Package 'bold'

August 29, 2016

Title Interface to Bold Systems 'API'

Description A programmatic interface to the Web Service methods provided by Bold Systems for genetic 'barcode' data. Functions include methods for searching by sequences by taxonomic names, ids, collectors, and institutions; as well as a function for searching for specimens, and downloading trace files.

Version 0.3.5

License MIT + file LICENSE

Date 2016-03-28

URL https://github.com/ropensci/bold

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

VignetteBuilder knitr

LazyData yes

Imports methods, utils, stats, xml2, httr, stringr, assertthat, jsonlite, reshape, plyr

Suggests sangerseqR, knitr, testthat, covr

RoxygenNote 5.0.1

NeedsCompilation no

Author Scott Chamberlain [aut, cre]

Maintainer Scott Chamberlain <myrmecocystus@gmail.com>

Repository CRAN

Date/Publication 2016-03-29 10:48:27

R topics documented:

bold-package 2
bold_identify 3
bold_seq 4
bold_seqspec 6
bold_specimens 7

bold_tax_id 9
bold_tax_name 10
bold_trace 11
sequences 13

Index **14**

`bold-package` *bold: A programmatic interface to the Barcode of Life data.*

Description

bold: A programmatic interface to the Barcode of Life data.

About

This package gives you access to data from BOLD System `http://www.boldsystems.org/` via their API.

Functions

- `bold_specimens` - Search for specimen data.
- `bold_seq` - Search for and retrieve sequences.
- `bold_seqspec` - Get sequence and specimen data together.
- `bold_trace` - Get trace files - saves to disk.
- `read_trace` - Read trace files into R.
- `bold_tax_name` - Get taxonomic names via input names.
- `bold_tax_id` - Get taxonomic names via BOLD identifiers.
- `bold_identify` - Search for match given a COI sequence.

Interestingly, they provide xml and tsv format data for the specimen data, while they provide fasta data format for the sequence data. So for the specimen data you can get back raw XML, or a data frame parsed from the tsv data, while for sequence data you get back a list (b/c sequences are quite long and would make a data frame unwieldy).

`bold_identify` *Search for matches to sequences against the BOLD COI database.*

Description

Search for matches to sequences against the BOLD COI database.

Usage

```
bold_identify(sequences, db = "COX1", response = FALSE, ...)
```

Arguments

`sequences`	(character) Returns all records containing matching marker codes. Required.
`db`	(character) The database to match against, one of COX1, COX1_SPECIES, COX1_SPECIES_PUBLIC, OR COX1_L604bp. See Details for more information.
`response`	(logical) Note that response is the object that returns from the Curl call, useful for debugging, and getting detailed info on the API call.
`...`	Further args passed on to httr::GET, main purpose being curl debugging

Details

Detailed description of options for the db parmeter:

- COX1 Every COI barcode record with a species level identification and a minimum sequence length of 500bp. This includes many species represented by only one or two specimens as well as all species with interim taxonomy.
- COX1_SPECIES Every COI barcode record on BOLD with a minimum sequence length of 500bp (warning: unvalidated library and includes records without species level identification). This includes many species represented by only one or two specimens as well as all species with interim taxonomy. This search only returns a list of the nearest matches and does not provide a probability of placement to a taxon.
- COX1_SPECIES_PUBLIC All published COI records from BOLD and GenBank with a minimum sequence length of 500bp. This library is a collection of records from the published projects section of BOLD.
- OR COX1_L604bp Subset of the Species library with a minimum sequence length of 640bp and containing both public and private records. This library is intended for short sequence identification as it provides maximum overlap with short reads from the barcode region of COI.

Value

A data.frame with details for each specimen matched.

References

http://www.boldsystems.org/index.php/resources/api?type=idengine

Examples

```
## Not run:
seq <- sequences$seq1
head(bold_identify(sequences=seq)[[1]])
head(bold_identify(sequences=seq, db='COX1_SPECIES')[[1]])
bold_identify(sequences=seq, response=TRUE)

# Multiple sequences
out <- bold_identify(sequences=c(sequences$seq2, sequences$seq3), db='COX1')
lapply(out, head)

# curl debugging
library('httr')
bold_identify(sequences=seq, response=TRUE, config=verbose())[[1]]

## End(Not run)
```

---

bold_seq *Search BOLD for sequences.*

Description

Get sequences for a taxonomic name, id, bin, container, institution, researcher, geographic place, or gene.

Usage

```
bold_seq(taxon = NULL, ids = NULL, bin = NULL, container = NULL,
  institutions = NULL, researchers = NULL, geo = NULL, marker = NULL,
  response = FALSE, ...)
```

Arguments

taxon	(character) Returns all records containing matching taxa. Taxa includes the ranks of phylum, class, order, family, subfamily, genus, and species.
ids	(character) Returns all records containing matching IDs. IDs include Sample IDs, Process IDs, Museum IDs and Field IDs.
bin	(character) Returns all records contained in matching BINs. A BIN is defined by a Barcode Index Number URI.
container	(character) Returns all records contained in matching projects or datasets. Containers include project codes and dataset codes
institutions	(character) Returns all records stored in matching institutions. Institutions are the Specimen Storing Site.

researchers	(character) Returns all records containing matching researcher names. Researchers include collectors and specimen identifiers.
geo	(character) Returns all records collected in matching geographic sites. Geographic sites includes countries and province/states.
marker	(character) Returns all records containing matching marker codes.
response	(logical) Note that response is the object that returns from the Curl call, useful for debugging, and getting detailed info on the API call.
...	Further args passed on to httr::GET, main purpose being curl debugging

Value

A list with each element of length 4 with slots for id, name, gene, and sequence.

References

http://www.boldsystems.org/index.php/resources/api#sequenceParameters

Examples

```
## Not run:
bold_seq(taxon='Coelioxys')
bold_seq(taxon='Aglae')
bold_seq(taxon=c('Coelioxys','Osmia'))
bold_seq(ids='ACRJP618-11')
bold_seq(ids=c('ACRJP618-11','ACRJP619-11'))
bold_seq(bin='BOLD:AAA5125')
bold_seq(container='ACRJP')
bold_seq(researchers='Thibaud Decaens')
bold_seq(geo='Ireland')
bold_seq(geo=c('Ireland','Denmark'))

# Return the httr response object for detailed Curl call response details
res <- bold_seq(taxon='Coelioxys', response=TRUE)
res$url
res$status_code
res$headers

## curl debugging
### You can do many things, including get verbose output on the curl call, and set a timeout
library("httr")
bold_seq(taxon='Coelioxys', config=verbose())[1:2]
# bold_seqspec(taxon='Coelioxys', config=timeout(0.1))

## End(Not run)
```

`bold_seqspec` *Get BOLD specimen + sequence data.*

Description

Get BOLD specimen + sequence data.

Usage

```
bold_seqspec(taxon = NULL, ids = NULL, bin = NULL, container = NULL,
  institutions = NULL, researchers = NULL, geo = NULL, marker = NULL,
  response = FALSE, format = "tsv", sepfasta = FALSE, ...)
```

Arguments

`taxon`	(character) Returns all records containing matching taxa. Taxa includes the ranks of phylum, class, order, family, subfamily, genus, and species.
`ids`	(character) Returns all records containing matching IDs. IDs include Sample IDs, Process IDs, Museum IDs and Field IDs.
`bin`	(character) Returns all records contained in matching BINs. A BIN is defined by a Barcode Index Number URI.
`container`	(character) Returns all records contained in matching projects or datasets. Containers include project codes and dataset codes
`institutions`	(character) Returns all records stored in matching institutions. Institutions are the Specimen Storing Site.
`researchers`	(character) Returns all records containing matching researcher names. Researchers include collectors and specimen identifiers.
`geo`	(character) Returns all records collected in matching geographic sites. Geographic sites includes countries and province/states.
`marker`	(character) Returns all records containing matching marker codes.
`response`	(logical) Note that response is the object that returns from the Curl call, useful for debugging, and getting detailed info on the API call.
`format`	(character) One of xml or tsv (default). tsv format gives back a data.frame object. xml gives back parsed xml as a
`sepfasta`	(logical) If TRUE, the fasta data is separated into a list with names matching the processid's from the data frame
`...`	Further args passed on to httr::GET, main purpose being curl debugging

Value

Either a data.frame, parsed xml, a httr response object, or a list with length two (a data.frame w/o nucleotide data, and a list with nucleotide data)

References

http://www.boldsystems.org/index.php/resources/api#combined

Examples

```
## Not run:
bold_seqspec(taxon='Osmia')
bold_seqspec(taxon='Osmia', format='xml')
bold_seqspec(taxon='Osmia', response=TRUE)
res <- bold_seqspec(taxon='Osmia', sepfasta=TRUE)
res$fasta[1:2]
res$fasta['GBAH0293-06']

# records that match a marker name
res <- bold_seqspec(taxon="Melanogrammus aeglefinus", marker="COI-5P")

# records that match a geographic locality
res <- bold_seqspec(taxon="Melanogrammus aeglefinus", geo="Canada")

## curl debugging
### You can do many things, including get verbose output on the curl call, and set a timeout
library("httr")
head(bold_seqspec(taxon='Osmia', config=verbose()))
## timeout
# head(bold_seqspec(taxon='Osmia', config=timeout(1)))
## progress
# x <- bold_seqspec(taxon='Osmia', config=progress())

## End(Not run)
```

`bold_specimens` *Search BOLD for specimens.*

Description

Search BOLD for specimens.

Usage

```
bold_specimens(taxon = NULL, ids = NULL, bin = NULL, container = NULL,
  institutions = NULL, researchers = NULL, geo = NULL, response = FALSE,
  format = "tsv", ...)
```

Arguments

`taxon`	(character) Returns all records containing matching taxa. Taxa includes the ranks of phylum, class, order, family, subfamily, genus, and species.
`ids`	(character) Returns all records containing matching IDs. IDs include Sample IDs, Process IDs, Museum IDs and Field IDs.

bin	(character) Returns all records contained in matching BINs. A BIN is defined by a Barcode Index Number URI.
container	(character) Returns all records contained in matching projects or datasets. Containers include project codes and dataset codes
institutions	(character) Returns all records stored in matching institutions. Institutions are the Specimen Storing Site.
researchers	(character) Returns all records containing matching researcher names. Researchers include collectors and specimen identifiers.
geo	(character) Returns all records collected in matching geographic sites. Geographic sites includes countries and province/states.
response	(logical) Note that response is the object that returns from the Curl call, useful for debugging, and getting detailed info on the API call.
format	(character) One of xml or tsv (default). tsv format gives back a data.frame object. xml gives back parsed xml as a
...	Further args passed on to httr::GET, main purpose being curl debugging

References

http://www.boldsystems.org/index.php/resources/api#specimenParameters

Examples

```
## Not run:
bold_specimens(taxon='Osmia')
bold_specimens(taxon='Osmia', format='xml')
# bold_specimens(taxon='Osmia', response=TRUE)
res <- bold_specimens(taxon='Osmia', format='xml', response=TRUE)
res$url
res$status_code
res$headers

# More than 1 can be given for all search parameters
bold_specimens(taxon=c('Coelioxys','Osmia'))

## curl debugging
### These examples below take a long time, so you can set a timeout so that it stops by X sec
library("httr")
head(bold_specimens(taxon='Osmia', config=verbose()))
# head(bold_specimens(geo='Costa Rica', config=timeout(6)))
# head(bold_specimens(taxon="Formicidae", geo="Canada", config=timeout(6)))

## End(Not run)
```

bold_tax_id *Search BOLD for taxonomy data by BOLD ID.*

Description

Search BOLD for taxonomy data by BOLD ID.

Usage

```
bold_tax_id(id, dataTypes = "basic", includeTree = FALSE,
  response = FALSE, ...)
```

Arguments

id	(integer) One or more BOLD taxonomic identifiers. required.
dataTypes	(character) Specifies the datatypes that will be returned. 'all' returns all data. 'basic' returns basic taxon information. 'images' returns specimen images.
includeTree	(logical) If TRUE (default: FALSE), returns a list containing information for parent taxa as well as the specified taxon.
response	(logical) Note that response is the object that returns from the Curl call, useful for debugging, and getting detailed info on the API call.
...	Further args passed on to httr::GET, main purpose being curl debugging

References

http://boldsystems.org/index.php/resources/api?type=taxonomy#idParameters

See Also

bold_tax_name

Examples

```
## Not run:
bold_tax_id(id=88899)
bold_tax_id(id=88899, includeTree=TRUE)
bold_tax_id(id=88899, includeTree=TRUE, dataTypes = "stats")
bold_tax_id(id=c(88899,125295))

## dataTypes parameter
bold_tax_id(id=88899, dataTypes = "basic")
bold_tax_id(id=88899, dataTypes = "stats")
bold_tax_id(id=88899, dataTypes = "images")
bold_tax_id(id=88899, dataTypes = "geo")
bold_tax_id(id=88899, dataTypes = "sequencinglabs")
bold_tax_id(id=88899, dataTypes = "depository")
bold_tax_id(id=88899, dataTypes = "thirdparty")
bold_tax_id(id=88899, dataTypes = "all")
```

```
bold_tax_id(id=c(88899,125295), dataTypes = "geo")
bold_tax_id(id=c(88899,125295), dataTypes = "images")

## Passing in NA
bold_tax_id(id = NA)
bold_tax_id(id = c(88899,125295,NA))

## get httr response object only
bold_tax_id(id=88899, response=TRUE)
bold_tax_id(id=c(88899,125295), response=TRUE)

## curl debugging
library('httr')
bold_tax_id(id=88899, config=verbose())

## End(Not run)
```

`bold_tax_name` *Search BOLD for taxonomy data by taxonomic name.*

Description

Search BOLD for taxonomy data by taxonomic name.

Usage

```
bold_tax_name(name, fuzzy = FALSE, response = FALSE, ...)
```

Arguments

`name`	(character) One or more scientific names. required.
`fuzzy`	(logical) Whether to use fuzzy search or not (default: FALSE).
`response`	(logical) Note that response is the object that returns from the Curl call, useful for debugging, and getting detailed info on the API call.
`...`	Further args passed on to httr::GET, main purpose being curl debugging

Details

The `dataTypes` parameter is not supported in this function. If you want to use that parameter, get an ID from this function and pass it into `bold_tax_id`, and then use the `dataTypes` parameter.

References

http://boldsystems.org/index.php/resources/api?type=taxonomy#nameParameters

See Also

`bold_tax_id`

Examples

```
## Not run:
bold_tax_name(name='Diplura')
bold_tax_name(name='Osmia')
bold_tax_name(name=c('Diplura','Osmia'))
bold_tax_name(name=c("Apis","Puma concolor","Pinus concolor"))
bold_tax_name(name='Diplur', fuzzy=TRUE)
bold_tax_name(name='Osm', fuzzy=TRUE)

## get httr response object only
bold_tax_name(name='Diplura', response=TRUE)
bold_tax_name(name=c('Diplura','Osmia'), response=TRUE)

## Names with no data in BOLD database
bold_tax_name("Nasiaeshna pentacantha")
bold_tax_name(name = "Cordulegaster erronea")
bold_tax_name(name = "Cordulegaster erronea", response=TRUE)

## curl debugging
library('httr')
bold_tax_name(name='Diplura', config=verbose())

## End(Not run)
```

`bold_trace` *Get BOLD trace files*

Description

Get BOLD trace files

Usage

```
bold_trace(taxon = NULL, ids = NULL, bin = NULL, container = NULL,
  institutions = NULL, researchers = NULL, geo = NULL, marker = NULL,
  dest = NULL, overwrite = TRUE, progress = TRUE, ...)

read_trace(x)
```

Arguments

`taxon`	(character) Returns all records containing matching taxa. Taxa includes the ranks of phylum, class, order, family, subfamily, genus, and species.
`ids`	(character) Returns all records containing matching IDs. IDs include Sample IDs, Process IDs, Museum IDs and Field IDs.
`bin`	(character) Returns all records contained in matching BINs. A BIN is defined by a Barcode Index Number URI.

container	(character) Returns all records contained in matching projects or datasets. Containers include project codes and dataset codes
institutions	(character) Returns all records stored in matching institutions. Institutions are the Specimen Storing Site.
researchers	(character) Returns all records containing matching researcher names. Researchers include collectors and specimen identifiers.
geo	(character) Returns all records collected in matching geographic sites. Geographic sites includes countries and province/states.
marker	(character) Returns all records containing matching marker codes.
dest	(character) A directory to write the files to
overwrite	(logical) Overwrite existing directory and file?
progress	(logical) Print progress or not. Uses progress.
...	Futher args passed on to GET.
x	Object to print or read.

References

http://www.boldsystems.org/index.php/resources/api#trace

Examples

```
## Not run:
# Use a specific destination directory
bold_trace(taxon='Bombus', geo='Alaska', dest="~/mytarfiles")

# Another example
bold_trace(ids='ACRJP618-11', dest="~/mytarfiles")
bold_trace(ids=c('ACRJP618-11','ACRJP619-11'), dest="~/mytarfiles")

# read file in
x <- bold_trace(ids=c('ACRJP618-11','ACRJP619-11'), dest="~/mytarfiles")
(res <- read_trace(x$ab1[2]))

# The progress dialog is pretty verbose, so quiet=TRUE is a nice touch, but not by default
# Beware, this one take a while
x <- bold_trace(taxon='Osmia', quiet=TRUE)

if (requireNamespace("sangerseqR", quietly = TRUE)) {
 library("sangerseqR")
 primarySeq(res)
 secondarySeq(res)
 head(traceMatrix(res))
}

## End(Not run)
```

`sequences` *List of 3 nucleotide sequences to use in examples for the* `bold_identify` *function*

Description

List of 3 nucleotide sequences to use in examples for the `bold_identify` function

Details

Each sequence is a character string, of lengths 410, 600, and 696.

Index

∗Topic **data**
sequences, 13

bold *(*bold-package*)*, 2
bold-package, 2
bold_identify, *2*, 3, *13*
bold_seq, *2*, 4
bold_seqspec, *2*, 6
bold_specimens, *2*, 7
bold_tax_id, *2*, 9, *10*
bold_tax_name, *2*, 10
bold_trace, *2*, 11

GET, *12*

progress, *12*

read_trace, *2*
read_trace *(*bold_trace*)*, 11

sequences, 13