Package 'rmangal'

October 3, 2019

Type Package

Title 'Mangal' Client

Version 2.0.0

Description
An interface to the 'Mangal' database - a collection of ecological networks. This package includes functions to work with the 'Mangal RESTful API' methods (<https://mangal.io/doc/api>).

URL <https://mangal.io>, <https://github.com/ropensci/rmangal>

BugReports <https://github.com/ropensci/rmangal/issues>

License MIT + file LICENSE

Encoding UTF-8

Imports httr (>= 1.3.1), igraph, jsonlite (>= 1.5), memoise, purrr

RoxygenNote 6.1.1

Suggests ggraph (>= 2.0.0), knitr, magrittr, mapview, rmarkdown, sf,
spelling, taxize, tidygraph, testthat, tibble, USAboundaries,

VignetteBuilder knitr

NeedsCompilation no

Author Steve Vissault [aut, cre] (<https://orcid.org/0000-0002-0866-4376>),
Kevin Cazelles [aut, ctb] (<https://orcid.org/0000-0001-6619-9874>),
Gabriel Bergeron [aut, ctb],
Benjamin Mercier [aut, ctb],
Clément Violet [aut, ctb],
Dominique Gravel [aut],
Timothée Poisot [aut],
Thomas Lin Pedersen [rev] (<https://orcid.org/0000-0002-5147-4711>),
Anna Willoughby [rev] (<https://orcid.org/0000-0002-0504-0605>)

Maintainer Steve Vissault <steve.vissault@usherbrooke.ca>

Repository CRAN

R **topics documented:**

---

as.igraph.mgNetwork   *Coerce* mgNetworksCollection *or* mgNetwork *objects to* igraph *objects.*

Description

Coerce mgNetworksCollection or mgNetwork objects to igraph objects.

Usage

```
## S3 method for class 'mgNetwork'
as.igraph(x, ...)

## S3 method for class 'mgNetworksCollection'
as.igraph(x, ...)
```

Arguments

x	either a mgNetworksCollection or a mgNetwork object.
...	currently ignored.

Value

An object of class igraph for a mgNetwork object and a list of igraph objects for mgNetworksCollection.

Methods (by class)

- mgNetwork: Convert mgNetwork objects to igraph objects.
- mgNetworksCollection: Convert mgNetworksCollection objects to a list of igraph objects.

---

avail_type	*List interactions type contains in mangal-db*

Description

List interactions type contains in mangal-db

Usage

```
avail_type()
```

---

clear_cache_rmangal	*Clear memoise cache*

Description

Clear memoise cache

Usage

```
clear_cache_rmangal()
```

Details

Clear memoise cache generated by mem_get(), a memoise function around httr::GET() used in get_gen in get_singleton

---

combine_mgNetworks	*Combine Mangal networks*

Description

Combine mgNetworksCollection and mgNetwork objects into a mgNetworksCollection object.

Usage

```
combine_mgNetworks(...)
```

Arguments

...	objects of class mgNetworksCollection or mgNetwork or a list #' of objects of these classes.

Value

An object of class `mgNetworksCollection`

Examples

```
mg_random_1071 <- get_collection(c(1071), verbose = FALSE)
mg_random_1074 <- get_collection(c(1074), verbose = FALSE)
combine_mgNetworks(mg_random_1071, mg_random_1074)
```

---

get_citation	*Retrieve all references pertaining to the networks collection or individual network*

Description

Retrieve all references pertaining to the networks collection or individual network

Usage

```
get_citation(x)

## S3 method for class 'mgNetwork'
get_citation(x)

## S3 method for class 'mgNetworksCollection'
get_citation(x)
```

Arguments

x   an object of class `mgNetworksCollection` or `mgNetworks`.

Value

Bibtex entries as a character vector.

Methods (by class)

- `mgNetwork`: Get BibTeX entries for the publication associated to the network.
- `mgNetworksCollection`: Get BibTeX entries for the publication associated to the networks.

Examples

```
# network collection
lagoon_net_collection <- get_collection(search_datasets("lagoon"))
get_citation(lagoon_net_collection)
# individual network
mg_18 <- get_network_by_id(18)
get_citation(mg_18)
```

---

get_collection *Get a collection of networks*

Description

Retrieve a set of networks based on the results of one of the search_*() function. The function also accept a numeric vector of Mangal network IDs.

Usage

```
get_collection(x, ...)

## Default S3 method:
get_collection(x, ...)

## S3 method for class 'mgSearchDatasets'
get_collection(x, ...)

## S3 method for class 'mgSearchNetworks'
get_collection(x, ...)

## S3 method for class 'mgSearchReferences'
get_collection(x, ...)

## S3 method for class 'mgSearchNodes'
get_collection(x, ...)

## S3 method for class 'mgSearchTaxonomy'
get_collection(x, ...)

## S3 method for class 'mgSearchInteractions'
get_collection(x, ...)
```

Arguments

x	numeric vector of Mangal network IDs or an object returned by by one of the search_*() functions.
...	arguments to be passed on to rmangal::get_network_by_id().

Value

If there is only one network to be retrieved, get_collection() returns a mgNetwork object, otherwise it returns a object of class mgNetworksCollection which is a collection (a list) of mgNetwork objects rmangal::get_network_by_id()).

Methods (by class)

- `default`: Get a collection of networks (default).
- `mgSearchDatasets`: Get a collection of networks from a `mgSearchDatasets` object.
- `mgSearchNetworks`: Get a collection of networks from a `mgSearchNetworks` object.
- `mgSearchReferences`: Get a collection of networks from a `mgSearchReferences` object.
- `mgSearchNodes`: Get a collection of networks from a `mgSearchNodes` object.
- `mgSearchTaxonomy`: Get a collection of networks from a `mgSearchTaxa` object.
- `mgSearchInteractions`: Get a collection of networks from a `mgSearchTaxa` object.

See Also

search_datasets(), search_interactions(), search_networks(), search_nodes(), search_references(), search_taxonomy().

Examples

```
mg_2 <- get_collection(c(1076:1077), verbose = FALSE)
mg_anemone <- get_collection(search_networks(query='anemone%'), verbose = FALSE)
```

---

get_network_by_id	*Retrieve network informations, nodes, edges and references for a given set of Mangal network IDs*

Description

Summarize mgNetwork properties.

Summarize mgNetworksCollection properties.

Usage

```
get_network_by_id(ids, as_sf = FALSE, verbose = TRUE)

get_network_by_id_indiv(id, as_sf = FALSE, verbose = TRUE)

## S3 method for class 'mgNetwork'
print(x, ...)

## S3 method for class 'mgNetworksCollection'
print(x, ...)

## S3 method for class 'mgNetwork'
summary(object, ...)

## S3 method for class 'mgNetworksCollection'
summary(object, ...)
```

Arguments

ids	a vector of Mangal ID for networks (`numeric`).
as_sf	a logical. Should networks metadata be converted into an sf object? Note that to use this feature `sf` must be installed.
verbose	a logical. Should extra information be reported on progress?
id	a single ID network (`numeric`).
x	an object of class `mgNetwork` or `mgNetworksCollection`.
...	ignored.
object	object of of class `mgNetwork` or `mgNetworksCollection`.

Value

A `mgNetwork` object includes five data frame:

- network: includes all generic information on the network (if `as_sf=TRUE` then it is an object of class `sf`);

- nodes: information pertaining to nodes (includes taxonomic information);

- interactions: includes ecological interactions and their attributes;

- dataset: information pertaining to the original dataset;

- reference: details about the original publication.

A summary method is available for objects of class `mgNetwork` object and returns the following network properties:

- the number of nodes;

- the number of edges;

- the connectance;

- the linkage density;

- the degree (in, out an total) and the eigenvector centrality of every nodes.

Functions

- `get_network_by_id_indiv`: Retrieve a network by its collection of networks (default).

Examples

```
net18 <- get_network_by_id(id = 18)
nets <- get_network_by_id(id = c(18, 23))
```

search_datasets   *Query datasets*

Description

Identify relevant datasets using a keyword or a custom query. If the query is a character string, then all character columns in the table are searched and the entries for which at least one partial match was found are returned. Alternatively, a named list can be used to look for an exact match in a specific column (see Details section)

Usage

```
search_datasets(query, verbose = TRUE, ...)
```

Arguments

query	either a character string including a single keyword or a list containing a custom query (see details section below). Note that if an empty character string is passed, then all datasets available are returned.
verbose	a logical. Should extra information be reported on progress?
...	further arguments to be passed to `httr::GET()`.

Details

Names of the list should match one of the column names within the table. For the `dataset` table, those are:

- id: unique identifier of the dataset
- name: name of the dataset
- date: date (`YYYY-mm-dd`) of the corresponding publication
- description: a brief description of the data set
- ref_id: the Mangal identifier of the dataset

Note that for lists with more than one element, only the first element is used, the others are ignored. Examples covering custom queries are provided below.

Value

An object of class `mgSearchDatasets`, which basically is a `data.frame` including all datasets corresponding to the query. For each dataset entry, the networks and the original reference are attached.

References

Metadata available at https://mangal-wg.github.io/mangal-api/#datasets

Examples

```
# Return all datasets (takes time)
all_datasets <- search_datasets("", verbose = FALSE)
all_datasets
class(all_datasets)

# Search with keyword
mg_lagoon <- search_datasets(query = 'lagoon', verbose = FALSE)
# Search with a custom query (specific column)
mg_kemp <- search_datasets(query = list(name = 'kemp_1977'), verbose = FALSE)
mg_16 <- search_datasets(query = list(ref_id = 16), verbose = FALSE)
```

---

search_interactions   *Query interactions*

Description

Search for specific interactions using a keyword or a specific type of interactions (e.g. mutualism). If the query is a character string, then all character columns in the table are searched and the entries for which at least one partial match was found are returned. Alternatively, a named list can be used to look for an exact match in a specific column (see Details section)

Usage

```
search_interactions(query, type = NULL, expand_node = FALSE,
  verbose = TRUE, ...)
```

Arguments

query	either a character string including a single keyword or a list containing a custom query (see details section below). Note that if an empty character string is passed, then all datasets available are returned.
type	a character one of the interactions type available, uses avail_type() to see the full list of type available. Note that query is ignored if type is used.
expand_node	a logical. Should the function returned extra information pertaining to nodes? Default is set to FALSE, which means that only the Mangal IDs of nodes are returned.
verbose	a logical. Should extra information be reported on progress?
...	further arguments to be passed to httr::GET().

Details

Names of the list should match one of the column names within the table. For the interaction table, those are:

- id: unique identifier of the interaction

- attr_id: identifier of a specific attribute

- direction: edge direction ("directed", "undirected" or "unknown")

- network_id: Mangal network identifier

- node_from: node id which the interaction end to

- node_to: node to which the interaction end to

- type: use argument `type` instead

Note that for lists with more than one element, only the first element is used, the others are ignored. Examples covering custom queries are provided below.

Value

An object of class `mgSearchInteractions`, i.e. a `data.frame` object including interactions. All networks in which interactions are involved are also attached to the `data.frame`.

References

Metadata available at <https://mangal-wg.github.io/mangal-api/#interactions>

Examples

```
df_inter <- search_interactions(type = "competition", verbose = FALSE)
# Get all networks containing competition

competition_networks <- get_collection(df_inter, verbose = FALSE)

df_net_926 <- search_interactions(list(network_id = 926), verbose = FALSE)
```

---

search_networks   *Query networks*

Description

Search over all networks using a keyword, a custom query or a spatial object If the query is a character string, then all character columns in the table are searched and the entries for which at least one partial match was found are returned. Alternatively, a named list can be used to look for an exact match in a specific column (see Details section)

Usage

```
search_networks(query, verbose = TRUE, ...)

search_networks_sf(query_sf, verbose = TRUE, ...)
```

Arguments

query	either a character string including a single keyword or a list containing a custom query (see details section below), or a spatial object (see the description of query_sf). Note that if an empty character string is passed, then all datasets available are returned.
verbose	a `logical`. Should extra information be reported on progress?
...	further arguments to be passed to `httr::GET()`.
query_sf	a spatial object of class `sf` used to search in a specific geographical area.

Details

Names of the list should match one of the column names within the table. For the `networks` table, those are

- id: unique identifier of the network
- all_interactions: false interaction can be considered as real false interaction
- dataset_id: the identifier of the dataset
- public: network publicly available

Note that for lists with more than one element, only the first element is used, the others are ignored. An example is provided below.

Value

An object of class `mgSearchNetworks`, which is a `data.frame` object with all networks informations

Functions

- search_networks_sf: Search networks within a spatial object passed as an argument. Note that `sf` must be installed to use this function.

References

Metadata available at <https://mangal-wg.github.io/mangal-api/#networks>

Examples

```
mg_insect <- search_networks(query="insect%")

# Retrieve the search results
nets_insect <- get_collection(mg_insect)
# Spatial query
library(USAboundaries)
area <- us_states(state="california")
networks_in_area <- search_networks(area, verbose = FALSE)
plot(networks_in_area)

# Retrieve network ID 5013
```

```
net_5013 <- search_networks(query = list(id = 5013))
# Network(s) of dataset ID 19
mg_19 <- search_networks(list(dataset_id = 19))
```

---

search_nodes   *Query nodes*

Description

Search for networks by querying the nodes table. If the query is a character string, then all character columns in the table are searched and the entries for which at least one partial match was found are returned. Alternatively, a named list can be used to look for an exact match in a specific column (see Details section)

Usage

```
search_nodes(query, verbose = TRUE, ...)
```

Arguments

query   either a character string including a single keyword or a list containing a custom query (see details section below). Note that if an empty character string is passed, then all datasets available are returned.

verbose   a logical. Should extra information be reported on progress?

...   further arguments to be passed to httr::GET().

Details

Names of the list should match one of the column names within the table. For the networks table, those are:

- id: unique identifier of the nodes
- original_name: taxonomic name as in the original publication
- node_level: either population, taxon or individu
- network_id: Mangal network identifier

Note that for lists with more than one element, only the first element is used, the others are ignored. An example is provided below.

Value

An object of class mgSearchNodes, which is a data.frame including taxa #' matching the query and corresponding information. All networks in which taxa are involved are also attached to the data.frame.

References

Metadata available at https://mangal-wg.github.io/mangal-api/#nodes

See Also

search_taxonomy()

Examples

```
res_acer <- search_nodes("Acer")
res_926 <- search_nodes(list(network_id = 926))
```

---

search_references   *Query references*

Description

Search for a specific reference using a key word or a Digital Object Identifier (DOI). If the `query` is a character string, then all character columns in the table are searched and the entries for which at least one partial match was found are returned. Alternatively, a named list can be used to look for an exact match in a specific column (see Details section)

Usage

```
search_references(query, doi = NULL, verbose = TRUE, ...)
```

Arguments

query	either a character string including a single keyword or a list containing a custom query (see details section below). Note that if an empty character string is passed, then all datasets available are returned.
doi	`character` a Digital Object Identifier (DOI) of the article. Note that `query` is ignored if `doi` is specified.
verbose	a `logical`. Should extra information be reported on progress?
...	further arguments to be passed to `httr::GET()`.

Details

Names of the list should match one of the column names within the table. For the `reference` table, those are:

- id: unique identifier of the reference
- first_author: first author
- doi: use `doi` instead
- jstor: JSTOR identifier

- year: year of publication.

Note that for lists with more than one element, only the first element is used, the others are ignored. An example is provided below.

Value

An object of class `mgSearchReferences`, which is a list that includes a wide range of details associated to the reference, including all datasets and networks related to the publication that are included in Mangal database.

References

Metadata available at <https://mangal-wg.github.io/mangal-api/#references>

Examples

```
search_references(doi = "10.2307/3225248")
search_references(list(jstor = 3683041))
search_references(list(year = 2010))
```

---

search_taxonomy   *Query taxonomy*

Description

Search network by taxon names and unique taxonomic identifiers. This function offers the opportunity to retrieve taxon based on (i) known identifier such as the taxonomic serial number (TSN), GBIF ID etc. or (ii) text search using partial match. Have a look at the list of arguments to see the complete list of identifiers accessible. If any unique identifier argument is used (i.e. tsn etc.), then query is ignored. Moreover, if several taxonomic identifiers are specified, then only the first one is considered.

Usage

```
search_taxonomy(query, tsn = NULL, gbif = NULL, eol = NULL,
  col = NULL, bold = NULL, ncbi = NULL, verbose = TRUE, ...)
```

Arguments

query	a character string including a single keyword. Note that if an empty character string is passed, then all datasets available are returned.
tsn	a numeric. Unique taxonomic identifier from Integrated Taxonomic Information System (<https://www.itis.gov/>).
gbif	a numeric. Unique taxonomic identifier from Global Biodiversity Information Facility (<https://www.gbif.org/>).
eol	a numeric. Unique taxonomic identifier from Encyclopedia of Life (<https://eol.org/>).

col	a numeric. Unique taxonomic identifier from Catalogue of Life (https://www.catalogueoflife.org/).
bold	a numeric. Unique taxonomic identifier from Barcode of Life (http://www.boldsystems.org/).
ncbi	a numeric. Unique taxonomic identifier from National Center for Biotechnology Information (https://www.ncbi.nlm.nih.gov/).
verbose	a logical. Should extra information be reported on progress?
...	further arguments to be passed to httr::GET().

Details

Taxon names of the taxonomy table were validated with TNRS (see http://tnrs.iplantcollaborative.org/ and/or GNR might not be the taxon name documented in the original publication. In order to identify relevant networks with the original name, use search_nodes().

The validation of taxon names was performed by an automated procedure and if there is any doubt, the original names recorded by authors should be regarded as the most reliable information. Please report any issue related to taxonomy at https://github.com/mangal-wg/mangal-datasets/issues.

Value

An object of class mgSearchTaxonomy, which is a data.frame including all taxa matching the query.

References

Metadata available at https://mangal-wg.github.io/mangal-api/#taxonomy

See Also

search_nodes()

Examples

```
search_taxonomy("Acer")
# Retrieve higher classification
tsn_acer <- search_taxonomy("Acer")$taxonomy.tsn
```

Index