada M, Williams JW, Shuman B, Bartlein PJ, Minckley T, Viau AE, Webb III T, Shafer S, Anderson P, Brubaker L. 2005. Modern pollen data from North America and Greenland for multi-scale paleoenvironmental applications. Quaternary Science Reviews 24: 1828-1848.

Williams J, Shuman B. 2008. Obtaining accurate and precise environmental reconstructions from the modern analog technique and North American surface pollen dataset. Quaternary Science Reviews. 27:669-687.

API Reference: http://wnapi.neotomadb.org/doc/resources/contacts

Examples

```
## Not run:
#  Search for sites with "Thuja" pollen that are older than 8kyr BP and
#  that are on the west coast of North America:
t8kyr.datasets <- get_dataset(taxonname='Thuja*', loc=c(-150, 20, -100, 60), ageyoung = 8000)

#  Returns 3 records (as of 04/04/2013), get dataset for the first record, Gold Lake Bog.
GOLDKBG <- get_download(t8kyr.datasets[[1]])

gold.p25 <- compile_taxa(GOLDKBG, 'P25')
```

```
## End(Not run)
```

`counts` *Access proxy count data*

Description

Extract pollen or other proxy counts from data objects and returns them in a useful format.

Usage

```
counts(obj, ...)

## S3 method for class 'download'
counts(obj, ...)

## S3 method for class 'download_list'
counts(obj, ...)
```

Arguments

`obj`	an R object from which counts are to be extracted.
`...`	arguments passed to other methods.

Details

Methods are available for "download" and "download_list" objects.

Value

Either a data frame of counts or a list of such objects.

Author(s)

Gavin Simpson

Examples

```
## Not run:
marion <- get_site('Marion Lake%')
louise <- get_site('Louise Pond%')
western.sites <- rbind(marion, louise)
western.data  <- get_dataset(western.sites)

western.dl <- get_download(western.data)
western.cnt <- counts(western.dl)
sapply(western.cnt, dim)
marion.cnt<- counts(western.dl[[1]])
```

```
dim(marion.cnt)

## End(Not run)
```

`depths` *Extracts the depth values from a ‘download‘ object*

Description

Using a `download` object, return the sample depths (if available).

Using a numeric value, `download`, `download_list`, `dataset` or `dataset_list` object, open up a browser window in the users default browser. Very large objects

Usage

```
depths(obj, ...)

## Default S3 method:
depths(obj, ...)

## S3 method for class 'download'
depths(obj, ...)

## S3 method for class 'download_list'
depths(obj, ...)
```

Arguments

`obj`	A `download` object.
`...`	arguments passed to other methods.

Value

Returns a vector of depths.

Author(s)

Simon J. Goring <`simon.j.goring@gmail.com`>

References

Neotoma Project Website: http://www.neotomadb.org API Reference: http://wnapi.neotomadb.org/doc/resources/sites

Examples

```
## Not run:
# Provide a vector of depths to generate a new age model:
# The dataset id 684 is for Devils Lake, a record published by Louis Maher Jr.

pollen.data <- get_download(684)
pollen.chron <- get_chroncontrol(pollen.data)[[1]]

age_sds <- pollen.chron$chron.control$age - focal$chron.control$age.young,
get_curves <- ifelse(regexpr(”Radiocarbon”,
                             pollen.chron$chron.control$control.type) > -1,
                     'intcal13', 'normal')

new_chron <- Bchron::Bchronology(ages   = pollen.chron$chron.control$age,
                                 ageSds = age_sds
                                 positions = pollen.chron$chron.control$depth,
                                 calCurves = ,
                                 predictPositions = depths(pollen.data))


## End(Not run)
```

`download` *A class for download objects.*

Description

A `download` is an object with the full record for a single dataset.

A `download` is an object with the full record for a single dataset.

Details

TO DO

TO DO

Author(s)

Simon Goring

`get_chroncontrol` *Function to return chronological control tables used to build age models.*

Description

Using the dataset ID, return all records associated with the data. At present, only returns the dataset in an unparsed format, not as a data table. This function will only download one dataset at a time.

Usage

```
get_chroncontrol(x, chronology = 1, verbose = TRUE, add = FALSE)
```

Arguments

`x`	A single numeric chronology ID, a vector of numeric dataset IDs as returned by `get_dataset` or a `download` or `download_list` object.
`chronology`	When `download` objects have more than associated chronology, which chronology do you want? Default is `1`.
`verbose`	logical, should messages on API call be printed?
`add`	logical, should this chron control be added to the download object?

Value

This command returns either an object of class `"try-error"` containing the error returned from the Neotoma API call, or a full data object containing all the relevant information required to build either the default or prior chronology for a core. When `download` or `download_list` objects are passes, the user can `add` the chroncontrol to the `download` object explicitly, in which case the function will return a download with `chroncontrol` embedded.

This is a list comprising the following items:

`chron.control`	A table describing the collection, including dataset information, PI data compatable with `get_contact` and site data compatable with `get_site`.
`meta`	Dataset information for the core, primarily the age-depth model and chronology. In cases where multiple age models exist for a single record the most recent chronology is provided here.

If Neotoma returns empty content, either the control table or the associated metadata (which happens in approximately 25

Author(s)

Simon J. Goring <`simon.j.goring@gmail.com`>

References

+ Neotoma Project Website: http://www.neotomadb.org + API Reference: http://wnapi.neotomadb.org/doc/resources/contacts

Examples

```
## Not run:
#  The point of pulling chronology tables is to re-build or examine the
#  chronological information that was used to build the age-depth model for
#  the core.  You can do this by hand, but the `write_agefile` function works
#  with `download` objects directly.

three_pines <- get_download(get_dataset(get_site("Three Pines Bog"),
                                        datasettype = "pollen"))
pines_chron <- get_chroncontrol(three_pines)

# Spline interpolation:
model <- smooth.spline(x = pines_chron[[1]]$chron.control$depth,
                       y = pines_chron[[1]]$chron.control$age)

new_ages <- predict(model, x = three_pines[[1]]$sample.meta$depth)


## End(Not run)
```

---

get_chroncontrol.dataset

Function to return chronological control tables from a `dataset`.

Description

Using a `dataset`, return the default chron-control table.

Usage

```
## S3 method for class 'dataset'
get_chroncontrol(x, chronology = 1, verbose = TRUE, add = FALSE)
```

Arguments

`x`	A `dataset`.
`chronology`	When `download` objects have more than associated chronology, which chronology do you want? Default is `1`.
`verbose`	logical; should messages on API call be printed?
`add`	Should the `chroncontrol` be added to the download object (only accepts `FALSE`)

`get_chroncontrol.dataset_list` *Function to return chronological control tables from a* `dataset_list`*.*

Description

Using a `dataset_list`, return the default chron-control table.

Usage

```
## S3 method for class 'dataset_list'
get_chroncontrol(x, chronology = 1, verbose = TRUE, add = FALSE)
```

Arguments

`x`	A `dataset_list` object.
`chronology`	When `download` objects have more than associated chronology, which chronology do you want? Default is `1`.
`verbose`	logical; should messages on API call be printed?
`add`	Should the `chroncontrol` be added to the download object (only accepts `FALSE`)

`get_chroncontrol.default` *Function to return chronological control tables from a chronologic ID.*

Description

Using the chronology ID, return the chron control table as a `data.frame`.

Usage

```
## Default S3 method:
get_chroncontrol(x, chronology = 1, verbose = TRUE, add = FALSE)
```

Arguments

`x`	A single numeric chronology ID or a vector of numeric chronology IDs as returned by `get_datasets`.
`chronology`	For `download` methods, which chronology controls should be used?
`verbose`	logical; should messages on API call be printed?
`add`	logical, should this chron control be added to the download object?

`get_chroncontrol.download` *Function to return chronological control tables from a* `download` *object.*

Description

Using a `download`, return the default chron-control table as a `data.frame`.

Usage

```
## S3 method for class 'download'
get_chroncontrol(x, chronology = 1, verbose = TRUE, add = FALSE)
```

Arguments

`x`	A single `download` object.
`chronology`	For `download` methods, which chronology controls should be used?
`verbose`	logical; should messages on API call be printed?
`add`	Should the `chroncontrol` be added to the download object (default `FALSE`)

`get_chroncontrol.download_list` *Function to return chronological control tables from a* `download_list` *object.*

Description

Using a `download_list`, return the default chron-control table as a `data.frame`.

Usage

```
## S3 method for class 'download_list'
get_chroncontrol(x, chronology = 1, verbose = TRUE, add = FALSE)
```

Arguments

`x`	A `download_list` object.
`chronology`	When `download` objects have more than associated chronology, which chronology do you want? Default is `1`.
`verbose`	logical; should messages on API call be printed?
`add`	Should the `chroncontrol` be added to the download object (default `FALSE`)

`get_closest` *Find the closest dataset records to a site, dataset or long/lat pair in Neotoma*

Description

Passing in a download object the function outputs a Bacon or Clam formatted file to a user defined destination for age modelling with existing age-depth modeling software.

Usage

```
get_closest(x, n, buffer, ...)
```

Arguments

`x`	A vector long/lat pair, or a dataset, site or download.
`n`	The maximum number of records to return (in the case of ties the return may be larger)
`buffer`	The size of the buffer for dataset search (in meters)
`...`	optional arguments to pass into `get_dataset`.

Details

The function uses the `sf` package to generate a circular buffer around a point of interest. From there a square bounding box is sent to Neotoma using the `get_dataset()` function. To use the buffering function we must convert from long/lat to UTM coordinates, which we do by guessing the UTM zone of the point of interest. Details can be found in the function's R code hosted on GitHub: `https://github.com/ropensci/neotoma/blob/master/R/get_closest.R`

Value

This command returns a `dataset` or `dataset_list`, or NULL if no records exist within the bounding box.

Author(s)

Simon J. Goring <`simon.j.goring@gmail.com`>, Andria Dawson <`andria.dawson@gmail.com`>

References

Neotoma Project Website: http://www.neotomadb.org API Reference: http://wnapi.neotomadb.org/doc/resources/contacts

Examples

```
## Not run:
#  The point of pulling chronology tables is to re-build or examine the chronological
#  information that was used to build the age-depth model for the core.
# Find the closest records to Madison, WI:
get_closest(x = c(-89.4012, 43.0731), n = 10, buffer = 5000, datasettype = "pollen")

## End(Not run)
```

`get_contact` *Get contact information.*

Description

A function to obtain contact information for data contributors from the Neotoma Paleoecological Database.

Usage

```
get_contact(contactid, contactname, contactstatus, familyname)
```

Arguments

`contactid`	Contact ID is a numerical value associated with the Neotoma Contact table's numerical Contact ID.
`contactname`	A character string indicating the data contributors' project, organization or personal name. May be a partial string and can include wildcards.
`contactstatus`	The current status of the contact. Possible values include: active, deceased, defunct, extant, inactive, retired, unknown.
`familyname`	A character string. Full or partial string indicating the contact's last name.

Value

The function takes parameters defined by the user and returns a list of contact information supplied by the Neotoma Paleoecological Database. The user may define all or none of the possible fields. The function contains data checks for each defined parameter.

The function returns either a single item of class `"try-error"` describing the reason for failure (either mis-defined parameters or an error from the Neotoma API), or a table of contacts, with rows corresponding to the number of individual contacts returned by the Neotoma API. Each row entry includes the following parameters:

`contact.name`	Full name of the person, last name first (e.g. `"Simpson, George Gaylord"`) or name of organization or project (e.g. `"Great Plains Flora Association"`).
`contact.status`	Current status of the person, organization, or project. Field links to the ContactStatuses lookup table.
`family.name`	Family or surname name of a person.

`leading.initials`	Leading initials for given or forenames without spaces (e.g. `"G.G."`).
`given.names`	Given or forenames of a person (e.g. `"George Gaylord"`). Initials with spaces are used if full given names are not known (e.g. `"G. G"`).
`suffix`	Suffix of a person's name (e.g. `"Jr."`, `"III"`).
`title`	A person's title (e.g. `"Dr."`, `"Prof."`, `"Prof. Dr"`).
`phone`	Telephone number.
`fax`	Fax number.
`email`	Email address.
`url`	Universal Resource Locator, an Internet World Wide Web address.
`address`	Full mailing address.
`notes`	Free form notes or comments about the person, organization, or project.
`contact.id`	Unique database record identifier for the contact.
`alias.id`	The ContactID of a person's current name. If the AliasID is different from the ContactID, the ContactID refers to the person's former name.

Author(s)

Simon J. Goring `<simon.j.goring@gmail.com>`

References

Neotoma Project Website: http://www.neotomadb.org API Reference: http://wnapi.neotomadb.org/doc/resources/contacts

Examples

```
## Not run:
#  To find all data contributors who are active:
active.cont <- get_contact(contactstatus = 'active')

# To find all data contributors who have the last name "Smith"
smith.cont <- get_contact(familyname = 'Smith')

## End(Not run)
```

`get_dataset` *Obtain dataset information from the Neotoma Paleoecological Database or an existing object.*

Description

A function to access the Neotoma API and return datasets corresponding to the parameters defined by the user.

Usage

```
get_dataset(
  x,
  datasettype,
  piid,
  altmin,
  altmax,
  loc,
  gpid,
  taxonids,
  taxonname,
  ageold,
  ageyoung,
  ageof,
  subdate
)
```

Arguments

`x`	An optional value, either a `numeric` site ID or object of class `download`, `download_list` or `site`.
`datasettype`	A character string corresponding to one of the allowed dataset types in the Neotoma Database. Allowed types include: `"geochronologic"`, `"loss-on-ignition"`, `"pollen"`, `"plant macrofossils"`, `"vertebrate fauna"`, `"mollusks"`, and `"pollen surface sample"`. See note in Details delow.
`piid`	Numeric value for the Principle Investigator's ID number.
`altmin`	Numeric value indicating the minimum altitude for the site (can be used alone or with `altmax`).
`altmax`	Numeric value indicating the maximum altitude for the site (can be used alone or with `altmin`).
`loc`	A numeric vector `c(lonW, latS, lonE, latN)` representing the bounding box within which to search for sites. The convention here is to use negative values for longitudes west of Greenwich or longitudes south of the equator
`gpid`	A character string or numeric value, must correspond to a valid geopolitical identity in the Neotoma Database. Use get.tables('GeoPoliticalUnits') for a list of acceptable values, or link here: `http://wnapi.neotomadb.org/apdx/geopol.htm`
`taxonids`	A numeric identifier for the taxon. See `get_table` and use `get_table('Taxa')` for a list of acceptable values.
`taxonname`	A character string corresponding to a valid taxon identity in the Neotoma Database. See `get_table` and use `get_table('Taxa')` for a list of acceptable values.
`ageold`	The oldest date acceptable for the search (in years before present).
`ageyoung`	The youngest date acceptable for the search.
`ageof`	If a taxon ID or taxon name is defined this parameter must be set to `"taxon"`, otherwise it may refer to `"sample"`, in which case the age bounds are for any

	samples within datasets or `"dataset"` if you want only datasets that are within the bounds of ageold and ageyoung.
`subdate`	Date of dataset submission, either YYYY-MM-DD or MM-DD-YYYY.

Details

With regards to `datasettypes`, because Neotoma is a "living" database, and new dataset types are being added in an ongoing manner as new research disciplines use the database, you can use `get_table("datasettypes")` to see the full list of available dataset types in the database.

Value

More details on the use of these parameters can be obtained from `http://wnapi.neotomadb.org/doc/resources/datasets`.

A list of class ‘dataset_list‘, with each item corresponding to an individual record. Searches that return no items will result in a NULL value being returned. Otherwise each list item (each dataset record) includes the following components:

`dataset.id`	Unique database record identifier for the dataset.
`dataset.name`	Name of the dataset; not commonly used.
`CollUnitHandle`	Code name of the Collection Unit with which the dataset is associated. This code may be up to 10 characters. Data are frequently distributed by Collection Unit, and the Handle is used for file names.
`CollUnitID`	Unique database record identifier for the collection unit.
`CollType`	The collection type. Types include cores, sections, excavations, and animal middens.
`DatasetType`	The dataset type, such as: geochronologic, loss-on-ignition, pollen, plant macrofossils, vertebrate fauna, etc.
`AgeOldest`	The oldest of all sample ages (in calendar years before present) in the dataset.
`AgeYoungest`	The youngest of all sample ages (in calendar years before present) in the dataset.
`SubDates`	An array of objects that describe dataset submission events. If multiple submissions occurred then this is a table.
`DatasetPIs`	An array of objects that describe Principal Investigators associated with a dataset.
`Site`	An object describing the site where the dataset samples were taken.

Author(s)

Simon J. Goring <`simon.j.goring@gmail.com`>

References

Neotoma Project Website: http://www.neotomadb.org API Reference: http://wnapi.neotomadb.org/doc/resources/contacts

Examples

```
## Not run:
# Search for sites with "Thuja" pollen that are older than 8kyr BP and
# that are on the west coast of North America:
t8kyr.datasets <- get_dataset(taxonname='Thuja*',
                              loc=c(-150, 20, -100, 60),
                              ageyoung = 8000)

# Search for vertebrate fossils in Canada (gpid: 756) within the last 2kyr.
gpids <- get_table(table.name='GeoPoliticalUnits')
canID <- gpids[which(gpids$GeoPoliticalName == 'Canada'),1]

v2kyr.datasets <- get_dataset(datasettype='vertebrate fauna',
                              gpid=canID,
                              ageold = 2000)

## End(Not run)
```

`get_dataset.default` *Obtain dataset information from the Neotoma Paleoecological Database or an existing object.*

Description

A function to access the Neotoma API and return datasets corresponding to the parameters defined by the user.

Usage

```
## Default S3 method:
get_dataset(
  x,
  datasettype,
  piid,
  altmin,
  altmax,
  loc,
  gpid,
  taxonids,
  taxonname,
  ageold,
  ageyoung,
  ageof,
  subdate
)
```

Arguments

`x`	A numeric value corresponding to the site ID.
`datasettype`	A character string corresponding to one of the allowed dataset types in the Neotoma Database. You can find the full list of allowed datasettypes using: `get_table("datasettypes")`.
`piid`	Numeric value for the Principle Investigator's ID number.
`altmin`	Numeric value indicating the minimum altitude for the site (can be used alone or with `altmax`).
`altmax`	Numeric value indicating the maximum altitude for the site (can be used alone or with `altmin`).
`loc`	A numeric vector `c(lonW, latS, lonE, latN)` representing the bounding box within which to search for sites. The convention here is to use negative values for longitudes west of Greenwich or longitudes south of the equator
`gpid`	A character string or numeric value, must correspond to a valid geopolitical identity in the Neotoma Database. Use get.tables('GeoPoliticalUnits') for a list of acceptable values, or link here: `http://wnapi.neotomadb.org/apdx/geopol.htm`
`taxonids`	A numeric identifier for the taxon. See `get_table` and use `get_table('Taxa')` for a list of acceptable values.
`taxonname`	A character string corresponding to a valid taxon identity in the Neotoma Database. See `get_table` and use `get_table('Taxa')` for a list of acceptable values.
`ageold`	The oldest date acceptable for the search (in years before present).
`ageyoung`	The youngest date acceptable for the search.
`ageof`	If a taxon ID or taxon name is defined this parameter must be set to `"taxon"`, otherwise it may refer to `"sample"`, in which case the age bounds are for any samples within datasets or `"dataset"` if you want only datasets that are within the bounds of ageold and ageyoung.
`subdate`	Date of dataset submission, either YYYY-MM-DD or MM-DD-YYYY.

`get_dataset.download` *Obtain dataset information from an existing* `download` *object.*

Description

A function to access a `dataset` within a `download` object.

Usage

```
## S3 method for class 'download'
get_dataset(x, ...)
```

Arguments

`x`	An object of class `download`.
`...`	objects passed from the generic. Not used in the call.

`get_dataset.download_list`

Obtain dataset information from a `download_list`.

Description

A function to return datasets corresponding to the objects within a `download_list`.

Usage

```
## S3 method for class 'download_list'
get_dataset(x, ...)
```

Arguments

`x`	An object of class `download_list`.
`...`	objects passed from the generic. Not used in the call.

`get_dataset.geochronologic`

Obtain dataset information from an object of class `geochronologic`.

Description

A function to access the Neotoma API and return datasets corresponding to the parameters defined by the user.

Usage

```
## S3 method for class 'geochronologic'
get_dataset(x, ...)
```

Arguments

`x`	An object of class `geochronologic`.
`...`	objects passed from the generic. Not used in the call.

`get_dataset.geochronologic_list` *Obtain dataset information from an object of class* `geochronologic_list`*.*

Description

A function to access the Neotoma API and return datasets corresponding to the parameters defined by the user.

Usage

```
## S3 method for class 'geochronologic_list'
get_dataset(x, ...)
```

Arguments

`x`	An object of class `geochronologic_list`.
`...`	objects passed from the generic. Not used in the call.

`get_dataset.integer` *Obtain dataset information from a vector of dataset IDs.*

Description

A function to access the Neotoma API and return datasets corresponding to the parameters defined by the user.

Usage

```
## S3 method for class 'integer'
get_dataset(x = NULL, ...)
```

Arguments

`x`	A single numeric dataset id, or a numeric vector.
`...`	objects passed from the generic. Not used in the call.

`get_dataset.numeric` *Obtain dataset information from a vector of dataset IDs.*

Description

A function to access the Neotoma API and return datasets corresponding to the parameters defined by the user.

Usage

```
## S3 method for class 'numeric'
get_dataset(x = NULL, ...)
```

Arguments

`x`	A single numeric dataset id, or a numeric vector.
`...`	objects passed from the generic. Not used in the call.

`get_dataset.site` *Obtain dataset information from an existing* `site` *object.*

Description

A function to access the Neotoma API and return datasets corresponding to the parameters defined by the user.

Usage

```
## S3 method for class 'site'
get_dataset(x, ...)
```

Arguments

`x`	An object of class `site`.
`...`	objects passed from the generic. Not used in the call.

`get_download` *Function to return full download records using* `sites`*,* `datasets`*, or dataset IDs.*

Description

Using the dataset ID, site object or dataset object, return all records associated with the data as a `download_list`.

Usage

```
get_download(x, verbose = TRUE)
```

Arguments

`x`	A single numeric dataset ID or a vector of numeric dataset IDs as returned by `get_datasets`, or a `site`, `dataset`, or `dataset_list`.
`verbose`	logical; should messages on API call be printed?

Value

This command returns either object of class `"try-error"`' (see `try`) defined by the error returned from the Neotoma API call, or an object of class `download_list`, containing a set of `download` objects, each with relevant assemblage information and metadata: The `download` object is a list of lists and data frames that describe an assemblage, the constituent taxa, the chronology, site and PIs who contributed the data. The following are important components:

`dataset`	A table describing the collection, including dataset information, PI data compatible with `get_contact` and site data compatable with `get_site`.
`sample.meta`	Dataset information for the core, primarily the age-depth model and chronology. In cases where multiple age models exist for a single record the most recent chronology is provided here.
`taxon.list`	The list of taxa contained within the dataset, unordered, including information that can be used in `get_taxa`
`counts`	The assemblage data for the dataset, arranged with each successive depth in rows and the taxa as columns. All taxa are described in `taxon.list`, the chronology is in `sample.data`
`lab.data`	A data frame of laboratory data, such as exotic pollen spike, amount of sample counted, charcoal counts, etc.
`chronologies`	A list of existing chronologies. If only a single chronology exists for a record then this is the same as the age-model in `sample.meta`.

Note

The function returns a warning in cases where single taxa are defined by multiple taphonomic characteristics, for example grains that are identified separately as crumpled and torn in the same sample and sums these values within a sample. In the case that a geochronology dataset is passed to `get_download` the function returns a message and a NULL object (that is later excised). Use `get_geochron` for these objects. The chronologies can be augmented using the function `get_chroncontrol`, where the individual chronology objects in `chronologies` will consist of a table equivalent to `sample.meta` and a `chroncontrol` object.

Author(s)

Simon J. Goring <`simon.j.goring@gmail.com`>

References

Neotoma Project Website: http://www.neotomadb.org API Reference: http://wnapi.neotomadb.org/doc/resources/contacts

Examples

```
## Not run:
#  Search for sites with "Pseudotsuga" pollen that are older than 8kyr BP and
#  that are roughly within western British Columbia:
t8kyr.datasets <- get_dataset(taxonname='*Picea*', loc=c(-90, 41, -89, 44),
                              ageold = 20000, ageyoung=10000)

#  Returns 20 records (as of 04/04/2013), get the dataset for all records:
pollen.records <- get_download(t8kyr.datasets)

#  Standardize the taxonomies for the different records using the WS64 taxonomy.
compiled.sites <- compile_taxa(pollen.records, list.name='WS64')

#  Extract the Pseudotsuga curves for the sites:
get.curve <- function(x, taxa) {
               if (taxa %in% colnames(x$counts)) {
                 count <- x$counts[,taxa]/rowSums(x$counts, na.rm=TRUE)
               } else {
                 count <- rep(0, nrow(x$count))
               }
               data.frame(site = x$dataset$site.data$site.name,
               age = x$sample.meta$age,
               count = count)
             }

curves <- do.call(rbind.data.frame,
                  lapply(compiled.sites, get.curve, taxa = 'Larix/Pseudotsuga'))

#  For illustration, remove the sites with no Pseudotsuga occurance:
curves <- curves[curves$count > 0, ]

smooth.curve <- predict(loess(sqrt(count)~age, data=curves),
                        data.frame(age=seq(20000, 0, by = -100)))
```

```
plot(sqrt(count) ~ age, data = curves,
     ylab = '% Pseudotsuga/Larix', xlab='Calibrated Years BP', pch=19,
     col=rgb(0.1, 0.1, 0.1, 0.1), xlim=c(0, 20000))
lines(seq(20000, 0, by = -100), smooth.curve, lwd=2, lty=2, col=2)

#  This figure shows us an apparent peak in Larix/Pseudotsuga pollen in the
#  early-Holocene that lends support to a warmer, drier early-Holocene in
#  western North America.

## End(Not run)
```

`get_download.dataset` *Function to return full download records using a* `dataset`.

Description

Using a `dataset`, return all records associated with the data as a `download_list`.

Usage

```
## S3 method for class 'dataset'
get_download(x, verbose = TRUE)
```

Arguments

`x`	An object of class `dataset`.
`verbose`	logical; should messages on API call be printed?

`get_download.dataset_list` *Function to return full download records using a* `dataset_list`.

Description

Using a `dataset_list`, return all records associated with the data as a `download_list`.

Usage

```
## S3 method for class 'dataset_list'
get_download(x, verbose = TRUE)
```

Arguments

`x`	An object of class `dataset_list`.
`verbose`	logical; should messages on API call be printed?

`get_download.default` *Function to return full download records using* `numeric` *dataset IDs.*

Description

Using the dataset ID, return all records associated with the data as a `download_list`.

Usage

```
## Default S3 method:
get_download(x, verbose = TRUE)
```

Arguments

`x`	A single numeric dataset ID or a vector of numeric dataset IDs as returned by `get_datasets`.
`verbose`	logical; should messages on API call be printed?

`get_download.site` *Function to return full download records using a* `site`*.*

Description

Using a `site`, return all records associated with the data as a `download_list`.

Usage

```
## S3 method for class 'site'
get_download(x, verbose = TRUE)
```

Arguments

`x`	An object of class `site`.
`verbose`	logical; should messages on API call be printed?

`get_geochron` *Function to return geochronological data from records.*

Description

Using the dataset ID, return all geochronological data associated with the dataID. At present, only returns the dataset in an unparsed format, not as a data table. This function will only download one dataset at a time.

Usage

```
get_geochron(x, verbose = TRUE)
```

Arguments

`x`	A numeric dataset ID or a vector of numeric dataset IDs, or an object of class of class `site`, `dataset`, `dataset_list`, `download` or `download_list` for which geochrons are required.
`verbose`	logical; should messages on API call be printed?

Value

This command returns either an object of class `"try-error"`' (see `try`) defined by the error returned from the Neotoma API call, or a `geochronologic` object, which is a list with two components, a `dataset` and a geochronology table, a `data.frame` with the following components:

`sample.id`	A unique identifier for the geochronological unit.
`age.type`	String. The age type, one of calendar years, radiocarbon years, etc.
`age`	Dated age of the material.
`e.older`	The older error limit of the age value. Commonly 1 standard deviation.
`e.young`	The younger error limit of the age value.
`delta13C`	The measured or assumed delta13C value for radiocarbon dates, if provided.
`material.dated`	A table describing the collection, including dataset information, PI data compatible with `get_contact` and site data compatable with `get_site`.
`geo.chron.type`	Text string, type of geochronological analysis, i.e., Radiocarbon dating, luminesence.
`notes`	Text string
`infinite`	Boolean, does the dated material return an "infinite" date?

A full data object containing all the relevant geochronological data available for a dataset.

Author(s)

Simon J. Goring <`simon.j.goring@gmail.com`>

References

Neotoma Project Website: http://www.neotomadb.org API Reference: http://wnapi.neotomadb.org/doc/resources/contacts

Examples

```
## Not run:
#  Search for the sites around Marion Lake, BC.  I want to find sites within
#  about 1km.

marion <- get_site(sitename = "Marion Lake*")

marion_close <- get_closest(marion, n = 10, buffer = 1)

#  Returns 116 records (as of 13/07/2015).  These are the pollen records though,
#  we want the sites:
geochron.records <- get_geochron(marion_close)

#  We want to extract all the radiocarbon ages from the records:

get_ages <- function(x){
  any.ages <- try(x[[2]]$age[x[[2]]$age.type == 'Radiocarbon years BP'])
  if(is(any.ages, 'try-error')) output <- NA
  if(!is(any.ages, 'try-error')) output <- unlist(any.ages)
  output
}

radio.chron <- unlist(sapply(geochron.records, get_ages))

hist(radio.chron[radio.chron<40000], breaks=seq(0, 25000, by = 1000),
     main = 'Radiocarbon dates for Pseudotsuga records',
     xlab = 'Radiocarbon date (14C years before 1950)')

## End(Not run)
```

`get_publication` *A function to get publications for sites or datasets in the Neotoma Database using the API.*

Description

The function takes the parameters, defined by the user, and returns a table with publication information from the Neotoma Paleoecological Database.

Usage

```
get_publication(x, contactid, datasetid, author, pubtype, year, search)
```

Arguments

`x`	Numeric Publication ID value, either from `get_dataset` or known.
`contactid`	Numeric Contact ID value, either from `get_dataset` or `get_contact`
`datasetid`	Numeric Dataset ID, known or from `get_dataset`
`author`	Character string for full or partial author's name. Can include wildcards such as 'Smit*' for all names beginning with 'Smit'.
`pubtype`	Character string, one of eleven allowable types, see `get_table`. For a list of allowed types run `get_table("PublicationTypes")`.
`year`	Numeric publication year.
`search`	A character string to search for within the article citation.

Value

A list is returned with two data frame components:

`meta`	A single row with Publication ID, type, year of publication and full citation.
`Authors`	`data.frame` of author names, order and IDs, can be of variable length.

Author(s)

Simon J. Goring <`simon.j.goring@gmail.com`>

References

Neotoma Project Website: http://www.neotomadb.org API Reference: http://wnapi.neotomadb.org/doc/resources/contacts

Examples

```
## Not run:
#  To find all publications from 1998:
year.cont <- get_publication(year = 1998)

# To find all data contributors who have the last name "Smith"
smith.cont <- get_publication(author = 'Smith')

## End(Not run)
```

`get_publication.dataset`

A function to get publications for datasets in the Neotoma Database using the API.

Description

The function takes a `dataset` and returns a table with publication information from the Neotoma Paleoecological Database.

Usage

```
## S3 method for class 'dataset'
get_publication(x, ...)
```

Arguments

`x`	an object of class `dataset`.
`...`	objects passed from the generic. Not used in the call.

`get_publication.dataset_list`

A function to get publications for dataset_lists in the Neotoma Database using the API.

Description

The function takes a `dataset_list` and returns a table with publication information from the Neotoma Paleoecological Database.

Usage

```
## S3 method for class 'dataset_list'
get_publication(x, ...)
```

Arguments

`x`	an object of class `dataset_list`.
`...`	objects passed from the generic. Not used in the call.

`get_publication.default`

A function to get publications for sites or datasets in the Neotoma Database using the API.

Description

The function takes the parameters, defined by the user, and returns a table with publication information from the Neotoma Paleoecological Database.

Usage

```
## Default S3 method:
get_publication(x, contactid, datasetid, author, pubtype, year, search)
```

Arguments

`x`	Numeric Publication ID value, either from `get_dataset` or known.
`contactid`	Numeric Contact ID value, either from `get_dataset` or `get_contact`
`datasetid`	Numeric Dataset ID, known or from `get_dataset`
`author`	Character string for full or partial author's name. Can include wildcards such as 'Smit*' for all names beginning with 'Smit'.
`pubtype`	Character string, one of eleven allowable types, see `get_table`. For a list of allowed types run `get_table("PublicationTypes")`.
`year`	Numeric publication year.
`search`	A character string to search for within the article citation.

`get_publication.download`

A function to get publications for downloads in the Neotoma Database using the API.

Description

The function takes a `download` and returns a table with publication information from the Neotoma Paleoecological Database.

Usage

```
## S3 method for class 'download'
get_publication(x, ...)
```

Arguments

`x`	an object of class `download`.
`...`	objects passed from the generic. Not used in the call.

`get_publication.download_list`

A function to get publications for datasets in the Neotoma Database using the API.

Description

The function takes a `download_list` and returns a table with publication information from the Neotoma Paleoecological Database.

Usage

```
## S3 method for class 'download_list'
get_publication(x, ...)
```

Arguments

`x`	an object of class `download_list`.
`...`	objects passed from the generic. Not used in the call.

`get_site` *Return Site Information.*

Description

Return site information from the Neotoma Paleoecological Database.

`get_site` returns site information from the Neotoma Paleoecological Database based on parameters defined by the user.

Usage

```
get_site(sitename, altmin, altmax, loc, gpid, ...)
```

Arguments

`sitename`	character string representing the full or partial site name, or an object of class `dataset`, `dataset_list`, `download` or `download_list`
`altmin`	Minimum site altitude (in m).
`altmax`	Maximum site altitude (in m).
`loc`	A numeric vector c(lonW, latS, lonE, latN) representing the bounding box within which to search for sites. The convention here is to use negative values for longitudes west of Grewnwich or longitudes south of the equator.
`gpid`	A character string or numeric value, must correspond to a valid geopolitical identity in the Neotoma Database. Use get.tables('GeoPoliticalUnits') for a list of acceptable values, or link here: http://wnapi.neotomadb.org/apdx/geopol.htm
`...`	Optional additional arguments

Value

A data frame:

`siteid`	Unique database record identifier for the site.
`sitename`	Name of the site.
`long`	Mean longitude, in decimal degrees, for a site (-180 to 180).
`lat`	Mean latitude, in decimal degrees, for a site (-90 to 90).

`elev`	Elevation in meters.
`description`	Free form description of a site, including such information as physiography and vegetation around the site.
`long_acc`	If the site is described by a bounding box this is the box width.
`lat_acc`	If the site is described by a bounding box this is the box height.

Author(s)

Simon J. Goring <`simon.j.goring@gmail.com`>

References

Neotoma Project Website: http://www.neotomadb.org API Reference: http://wnapi.neotomadb.org/doc/resources/sites

Examples

```
## Not run:
#  What is the distribution of site elevations in Neotoma?
all.sites <- get_site()  #takes a bit of time.

plot(density(all.sites$elev, from = 0, na.rm=TRUE),
main = 'Altitudinal Distribution of Neotoma Sites', xlab = 'Altitude (m)', log='x')

#  Get site information from a dataset:
nw.datasets <- get_dataset(loc = c(-140, 50, -110, 65),
                           datasettype='pollen',
                           taxonname='Pinus*')

nw.sites <- get_site(nw.datasets)


## End(Not run)
```

`get_site.dataset` *Return Site Information from a numeric list of site ids.*

Description

Return site information from the Neotoma Paleoecological Database.

Usage

```
## S3 method for class 'dataset'
get_site(sitename, ...)
```

Arguments

`sitename`	An object of class `dataset`.
`...`	Arguments passed from the generic method, not used.

`get_site.dataset_list` *Return Site Information from a* `dataset_list`

Description

Return site information from the Neotoma Paleoecological Database.

Usage

```
## S3 method for class 'dataset_list'
get_site(sitename, ...)
```

Arguments

`sitename`	An object of class `dataset_list`.
`...`	Arguments passed from the generic method, not used.

`get_site.default` *Return Site Information.*

Description

Return site information from the Neotoma Paleoecological Database.

Usage

```
## Default S3 method:
get_site(sitename, ...)
```

Arguments

`sitename`	A character string representing the full or partial site name.
`...`	Arguments passed from the generic method, not used.

`get_site.download` *Return Site Information from a* `download`

Description

Return site information from the Neotoma Paleoecological Database.

Usage

```
## S3 method for class 'download'
get_site(sitename, ...)
```

Arguments

`sitename`	An object of class `download`.
`...`	Arguments passed from the generic method, not used.

`get_site.download_list` *Return Site Information from a* `download_list`

Description

Return site information from the Neotoma Paleoecological Database.

Usage

```
## S3 method for class 'download_list'
get_site(sitename, ...)
```

Arguments

`sitename`	An object of class `download_list`.
`...`	Arguments passed from the generic method, not used.

`get_site.geochronologic`

Return Site Information from a `geochronologic`

Description

Return site information from the Neotoma Paleoecological Database.

Usage

```
## S3 method for class 'geochronologic'
get_site(sitename, ...)
```

Arguments

`sitename`	An object of class `geochronologic`.
`...`	Arguments passed from the generic method, not used.

`get_site.geochronologic_list`

Return Site Information from a `geochronologic_list`

Description

Return site information from the Neotoma Paleoecological Database.

Usage

```
## S3 method for class 'geochronologic_list'
get_site(sitename, ...)
```

Arguments

`sitename`	An object of class `geochronologic_list`.
`...`	Arguments passed from the generic method, not used.

`get_site.integer` *Return Site Information from a vector of integers.*

Description

Return site information from the Neotoma Paleoecological Database.

Usage

```
## S3 method for class 'integer'
get_site(sitename, ...)
```

Arguments

`sitename`	An integer or vector of integers.
`...`	Arguments passed from the generic method, not used.

`get_site.numeric` *Return Site information from a vector of numeric elements.*

Description

Return site information from the Neotoma Paleoecological Database.

Usage

```
## S3 method for class 'numeric'
get_site(sitename = NULL, ...)
```

Arguments

`sitename`	A numeric value or vector of numeric elements.
`...`	Arguments passed from the generic method, not used.

`get_table` *Get Neotoma value tables.*

Description

Get Neotoma value tables.

Usage

```
get_table(table.name = NULL)
```

Arguments

`table.name`	Call one of the available tables in the Neotoma Database. A full listing of tables can be found here: `http://wnapi.neotomadb.org/doc/resources/dbtables`. By default it returns all objects in the table.

Details

A table of values corresponding to the parameter of interest.

Author(s)

Simon J. Goring <`simon.j.goring@gmail.com`>

References

Neotoma Project Website: http://www.neotomadb.org API Reference: http://wnapi.neotomadb.org/doc/resources/contacts

Examples

```
## Not run:
taxon.table <- get_table('Taxa')

#  Get the frequency of a random taxon in Neotoma.
tax_sample <- sample(nrow(taxon.table), 1)
cat("The taxon",
    taxon.table$TaxonName[tax_sample],
    "occurs in Neotoma",
    length(get_dataset(taxonname = taxon.table$TaxonName[tax_sample])),
    "times.")

## End(Not run)
```

`get_taxa` *Get taxon information from Neotoma.*

Description

Get taxon information from Neotoma.

Usage

```
get_taxa(taxonid, taxonname, status, taxagroup, ecolgroup)
```

Arguments

`taxonid`	Numeric taxon identifier used in Neotoma
`taxonname`	A character string representing the full or partial name of taxa of interest.
`status`	The current status of the taxon, one of 'extinct', 'extant', 'all'.
`taxagroup`	The taxonomic grouping for the taxa. See `http://wnapi.neotomadb.org/doc/resources/taxa` for the list of approved groupings.
`ecolgroup`	The ecological group of the taxa. More detailed than `taxagroup`, can be obtained using `get_table("EcolGroupTypes")`.

Value

Returns a data frame with the following components:

`TaxonID`	Unique database record identifier for a taxon
`TaxonCode`	Shorthand notation for a taxon identification
`TaxonName`	Name of the taxon
`Author`	Author(s) of the name. Used almost exclusively with beetle taxa
`Extinct`	True if extinct; false if extant
`TaxaGroup`	Code for taxa group to which taxon belongs
`EcolGroups`	Array of ecological group codes to which the taxon belongs
`HigherTaxonID`	TaxonID of the next higher taxonomic rank
`PublicationID`	Publication identification number
`Notes`	Free-form notes or comments about the taxon

Author(s)

Simon J. Goring <`simon.j.goring@gmail.com`>

References

Neotoma Project Website: http://www.neotomadb.org API Reference: http://wnapi.neotomadb.org/doc/resources/contacts

Examples

```
## Not run:
## Return all species taxa with "Abies" in name - note wildcard
taxa <- get_taxa(taxonname = "Abies*")

## End(Not run)
```

`gp.table` *A list of all the geopolitical entities in the Neotoma database.*

Description

A list of geopolitical entities with associated numeric ID values.

Usage

```
gp.table
```

Format

a `data.frame` object

Author(s)

Simon J. Goring <`simon.j.goring@gmail.com`>

Source

The Neotoma database.

`plot_leaflet` *Leaflet plots for neotoma data.*

Description

A plotting function to provide interactive data investigation using the leaflet tools. This package requires a connection to the internet for proper functioning.

Usage

```
plot_leaflet(x, providerTiles = "Stamen.TerrainBackground", ...)
```

Arguments

`x`	A neotoma data object
`providerTiles`	Default "Stamen.TerrainBackground", a character string indicating the tile background to be used for plotting.
`...`	Other terms to be passed to the function.

Value

A `leaflet` object

`pollen.equiv` *A table to convert the pollen taxa identified by investigators to standardized lists.*

Description

A list of standardized (published) taxonomies from the literature to help standardize taxonomies for synthesis work.

Usage

```
data(pollen.equiv)
```

Format

a `data.frame` object

Details

Taxon conversion table (readable).

Author(s)

Simon J. Goring <`simon.j.goring@gmail.com`>; Jeremiah Marsicek

`read.tilia` *Read proxy data from Tilia TLX files*

Description

Read proxy data from a Tilia TLX format file.

Usage

```
read.tilia(file)
```

Arguments

`file`	a string representing a Tilia TLX format file.

Value

Return a ‘download‘ object.

Author(s)

Simon J. Goring <`simon.j.goring@gmail.com`>

Examples

```
## Not run:
  crystal <- read.tilia('crystal.tlx')

## End(Not run)
```

`read_bacon` *Function to read in defined Bacon outputs.*

Description

Reads in Bacon output and formats it for inclusion in a download object.

Usage

```
read_bacon(
  x,
  path = ".",
  add = FALSE,
  chron_name = "Bacon",
  as_default = TRUE,
  download = NULL,
  sections = NULL,
  age_field = "median",
  interp = TRUE
)
```

Arguments

`x`	A folder path that contains a Bacon age file.
`path`	The location of the `Cores` folder.
`add`	Should the results be added to an existing `download`? Defaults to `FALSE`.
`chron_name`	The name for the chronology if the Bacon file is being added to a `download`.
`as_default`	Should the chronology become the default?
`download`	The target `download` if add is `TRUE`.
`sections`	If there are multiple Bacon runs in a folder, identify the file by the number of sections in the run.
`age_field`	Should the age be assigned to the `"median"` or the `"wmean"`?
`interp`	If the depths don't match up, should we interpolate from the Bacon output? (default `TRUE`)

Details

The function expects that you are in a working directory containing a "Cores" which would then contain output files from Bacon runs. The output can either be added to an existing record (for example, replacing the default age model returned by Neotoma), or it can be loaded on its own. If the depths for the loaded file do not match with the depths in the 'download''s 'sample.meta' then the user can use the 'interp' parameter to interpolate between depths. This method uses linear interpolation.

Examples

```
## Not run:
# Download the record for Lake O' Pines:
lake_o_dl <- get_download(15925)

# This assumes that you have Bacon installed in a folder and have
# set it to your working directory.

write_agefile(lake_o_dl[[1]], path = ".", chronology = 1,
              corename = "LAKEPINES", cal.prog = 'Bacon')
```

```
source("Bacon.R")

# These defaults just help the core run quickly, they're not
# neccesarily good parameters.

Bacon("LAKEPINES", acc.mean = 10,
      thick = 50, depths.file = TRUE,
      suggest = FALSE, ask = FALSE)

lake_o_dl <- read_bacon("LAKEPINES", add = TRUE,
                        download = download, sections = 17)


## End(Not run)
```

`Stratiplot.download` *Palaeoecological stratigraphic diagrams*

Description

Draws paleoecological diagrams from a `download` object. Allows control of variable type (using the `tran` function from the `analogue` package), and taxonomic grouping.

Usage

```
## S3 method for class 'download'
Stratiplot(x, yaxis = "age", method = "none", group = NULL, ...)
```

Arguments

`x`	A `download` object.
`yaxis`	One of the columns in `sample.meta`, including `depth`, `age`, `age.younger`, or `age.older`, default `age`.
`method`	An option for axis transformation using `tran` from the `analogue` package. `"none"` by default.
`group`	An ecological group from the taxon table.
`...`	variables to be passed to `Stratiplot`.

Details

A wrapper for the `analogue` package's `Stratiplot` function. Allowing the user to plot a stratigraphic diagram directly from a `download` object.

Value

A `trellis` object.

Examples

```
## Not run:
lake_o_dl <- get_download(15925)
Stratiplot(lake_o_dl[[1]])

## End(Not run)
```

`Stratiplot.download_list`

Palaeoecological stratigraphic diagrams

Description

Draws paleoecological diagrams from a `download_list` object. Allows control of variable type (using the `tran` function from the `analogue` package), and taxonomic grouping. This function only works for `download_list` objects that contain a single object.

Usage

```
## S3 method for class 'download_list'
Stratiplot(x, yaxis = "age", method = "none", group = NULL, ...)
```

Arguments

`x`	A `download_list` object.
`yaxis`	One of the columns in `sample.meta`, including `depth`, `age`, `age.younger`, or `age.older`, default `age`.
`method`	An option for axis transformation using `tran` from the `analogue` package. `"none"` by default.
`group`	An ecological group from the taxon table.
`...`	variables to be passed to `Stratiplot`.

Details

A wrapper for the `analogue` package's `Stratiplot` function. Allowing the user to plot a stratigraphic diagram directly from a `download` object.

Value

A `trellis` object.

Examples

```
## Not run:
lake_o_dl <- get_download(15925)
# This works:
Stratiplot(lake_o_dl)

lakes_o_nw <- get_download(get_site(sitename = "Lake B%"))
# This Fails:
# Stratiplot(lake_o_nw)


## End(Not run)
```

`taxa` *Access proxy taxonomic data*

Description

Extracts taxa from `download` objects and returns them in a useful format.

Usage

```
taxa(obj, ...)

## S3 method for class 'download'
taxa(obj, ...)

## S3 method for class 'download_list'
taxa(obj, collapse = TRUE, hierarchy = FALSE, ...)
```

Arguments

`obj`	an R object from which counts are to be extracted.
`...`	arguments passed to other methods.
`collapse`	should the results be returned as a list, one for each site (`FALSE`), or a single dataframe of all taxa? Default is `TRUE`
`hierarchy`	Should the taxonomic hierarchy be included?

Details

Methods are available for "download" and "download_list" objects.

Value

Either a data frame of taxa or a list of such objects.

Author(s)

Simon Goring

Examples

```
## Not run:
ostracodes <- get_dataset(datasettype = 'ostracode')

ostro.dl <- get_download(ostracodes)
ostro.taxa <- taxa(ostro.dl)

## End(Not run)
```

`taxon.list` *Neotoma taxon list*

Description

The taxonomy table for datasets in neotoma, as would be returned by `get_table`

Usage

```
data(taxon.list)
```

Format

a `data.frame` object

Author(s)

Simon J. Goring <`simon.j.goring@gmail.com`>

Source

The Neotoma database.

`write_agefile` *Write age control file to disk formatted for either Bacon or Clam*

Description

Passing in a download object the function outputs a Bacon or Clam formatted file to a user defined destination for age modelling with existing age-depth modeling software.

Usage

```
write_agefile(download, chronology = 1, path, corename, cal.prog = "Bacon")
```

Arguments

`download`	A single site returned by `get_download`.
`chronology`	Default is `1`, the default chronology for the core. If a core has more than one chronology the user can define a different set of chronological controls.
`path`	The location of the 'Cores' folder & working directory for Bacon. Do not include "Cores" in the path name.
`corename`	The intended handle for the core, to be used in writing to file.
`cal.prog`	The method intended to build the age model, either `'Bacon'` or `'Clam'`.

Value

This command returns a file in location `path/Cores` containing all the relevant information required to build either the default or prior chronology for a core.

Author(s)

Simon J. Goring <`simon.j.goring@gmail.com`>

References

Neotoma Project Website: http://www.neotomadb.org API Reference: http://wnapi.neotomadb.org/doc/resources/contacts

Examples

```
## Not run:
# Find a particular record:

three_pines <- get_download(get_dataset(get_site("Three Pines Bog"),
                                        datasettype = "pollen"))

# You will need to edit the `path` argument here to point to a directory that
# contains a `Cores` directory.

write_agefile(download = three_pines[[1]],
```

```
               path = "./inst",
               corename = "THREEPINES",
               cal.prog = "Bacon")

## End(Not run)
```

Index