

Package: vectorsurvR (via r-universe)

October 1, 2024

Type Package

Title Data Access and Analytical Tools for 'VectorSurv' Users

Version 1.2.1

Description Allows registered 'VectorSurv' <<https://vectorsurv.org/>> users access to data through the 'VectorSurv API' <<https://api.vectorsurv.org/>>. Additionally provides functions for analysis and visualization.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Imports rstudioapi, dplyr, jsonlite, kableExtra, knitr, lubridate, stringr, plotly, ggplot2, httr, tidyr, magrittr, DT

Suggests testthat (>= 3.0.0), rmarkdown, devtools,

VignetteBuilder knitr

Config/testthat/edition 3

Depends R (>= 2.10)

NeedsCompilation no

Author Christina De Cesaris [aut, cre]

Maintainer Christina De Cesaris <cmdecesaris@ucdavis.edu>

Repository CRAN

Date/Publication 2024-09-30 22:10:02 UTC

Contents

getAbundance	2
getAbundanceAnomaly	3
getArthroCollections	4
getInfectionRate	5
getPools	6

getPoolsComparisionTable	7
getRegions	8
getSites	8
getToken	9
getVectorIndex	9
plotInfectionRate	10
processAbunAnom	11
sample_collections	12
sample_pools	12
Index	14

getAbundance	<i>Calculate abundance</i>
--------------	----------------------------

Description

Calculates abundance

Usage

```
getAbundance(  
  collections,  
  interval,  
  species_list = NULL,  
  trap_list = NULL,  
  species_separate = FALSE  
)
```

Arguments

collections	Collections data retrieved from getArthroCollections()
interval	Calculation interval for abundance, accepts “collection_date”, “Biweek”, “Week”, and “Month.
species_list	Species filter for calculating abundance. Species_display_name is the accepted notation.To see a list of species present in your data run unique(collections\$species_display_name). If species is unspecified, the default NULL will return data for all species in data.
trap_list	Trap filter for calculating abundance. Trap_acronym is the is the accepted notation. Run unique(collections\$trap_acronym) to see trap types present in your data. If trap_list is unspecified, the default NULL will return data for all trap types.
species_separate	Should the species in species_list have abundance calculated separately? Setting to FALSE calculates the combined abundance. The same result can be performed by calculating on one species at the time.

Value

A dataframe of abundance calculations

Examples

```
getAbundance(sample_collections,
              interval = 'Week',
              species_list = list('Cx pipiens'),
              trap_list = list('GRVD', 'CQ2'),
              species_separate = FALSE)
```

getAbundanceAnomaly	<i>Get Abundance Anomaly</i>
---------------------	------------------------------

Description

‘getAbundanceAnomaly(...)’ requires at least five years prior to the target_year of arthro collections data to calculate for the specified parameters. The function uses the methods of the Gateway Abundance Anomaly calculator, and will not work if there is fewer than five years of data present.

Usage

```
getAbundanceAnomaly(
  collections,
  interval,
  target_year,
  species_list = NULL,
  trap_list = NULL,
  species_separate = FALSE
)
```

Arguments

collections	Collections data retrieved from ‘getArthroCollections()’
interval	Calculation interval for abundance, accepts “collection_date”, “Biweek”, “Week”, and “Month
target_year	Year to calculate analysis on. Collections data must have a year range of at least (target_year - 5, target_year)
species_list	Species filter for calculating abundance. Species_display_name is the accepted notation. To see a list of species present in your data run ‘unique(collections\$species_display_name)’. If species is unspecified, the default NULL will return data for all species in data
trap_list	Trap filter for calculating abundance. Trap_acronym is the is the accepted notation. Run ‘unique(collections\$trap_acronym)’ to see trap types present in your data. If trap_list is unspecified, the default NULL will return data for all trap types

species_separate

Should the species in species_list have abundance calculated separately? Setting to FALSE calculates the combined abundance. The same result can be performed by calculating on one species at the time description

Value

Abundance anomaly calculation

Examples

```
getAbundanceAnomaly(sample_collections,"Biweek",target_year=2020, species_list="Cx pipiens")
```

getArthroCollections *Get arthropod collections data*

Description

‘getArthroCollections()’ obtains collections data on a year range [start_year, end_year] for authorized VectorSurv Gateway accounts.

Usage

```
getArthroCollections(token, start_year, end_year, arthropod, agency_ids = NULL)
```

Arguments

token	A valid access token returned from ‘getToken()’
start_year	Start year of data
end_year	End year of data
arthropod	Specify arthropod type from: ‘mosquito’, ‘tick’
agency_ids	Filter on agency id, default to NULL for all available agencies, otherwise provide a vector of agency ids

Value

A dataframe of collections

Examples

```
## Not run:
token = getToken()
collections = getArthroCollections(token, 2021, 2022, 'mosquito', 55, TRUE)
## End(Not run)
```

getInfectionRate	<i>Calculate infection rate</i>
------------------	---------------------------------

Description

'getInfectionRate()' requires at least five years prior to the target_year of arthro collections data to calculate for the specified parameters. The function uses the methods of the Gateway Abundance Anomaly calculator, and will not work if there is fewer than five years of data present.

Usage

```
getInfectionRate(
  pools,
  interval,
  target_disease,
  pt_estimate,
  scale = 1000,
  species_list = NULL,
  trap_list = NULL
)
```

Arguments

pools	Pools data retrieved from 'getPools()'
interval	Calculation interval for infection rate, accepts "collection_date", "Biweek", "Week", and "Month"
target_disease	The disease to calculate infection rate for--i.e. "WNV". Disease acronyms are the accepted input. To see a list of disease acronyms, run 'unique(pools\$target_acronym)'
pt_estimate	The estimation type for infection rate. Options include: "mle", "bc-mle", "mir"
scale	Constant to multiply infection rate by
species_list	Species filter for calculating infection rate species_display_name is the accepted notation. To see a list of species present in your data run 'unique(pools\$species_display_name)'. If species is unspecified, the default 'NULL' will return data for all species in data.
trap_list	Trap filter for calculating infection rate. Trap_acronym is the is the accepted notation. Run 'unique(pools\$trap_acronym)' to see trap types present in your data. If trap_list is unspecified, the default 'NULL' will return data for all trap types.

Value

Dataframe of infection rate calculation

Examples

```
getInfectionRate(sample_pools,
  interval = "Biweek",
  target_disease = "WNV",
  pt_estimate = "mle",
  scale = 1000,
  species_list = list("Cx pipiens"),
  trap_list = list("C02"))
```

getPools

*Get Pools data***Description**

Retrieves VectorSurv pools data for desired year range

Usage

```
getPools(token, start_year, end_year, arthropod, agency_ids = NULL)
```

Arguments

token	access token retrived from 'getToken()'
start_year	Beginning of year range
end_year	End of year range
arthropod	Specify arthropod type from: 'mosquito', 'tick', 'nontick'
agency_ids	Filter on agency id, default to NULL for all available agencies,otherwise provide a vector of agency ids

Value

Dataframe of pools data

Examples

```
## Not run:
token = getToken()
getPools(token, start_year = 2020, end_year = 2021, arthropod = 'tick', 55)
## End(Not run)
```

`getPoolsComparisionTable`*Get Pools Frequency Table*

Description

`'getPoolsComparisionTable()'` produces a frequency table for positive, negative, and pending pools counts by year and species. The more years present in the data, the larger the table.

Usage

```
getPoolsComparisionTable(  
  pools,  
  interval,  
  target_disease,  
  species_separate = FALSE  
)
```

Arguments

<code>pools</code>	Pools data retrieved from <code>'getPools()'</code>
<code>interval</code>	Calculation interval for comparison table, accepts <code>"collection_date"</code> , <code>"Biweek"</code> , <code>"Week"</code> , and <code>"Month"</code>
<code>target_disease</code>	The disease to calculate infection rate for–i.e. <code>"WNV"</code> . Disease acronyms are the accepted input. To see a list of disease acronyms, run <code>'unique(pools\$target_acronym)'</code>
<code>species_separate</code>	Should the pools comparison be split by species of each pool. Default is <code>FALSE</code>

Value

Frequency table of for pools data

Examples

```
getPoolsComparisionTable(sample_pools,  
  interval = "Biweek",  
  target_disease = "WNV",  
  species_separate = TRUE)
```

`getRegions`*Get region data*

Description

`'getSites()'` obtains site data for authorized VectorSurv Gateway accounts.

Usage

```
getRegions(token)
```

Arguments

token A valid access token returned from `'getToken()'`

Value

A dataframe of region data, used internally to merge spatial information to collections

`getSites`*Get sites data*

Description

`'getSites()'` obtains site data for authorized VectorSurv Gateway accounts.

Usage

```
getSites(token)
```

Arguments

token A valid access token returned from `'getToken()'`

Value

A dataframe of site data

Examples

```
## Not run:  
token = getToken()  
sites = getSites(token)  
## End(Not run)
```

getToken	<i>Get authentication token</i>
----------	---------------------------------

Description

getToken() returns a token needed to run getArthroCollections() and getPools(). Prints agencies associated with account credentials. The function prompts users for a VectorSurv account credentials.

Usage

```
getToken()
```

Value

User token

Examples

```
## Not run: token = getToken()
```

getVectorIndex	<i>Calculate vector index</i>
----------------	-------------------------------

Description

'getVectorIndex()' requires at least five years prior to the target_year of arthro collections data to calculate for the specified parameters. The function uses the methods of the Gateway Abundance Anomaly calculator, and will not work if there is fewer than five years of data present.

Usage

```
getVectorIndex(  
  collections,  
  pools,  
  interval,  
  target_disease,  
  pt_estimate,  
  scale = 1000,  
  species_list = NULL,  
  trap_list = NULL  
)
```

Arguments

<code>collections</code>	Collections data retrieved from <code>'getArthroCollections()'</code>
<code>pools</code>	Pools data retrieved from <code>'getPools()'</code>
<code>interval</code>	Calculation interval for vector index, accepts <code>"collection_date"</code> , <code>"Biweek"</code> , <code>"Week"</code> , and <code>"Month"</code>
<code>target_disease</code>	The disease to calculate infection rate for—i.e. <code>"WNV"</code> . Disease acronyms are the accepted input. To see a list of disease acronyms, run <code>'unique(pools\$target_acronym)'</code>
<code>pt_estimate</code>	The estimation type for infection rate. Options include: <code>"mle"</code> , <code>"bc-"mle"</code> , <code>"mir"</code>
<code>scale</code>	Constant to multiply infection rate, default is 1000
<code>species_list</code>	Species filter for calculating abundance. <code>Species_display_name</code> is the accepted notation. To see a list of species present in your data run <code>'unique(pools\$species_display_name)'</code> . If species is unspecified, the default <code>'NULL'</code> will return data for all species in data.
<code>trap_list</code>	Trap filter for calculating abundance. <code>Trap_acronym</code> is the accepted notation. Run <code>'unique(pools\$trap_acronym)'</code> to see trap types present in your data. If <code>trap_list</code> is unspecified, the default <code>'NULL'</code> will return data for all trap types.

Value

Dataframe containing the vector index calculation

Examples

```
getVectorIndex(sample_collections, sample_pools, "Month", "WNV", "mle" )
```

<code>plotInfectionRate</code>	<i>Plot infection rate</i>
--------------------------------	----------------------------

Description

`plotInfectionRate()` plots the output returned from `'getInfectionRate()'` with confidence intervals using `ggplot`

Usage

```
plotInfectionRate(InfRtOutput, year)
```

Arguments

<code>InfRtOutput</code>	Output from returned <code>'getInfectionRate()'</code>
<code>year</code>	Year to plot infection rate on

Value

`ggplot` object

Examples

```
IR = getInfectionRate(sample_pools,
                      interval = "Week",
                      target_disease = "WNV",
                      pt_estimate = "mle", species_list = c("Cx pipiens"),
                      trap_list = c("C02", "GRVD") )
plotInfectionRate(InfRtOutput = IR, year = 2017)
```

processAbunAnom	<i>Process abundance anomaly</i>
-----------------	----------------------------------

Description

‘processAbunAnom()’ processes the output returned from ‘getAbundanceAnomaly()’ into a long form suitable for plotting using ‘ggplot’

Usage

```
processAbunAnom(AbAnomOutput)
```

Arguments

AbAnomOutput output from ‘getAbunAnom()’

Value

Abundance anomaly output processed into long form

Examples

```
AbAnOut = getAbundanceAnomaly(sample_collections,
                              interval = "Biweek",
                              target_year = 2020,
                              species_list = c("Cx tarsalis", "Cx pipiens"),
                              species_separate = TRUE)
AbAnOut_L = processAbunAnom(AbAnOut)
```

sample_collections	<i>Sample Mosquito Collections Data</i>
--------------------	---

Description

Sample Mosquito Collections data imitates the essential components of real mosquito collections data

Usage

```
sample_collections
```

Format

A data frame with 2500 rows and 13 variables:

```
collection_id double Collection identification number
collection_date character The date the trap was picked up for collection
num_trap integer The number of unique traps in operation at one site
site_code integer Identifying code of site
surv_year double Surveillance year of collection
trap_nights integer The number of nights a trap was in the field
trap_problem_bit logical If there was an issue with the trap
num_count integer Number of arthropods present in collection
sex_type character Sex of collected arthropods
species_display_name character Species name of collected arthropods
trap_acronym character The acronym of the trap placed in the field
```

Source

<https://vectorsurv.org/>

sample_pools	<i>Sample Pools Data</i>
--------------	--------------------------

Description

Sample Pools data imitates the essential components of real mosquito pools data needed for calculations

Usage

```
sample_pools
```

Format

A data frame with 2500 rows and 10 variables:

`pool_id` integer Pool identification number
`surv_year` integer Surveillance year of pool
`site_code` integer Identifying code of site
`collection_date` character The date the trap was picked up for collection
`sex_type` integer Sex type of collected arthropods
`num_count` integer Number of arthropods present in collection
`target_acronym` character The disease being tested for in the pool
`method_name` character Method used to test pool for disease
`status_name` character Status of the tested disease, confirmed or negative
`trap_acronym` character The acronym of the trap placed in the field
`species_display_name` character Species name of collected arthropods

Source

<https://vectorsurv.org/>

Index

- * **abundance**
 - getAbundanceAnomaly, [3](#)
- * **authentication**
 - getToken, [9](#)
- * **datasets**
 - sample_collections, [12](#)
 - sample_pools, [12](#)
- * **infection**
 - getInfectionRate, [5](#)
- * **pools**
 - getInfectionRate, [5](#)
 - getPools, [6](#)
- * **rate**
 - getInfectionRate, [5](#)

getAbundance, [2](#)
getAbundanceAnomaly, [3](#)
getArthroCollections, [4](#)
getInfectionRate, [5](#)
getPools, [6](#)
getPoolsComparisionTable, [7](#)
getRegions, [8](#)
getSites, [8](#)
getToken, [9](#)
getVectorIndex, [9](#)

plotInfectionRate, [10](#)
processAbunAnom, [11](#)

sample_collections, [12](#)
sample_pools, [12](#)