Package: rmangal (via r-universe)

October 29, 2026

Type Package

Title 'Mangal' Client

Version 2.2.2

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-interactions.github.io/mangal-api/>).

URL https://docs.ropensci.org/rmangal/, https://mangal.io, https://github.com/ropensci/rmangal

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

License MIT + file LICENSE

Encoding UTF-8

Language en-US

LazyData true

Depends R (>= 4.1)

Imports cli, httr2 (>= 1.2.0), igraph (>= 2.0.0), jsonlite (>= 1.5), methods, rlang

RoxygenNote 7.3.3

Suggests ggraph (>= 2.0.0), knitr, mapview, rmarkdown, sf, spelling, tidygraph, testthat (>= 3.0.0), tibble, vcr, withr

VignetteBuilder knitr

Roxygen list(markdown = TRUE)

X-schema.org-applicationCategory ecologicalNetworks

X-schema.org-keywords ecology, networks, food webs, interactions, data publications, open access

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

Config/testthat/edition 3

Config/pak/sysreqs libglpk-dev libxml2-dev libssl-dev

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

Date/Publication 2026-01-22 14:39:14 UTC

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

RemoteRef HEAD

RemoteSha 9c566bad9992c02dfcb32bc2cfaaf665f86e7a5f

Contents

as.igraph.mgNetwork . . . 2
avail_type . . . 3
combine_mgNetworks . . . 3
get_citation . . . 4
get_collection . . . 5
get_network_by_id . . . 6
rmangal_endpoints . . . 8
rmangal_request . . . 8
search_datasets . . . 9
search_interactions . . . 11
search_networks . . . 12
search_nodes . . . 14
search_references . . . 15
search_taxonomy . . . 16

Index **18**

`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)

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

---

avail_type *List interactions type contained in mangal-db*

Description

List interactions type contained in mangal-db

Usage

```
avail_type()
```

---

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))
mg_random_1074 <- get_collection(c(1074))
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)

- `get_citation(mgNetwork)`: Get BibTeX entries for the publication associated to the network.
- `get_citation(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 accepts 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 from any `search_*()` function.
`...`	arguments to be passed on to `get_network_by_id()`.

Value

Returns a object of class `mgNetworksCollection` which is a collection (actually, a list) of `mgNetwork` objects `get_network_by_id()`).

Methods (by class)

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

See Also

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

Examples

```
mg_2 <- get_collection(c(1076:1077))
mg_anemone <- get_collection(search_networks(query = "anemone%"))
```

---

get_network_by_id *Retrieve network information, 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, force_collection = FALSE)

get_network_by_id_indiv(id, as_sf = FALSE)

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

## S3 method for class 'mgNetworksCollection'
```

```
print(x, n = 6, ...)

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

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

Arguments

ids	a vector of Mangal identifier for networks (integer).
as_sf	a logical. Should networks metadata be converted into an sf object? Note that to use this feature sf must be installed.
force_collection	a logical. Should the output to be of class mgNetworksCollection even if it includes only one network.
id	a single network identifier (integer).
x	an object of class mgNetwork or mgNetworksCollection.
...	ignored.
n	maximum number of networks to display.
object	object of class mgNetwork or mgNetworksCollection.

Value

A mgNetwork object includes five data frames:

- 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 an individual network by its identifier.

Examples

```
net18 <- get_network_by_id(id = 18)
net18_c <- get_network_by_id(id = 18, force_collection = TRUE)
nets <- get_network_by_id(id = c(18, 23))
```

`rmangal_endpoints` *Mangal API's endpoints*

Description

Mangal API's endpoints

Usage

```
rmangal_endpoints
```

Format

A data frame with 2 columns and 1 row per endpoint.

References

- https://mangal-interactions.github.io/mangal-api/

`rmangal_request` *Low-level request function for the Mangal API*

Description

These functions send a request to a specified endpoint of the Mangal API.

Usage

```
rmangal_request(endpoint, query = NULL, limit = 100, cache = FALSE, ...)

rmangal_request_singleton(endpoint, id, cache = FALSE, ...)
```

Arguments

endpoint	A character string specifying the API endpoint to query. Must be one of the endpoints listed in rmangal_endpoints().
query	Either a character string for a keyword search, or a named list for custom queries. If NULL, all records are returned.
limit	An integer specifying the maximum number of results per page. Default is 100.
cache	Logical or character. If TRUE, results are cached in a temporary directory. If a character string, it specifies the cache directory. If FALSE (default), no caching is used. See httr2::req_cache() for further details about caching in this context.
...	Additional arguments (currently ignored).
id	An integer or character specifying the unique identifier of the resource to retrieve. Must be a single value.

Value

An object of class mgGetResponses containing the response body and the raw HTTP response(s).

An object of class mgGetResponses containing the response body and the raw HTTP response.

Functions

- rmangal_request(): Send a request to a specified endpoint of the Mangal API and returns all matching results.
- rmangal_request_singleton(): Retrieves a single resource by its ID from a specified endpoint of the Mangal API.

Examples

```
# Search for networks with keyword "insect"
result <- rmangal_request("network", query = "insect", limit = 10)

# Custom query for a specific dataset
result <- rmangal_request("network", query = list(dataset_id = 19))


# Retrieve network with ID 5013
result <- rmangal_request_singleton("network", id = 5013)
```

---

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, ...)
```

Arguments

query	either a character string including a single keyword or a named list containing a custom query (see details section below). Note that if an empty character string is passed, then all available entries are returned.
...	Additional arguments passed to rmangal_request(), including the argument cache that allows requests caching.

Details

If query is a named list, the name used should be one of the following:

- 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 dataset
- 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 object including all datasets corresponding to the query. For each dataset entry, the networks and the original reference are attached.

References

- https://mangal.io/#/
- https://mangal-interactions.github.io/mangal-api/#datasets

Examples

```
# Return all datasets (takes time)
# all_datasets <- search_datasets("")
# all_datasets
# class(all_datasets)
# Search with keyword
mg_lagoon <- search_datasets(query = "lagoon")
# Search with a custom query (specific column)
mg_kemp <- search_datasets(query = list(name = "kemp_1977"))
mg_16 <- search_datasets(query = list(ref_id = 16))
```

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).

Usage

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

Arguments

query	either a character string including a single keyword or a named list containing a custom query (see details section below). Note that if an empty character string is passed, then all available entries are returned.
type	a character one of the interactions type available (see details). 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.
...	Additional arguments passed to rmangal_request(), including the argument cache that allows requests caching.

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 originates;
- node_to: node to which the interaction goes;
- type: use argument type instead.

Note that for lists with more than one element, only the first element is used, the others are ignored. The type of interactions (argument type) currently available are the following

- "competition";
- "amensalism";

- "neutralism";
- "commensalism";
- "mutualism";
- "parasitism";
- "predation";
- "herbivory";
- "symbiosis";
- "scavenger";
- "detritivore".

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

- https://mangal.io/#/
- https://mangal-interactions.github.io/mangal-api/#taxonomy

Examples

```
df_inter <- search_interactions(type = "competition")
# Get all networks containing competition
competition_networks <- get_collection(df_inter)
df_net_926 <- search_interactions(list(network_id = 926))
```

`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, ...)

search_networks_sf(query_sf, ...)
```

Arguments

query	either a character string including a single keyword or a named 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 networks available are returned.
...	Further arguments passed to rmangal_request(), including the argument cache that allows requests caching.
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 information.

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

- https://mangal.io/#/
- https://mangal-interactions.github.io/mangal-api/#networks

Examples

```
mg_insect <- search_networks(query = "insect%")
# Retrieve the search results
nets_insect <- get_collection(mg_insect)
# Spatial query
if (requireNamespace("sf", quietly = TRUE)) {
  area <- sf::st_read(system.file("shape/nc.shp", package = "sf"))
  networks_in_area <- search_networks_sf(area)
  plot(networks_in_area)
} else {
  cli::cli_warn("Package sf is missing")
```

```
}
# 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, ...)
```

Arguments

`query`	either a character string including a single keyword or a named list containing a custom query (see details section below). Note that if an empty character string is passed, then all available entries are returned.
`...`	Additional arguments passed to `rmangal_request()`, including the argument `cache` that allows requests caching.

Details

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

- `id`: unique identifier of the nodes;
- `original_name`: taxonomic name as in the original publication;
- `node_level`: either population, taxon or individual;
- `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 basically is a `data.frame` object including taxa that are matching the query and corresponding information. All networks in which taxa are involved are also attached to the `data.frame`.

References

- https://mangal.io/#/
- https://mangal-interactions.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 keyword 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, ...)
```

Arguments

query	either a character string including a single keyword or a named list containing a custom query (see details section below). Note that if an empty character string is passed, then all available entries are returned.
doi	character a Digital Object Identifier (DOI) of the article. Note that query is ignored if doi is specified.
...	Additional arguments passed to rmangal_request(), including the argument cache that allows requests caching.

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

- https://mangal.io/#/
- https://mangal-interactions.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,
  ...
)
```

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 (<itis.gov>).
gbif	a numeric. Unique taxonomic identifier from Global Biodiversity Information Facility (<www.gbif.org>).
eol	a numeric. Unique taxonomic identifier from Encyclopedia of Life (<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 (https://boldsystems.org/).
ncbi	a numeric. Unique taxonomic identifier from National Center for Biotechnology Information (https://www.ncbi.nlm.nih.gov).
...	Further arguments passed to rmangal_request(), including the argument cache that allows requests caching.

Details

Taxon names of the taxonomy table were validated with TNRS (see https://tnrs.biendata.org) and/or GNR (see https://resolver.globalnames.org/). The taxon names in this table 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-interactions/contribute/issues/new/choose/.

Value

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

References

- https://mangal.io/#/
- https://mangal-interactions.github.io/mangal-api/#taxonomy

See Also

search_nodes()

Examples

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

Index

∗ **datasets**
    rmangal_endpoints, 8

as.igraph.mgNetwork, 2
as.igraph.mgNetworksCollection
        *(*as.igraph.mgNetwork*)*, 2
avail_type, 3

combine_mgNetworks, 3

get_citation, 4
get_collection, 5
get_network_by_id, 6
get_network_by_id(), *5*
get_network_by_id_indiv
        *(*get_network_by_id*)*, 6

httr2::req_cache(), *9*

print.mgNetwork *(*get_network_by_id*)*, 6
print.mgNetworksCollection
        *(*get_network_by_id*)*, 6

rmangal_endpoints, 8
rmangal_endpoints(), *9*
rmangal_request, 8
rmangal_request(), *10, 11*, *13–15*, *17*
rmangal_request_singleton
        *(*rmangal_request*)*, 8

search_datasets, 9
search_datasets(), *6*
search_interactions, 11
search_interactions(), *6*
search_networks, 12
search_networks(), *6*
search_networks_sf *(*search_networks*)*, 12
search_nodes, 14
search_nodes(), *6*, *17*
search_references, 15
search_references(), *6*
search_taxonomy, 16
search_taxonomy(), *6*, *15*
summary.mgNetwork *(*get_network_by_id*)*, 6
summary.mgNetworksCollection
        *(*get_network_by_id*)*, 6