

Package ‘vortexR’

February 6, 2019

Type Package

Title Post Vortex Simulation Analysis

Description Facilitate Post Vortex Simulation Analysis by offering tools to collate multiple Vortex (v10) output files into one R object, and analyse the collated output statistically. Vortex is a software for the development of individual-based model for population dynamic simulation (see <<http://www.vortex10.org/Vortex10.aspx>>).

Version 1.1.6

Author Carlo Pacioni [aut, cre],
Florian W. Mayer [aut]

Maintainer Carlo Pacioni <C.Pacioni@Murdoch.edu.au>

License GPL-3

URL <https://github.com/carlopacioni/vortexR/>

BugReports <https://github.com/carlopacioni/vortexR/issues>

Depends R (>= 3.1.0)

Imports betareg, data.table, ggplot2, glmulti, gtools (>= 3.4.2),
GGally (>= 1.3.0), irr (>= 0.84.1), plyr, R.utils, stringr,
vortexRdata (>= 1.0.3)

Suggests devtools, gridExtra, knitr, rmarkdown, roxygen2, testthat

VignetteBuilder knitr

RoxygenNote 6.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2019-02-06 12:50:03 UTC

R topics documented:

collate_dat	2
collate_one_dat	4
collate_proc_data	4

collate_run	5
collate_yr	6
CompileIter	7
conv_1_yr	8
df2disk	9
dot_plot	9
fit_regression	10
get_file_paths	13
line_plot_year	14
line_plot_year_mid	15
lookup_table	16
m_scatter	17
Nadults	18
Ne	19
pac.clas	20
pac.clas.lookup	21
pac.clas.Nadults	21
pac.clas.Ne	22
pac.clas.pairw	22
pac.lhs	23
pac.run.lhs	23
pac.yr	24
pairwise	24
Pextinct	27
PrefixAndRepeat	28
pval	28
rRec	29
se2sd	30
SSMD_matrix	31
sta.evy5	32
sta.evy5.b11	33
sta.main	33
vortexR	34
Index	36

collate_dat	<i>Collate Vortex .dat or .stdat output files into one data.frame</i>
-------------	---

Description

collate_dat collates all Vortex output files matching a given project name (and scenario name when relevant) in a given directory into one data.frame using collate_one_dat.

Usage

```
collate_dat(project, runs, scenario = NULL, dir_in = NULL,
            save2disk = TRUE, dir_out = "ProcessedData", verbose = TRUE)
```

Arguments

project	The Vortex project name to be imported
runs	The number of Vortex simulation runs
scenario	The scenario name if ST, default: NULL
dir_in	The local folder containing Vortex files, default: NULL. If not specified, will fall back to use current working directory.
save2disk	Whether to save the data as rda and csv, default: TRUE
dir_out	The local path to store the output. Default: ProcessedData
verbose	Progress messages, default: TRUE

Details

The number of Vortex simulation runs has to be specified, as it cannot be inferred by vortexR.

To read Vortex output files from Sensitivity Testing with the extension '.stdat', specify the scenario name. If scenario=NULL, all files with extension .dat, matching the project name will be imported.

dir_in may contain other files; only files matching the project (and, optionally, the scenario) name will be read.

dir_out is created within the working directory unless a full path is provided.

If no matching files are found in the given directory, an error is reported.

When verbose=TRUE the progress (i.e. the file being read) is reported on screen.

Value

a data.frame with data from all matching Vortex files or NULL

Examples

```
# Using Campbell et al. and Pacioni et al. example files.
# See ?pacioni and ?campbell for more details on example files.
require(vortexRdata)
camp.dir <- system.file('extdata', 'campbell', package='vortexRdata')
pac.dir <- system.file('extdata', 'pacioni', package='vortexRdata')

# Campbell example, project 'Starlingv3PopBased' (.dat)
starling <- collate_dat('Starlingv3PopBased', 10000,
  dir_in=camp.dir, save2disk=FALSE)

# Read data from all .stdat of the project 'Pacioni_et_al' and the ST scenario
# 'ST_Classic' and store the output in the object 'woylie.st.classic'
woylie.st.classic <- collate_dat('Pacioni_et_al', 3, scenario = 'ST_Classic',
  dir_in = pac.dir, save2disk=FALSE)

# Save collated data as .Rda and .txt
## Not run:
# Read data from all .stdat of the project 'Pacioni_et_al' and the ST scenario
# 'ST_Classic'. Store the output in the object 'woylie.st.classic' and save
# to disk
```

```
woylie.st.classic <- collate_dat('Pacioni_et_al', 3, scenario = 'ST_Classic',
                               dir_in = pac.dir, save2disk=TRUE)

## End(Not run)
```

collate_one_dat *Collate one local Vortex output file into a data.frame*

Description

collate_one_dat parses one Vortex .dat or .stdat file, and returns the data within as one data.frame.

Usage

```
collate_one_dat(filename, runs, verbose = FALSE)
```

Arguments

filename	The fully qualified filename of a Vortex .dat or .stdat file
runs	The number of simulation runs
verbose	Progress messages, default: FALSE

Value

A data.frame with data from one .dat or .stdat file and population/scenario names as factors

Examples

```
# Pacioni et al. example files. See ?pacioni for more details.
require(vortexRdata)
pac.dir <- system.file('extdata', 'pacioni', package='vortexRdata')
f <- file.path(pac.dir, 'Pacioni_et_al_ST_Classic(Base).stdat')
one.st.classic <- collate_one_dat(f, 3)
```

collate_proc_data *Collate processed data generated by any of the 'collate' functions*

Description

collate_proc_data collates multiple data frames generated by any of the 'collate' functions. This may be useful when, for example, data generated by different ST scenarios and/or 'standard' scenario runs are to be combined into a unique dataframe that can then be passed to other functions.

Usage

```
collate_proc_data(data, save2disk = TRUE, dir_out = "ProcessedData")
```

Arguments

data	A list where each element is a df from collate_[dat, yr, run]
save2disk	Whether to save the data as rda and csv, default: TRUE
dir_out	The local path to store the output. Default: ProcessedData

Details

Only dfs generated by the same function can be combined together. Missing data are filled with NA.

When save2disk=TRUE the output file will be named 'CombinedDB'. dir_out is created within the working directory unless a full path is provided.

Value

A data.frame with the collated data. Missing data are filled with NA.

Examples

```
# Using Campbell et al. example data. See ?sta.main, ?sta.evy5, ?sta.evy5.b11
# for more details.
data(sta.main, sta.evy5, sta.evy5.b11)
dfs <- list(sta.main, sta.evy5, sta.evy5.b11)
combined <- collate_proc_data(dfs, save2disk=FALSE)
```

collate_run	<i>Collate Vortex .run output files</i>
-------------	---

Description

collate_run collates all Vortex output files with extension .run matching the project and scenario name in a given directory into one named list.

Usage

```
collate_run(project, scenario, npops = 1, dir_in = NULL,
  save2disk = TRUE, dir_out = "ProcessedData", verbose = TRUE)
```

Arguments

project	The Vortex project name to be imported
scenario	The scenario name
npops	The total number of simulated populations including the metapopulation
dir_in	The local folder containing Vortex files, default: NULL. If not specified, will fall back to use current working directory.
save2disk	Whether to save the data as rda and csv, default: TRUE
dir_out	The local path to store the output. Default: ProcessedData
verbose	Progress messages, default: TRUE

Details

dir_in may contain other files; only files matching the project and the scenario name will be read.

dir_out is created within the working directory unless a full path is provided.

If no matching files are found in the given directory, an error is reported.

When verbose=TRUE the progress (i.e. the file being read) is reported on screen.

Value

a list with two elements: run, a data.frame with data from all Vortex files and lrund, where the same data are re-arranged in long format

Examples

```
# Using Pacioni et al. example files. See ?pacioni for more details.
pac.dir <- system.file('extdata', 'pacioni', package='vortexRdata')
# Run collate_run on all .run of the project 'Pacioni_et_al' and
run <- collate_run('Pacioni_et_al', 'ST_LHS', 1, dir_in=pac.dir,
                  save2disk=FALSE)
```

collate_yr

Collate Vortex .yr output files

Description

collate_yr collates all the .yr output from Vortex matching the project and scenario name into one R object and calculates the mean for each simulated year across all iterations.

Usage

```
collate_yr(project, scenario, npops_noMeta = 1, dir_in = NULL,
           save2disk = TRUE, dir_out = "ProcessedData", verbose = TRUE)
```

Arguments

project	The Vortex project name to be imported
scenario	The scenario name, default: NULL
npops_noMeta	The total number of populations excluding the metapopulation, default: 1
dir_in	The local folder containing Vortex files, default: NULL. If not specified, will fall back to use current working directory.
save2disk	Whether to save the data as rda and csv, default: TRUE
dir_out	The local path to store the output. Default: ProcessedData
verbose	Progress messages, default: TRUE

Details

dir_in may contain other files; only files matching the project and the scenario name will be read.

dir_out is created within the working directory unless a full path is provided.

If no matching files are found in the given directory, an error is reported.

When verbose=TRUE the progress (i.e. the file being read) is reported on screen.

Value

a list with two elements: 'census', a data.frame with data from all Vortex files and 'census_means', a data.table with the mean of each parameter across all iterations for each simulated year

Examples

```
# Using Pacioni et al. example files. See ?pacioni for more details.
pac.dir <- system.file('extdata', 'pacioni', package='vortexRdata')

# Run collate_yr on all .yr of project 'Pacioni_et_al' and the ST scenario
# 'ST_Classic' in the selected folder and store the result in 'yr.st.classic'
yr.st.classic <- collate_yr(project='Pacioni_et_al', scenario='ST_Classic',
                           dir_in = pac.dir, save2disk=FALSE)
```

 CompileIter

Compile iterations from one .yr file

Description

Compile iterations from one .yr file and add a column with scenario names and one with iteration number

Usage

```
CompileIter(iter, filename, n_rows, iter_ln, lines, header)
```

Arguments

iter	The iteration (run) number
filename	The fully qualified filename to read from
n_rows	The number of rows to read from the file
iter_ln	The number of rows to skip from the file
lines	An object returned from readLines()
header	A character vector of column names

Value

A data.frame

conv_l_yr	<i>Convert 'census' data into long format</i>
-----------	---

Description

conv_l_yr converts the first element of the output from collate_yr (census) in long format. This can be then fed into downstream analysis (e.g. fit_regression)

Usage

```
conv_l_yr(data, npops_noMeta = 1, appendMeta = FALSE, project,
  scenario, yrs = c(1, 2), save2disk = TRUE,
  dir_out = "ProcessedData")
```

Arguments

data	The df 'census' from collate_yr
npops_noMeta	The total number of populations excluding the metapopulation, default: 1
appendMeta	Whether to calculate data for the metapopulation, default: FALSE
project	Vortex project name (used to name the output)
scenario	Vortex scenario name (used to name the output)
yrs	The year(s) that need to be retained in the output
save2disk	Whether to save the data as rda and csv, default: TRUE
dir_out	The local path to store the output. Default: ProcessedData

Details

yrs is used to indicate the years to be retained in the output. If more than one year is required, these may be requested defining a numeric vector, e.g. yrs=c(10, 20, 30). All simulated years can be included in the output by passing a numeric vector with all years. For example, assuming that 100 years were simulated, using yrs=1:100 would retain all 100 years in the output.

dir_out is created within the working directory unless a full path is provided.

Value

The census data.frame in long format

Examples

```
# Using Pacioni et al. example data. See ?pac.yr for more details.
data(pac.yr)
lyr.classic <- conv_l_yr(pac.yr[[1]] , yrs=c(60, 120), save2disk=FALSE)
```

df2disk	<i>Save a data.frame as both Rdata and CSV</i>
---------	--

Description

df2disk saves to disk a given data.frame as both Rdata and CSV with a given name and optional name postfix to a given location.

Usage

```
df2disk(df, dirpath, fname, postfix = "", row_names = FALSE)
```

Arguments

df	A data.frame
dirpath	The destination path for written files, will be created if necessary
fname	The file name
postfix	An optional name postfix
row_names	Whether to include row names inthe csv file

Details

df2disk is used by the collate_ functions when the operator chooses to save2disk.

Examples

```
my.df <- data.frame(1, 1:10, sample(LETTERS[1:3], 10, replace = TRUE))
my.folder <- paste0(getwd(), '/test')
df2disk(df=my.df, dirpath=getwd(), fname='testname')
df2disk(df=my.df, dirpath=my.folder, fname='testname', postfix='_testpostfix')
```

dot_plot	<i>Dot plots of mean Vortex parameters</i>
----------	--

Description

dot_plot generates dot plots of mean parameter values for each population (row) at each year value requested with 'yrs' (columns). Bars represent standard deviation.

Usage

```
dot_plot(data, project, scenario, yrs = c(1, 2), params = c("PExtinct",
  "Nextant", "Het", "Nalleles"), setcolour = "scen.name",
  plotpops = c("all"), save2disk = TRUE, dir_out = "Plots")
```

Arguments

data	A df from collate_dat
project	Vortex project name (used to name the output)
scenario	Vortex scenario name (used to name the output)
yrs	The years to be included in the plot
params	Vortex parameters to be plotted, default: c('PExtinct', 'Nextant', 'Het', 'Nalleles')
setcolour	Variable to be used to set colours of data, default: scen.name
plotpops	The populations to be included in the plot, default: 'all'
save2disk	Whether to save the output to disk, default: TRUE
dir_out	The local path to store the output. Default: Plots

Details

Plots are ggplot objects. When save2disk=TRUE these are saved as .rda and .pdf files

yrs can be a numeric vector of length ≥ 1 (e.g. yrs=c(50,100)). Each point in time will be plotted in different columns.

If a continuous variable is passed to setcolour, a continuous gradient of colour will be assigned to the marker (e.g. for example, a scale from blue to black). If a sharp change of colours between different values of a continuous variable is desired, it has to be converted into a factor.

Value

Dot plots of mean parameter values with standard deviation

Examples

```
# Using Pacioni et al. example data. See ?pac.clas for more details.
data(pac.clas)
dot <- dot_plot(data=pac.clas, project='Pacioni_et_al', scenario='ST_Classic',
               yrs=c(80, 120),
               params=c('PExtinct', 'Nextant', 'Het', 'Nalleles'),
               save2disk=FALSE)
```

fit_regression

Search for the best regression model(s)

Description

fit_regression fits either a Generalized Linear Model or a betareg model to the data and search for the best model(s) given a list of predictors using the R package glmulti.

Usage

```
fit_regression(data, lookup = NULL, census = TRUE, yr, project,
  scenario, popn, param = "N", vs = c("GS1"),
  count_data = c("Nextant", "Nall", "Nalleles", "N", "AM", "AF",
  "Subadults", "Juv", "nDams", "nBroods", "nProgeny", "nImmigrants",
  "nEmigrants", "nHarvested", "nSupplemented", "YrExt", "Alleles"),
  ic = "aic", l = 1, ncand = 30, set_size = NA, save2disk = TRUE,
  dir_out = "DataAnalysis/FitRegression")
```

Arguments

data	The long format of census (from conv_1_yr) or run (lrn, the second element) of the output from collate_run
lookup	(Optional) A look-up table where the scenario names are listed together with the (missing) variables needed to fit the regression models
census	Whether the input is census data
yr	The year that has to be used in the analysis if census=TRUE
project	The Vortex project name
scenario	Vortex scenario name
popn	The sequential number of the population (in integer)
param	The dependent variable
vs	Character vector with independent variable(s)
count_data	Character vector with param(s) that are counts and would use a Poisson error distribution
ic	Information criterion
l	Level for glmulti search: 1 main effects, 2 main effects + interactions
ncand	The threshold of candidate models after which switch to the genetic search method, default: 30
set_size	Value to be used in confsetsize (from <code>glmulti</code> The number of models to be looked for, i.e. the size of the returned confidence set.)
save2disk	Whether to save the output to disk, default: TRUE
dir_out	The local path to store the output. Default: DataAnalysis/FitRegression

Details

`fit_regression` fits a different type of regression model depending on the dependent variable. When this is a count (e.g. N or the number of alleles), the function will fit a Generalized Linear Model. The first fit is attempted with a Poisson error distribution and if the dispersion parameter (calculated as $residualdeviance/df$ is larger than (the somewhat arbitrary cut off of) 1.5, the model will be refitted with a quasipoisson error distribution (a message is displayed if this happens).

`fit_regression` establishes whether the dependent variable is a count by searching for it in `count_data`. If the users generated their own dependent variable (e.g. through a PS), this has to be included in `count_data` to indicate `fit_regression` that it is analysing count data.

When the number of alleles is the dependent variable (from .run files), this is rounded to integer (to meet R requirement that count data are integers) before a GLM is fitted to the data.

If param is a proportion (e.g. Gene Diversity and Inbreeding), then the function uses a Beta regression from the R package betareg (Cribari-Neto & Zeileis 2010). Different link functions are tested and the one with the lowest AIC value is selected. The selected link function is displayed on the R console and the difference in the AIC scores relative to the best link function is also displayed.

In the initial fit of the model the main and interactions effects are included.

Successively, a search for the best model is carried out. This is performed with the R package `glmulti` (Calcagno & de Mazancourt 2010). `fit_regression` will conduct an exhaustive search if `ncand` is less or equal to the number of candidate models, otherwise it will use a genetic search method (see `glmulti` documentations for more details about the search methods). When `glmulti` uses the genetic search method, two small files (with extension `.modgen.back` and `.mods.back`. are written in the working directory even if `save2disk=FALSE`.

`fit_regression` explicitly ignores NA.

Depending on the data, fitting several Beta regression models to complete the search may be a long (and memory hungry) process. Also, the package `betareg` has the limitation (at least at the moment of writing) that cannot handle analysis of data when the dependent variable takes value of either exactly 0 or 1.

See vignette for a more detailed explanation of `fit_regression`.

Value

A `glmulti` object with the best models found.

References

Calcagno, V., and C. de Mazancourt. 2010. `glmulti`: an R package for easy automated model selection with (generalized) linear models. *Journal of Statistical Software* 34:1-29.

Cribari-Neto, F., and Zeileis, A. (2010) Beta regression in R. *Journal of Statistical Software* 34(2).

Examples

```
# Using Pacioni et al. example data. See ?pac.run.lhs and ?pac.lhs for more
# details.
data(pac.run.lhs, pac.lhs)

# Remove base scenario from .stdat data
pac.lhs.no.base <- pac.lhs[!pac.lhs$scen.name == 'ST_LHS(Base)', ]

# Use function lookup_table to obtain correct parameter values at year 0
lkup.ST_LHS <- lookup_table(data=pac.lhs.no.base, project='Pacioni_et_al',
                             scenario='ST_LHS',
                             pop='Population 1',
                             SVs=c('SV1', 'SV2', 'SV3', 'SV4', 'SV5', 'SV6', 'SV7'),
                             save2disk=FALSE)

# Remove base scenario from .run output in long format
lrun.ST_LHS.no.base <- pac.run.lhs[[2]][!pac.run.lhs[[2]]$Scenario == 'ST_LHS(Base)', ]
```

```

reg <- fit_regression(data=lrun.ST_LHS.no.base, lookup=lkup.ST_LHS,
                    census=FALSE,
                    project='Pacioni_et_al', scenario='ST_LHS', popn=1,
                    param='N', vs=c('SV1', 'SV2', 'SV3'), l=2, ncand=30,
                    save2disk=FALSE)

# Clean up of residual files written by glmulti
# Note, in some OS (W) these files may be locked because in use by R and have
# to be manually after the R session has been either terminated or restarted
file.remove(c('Pacioni_et_al_ST_LHS_N.modgen.back',
              'Pacioni_et_al_ST_LHS_N.mods.back'))

# Example of information you can obtained once you have run fit_regression

# The formula for the best model
bestmodel <- reg@formulas[1]

# The formulae for the best 30 model
bestmodels <- reg@formulas

# List of IC values
qaicvalues <- reg@crits

# QAIC differences between the first 5 best models (stored in 'delta')
delta <- as.vector(NULL)
for (i in 1:5) {
  del <- qaicvalues[i+1] - qaicvalues[i]
  delta <- c(delta, del)
}

# The best model's coefficients
coef.best <- coef(reg@objects[[1]])

# The model averaged coefficients
coef.all <- glmulti::coef.glmulti(reg)
coefs <- data.frame(Estimate=coef.all[,1],
                   Lower=coef.all[,1] - coef.all[,5],
                   Upper=coef.all[,1] + coef.all[,5])

# Plot IC profile
plot(reg, type='p')

# Plot of model averaged importance of terms
plot(reg, type='s')

```

Description

File names are sorted using `gtools::mixedsort()`.

Usage

```
get_file_paths(path, pattern, fn_name, fname, verbose = FALSE)
```

Arguments

path	The directory to search in
pattern	A pattern to match file names
fn_name	The vortexR function name for verbose messages
fname	Name of file name root
verbose	Progress messages, default: FALSE

Value

A character vector of fully qualified file paths

line_plot_year	<i>Line plots of Vortex parameters vs years</i>
----------------	---

Description

`line_plot_year` generates line plots of the selected Vortex parameters for the selected populations, for all simulated years.

Usage

```
line_plot_year(data, project, scenario, params = c("PExtinct", "Nextant",
  "Het", "Nalleles"), plotpops = c("all"), save2disk = TRUE,
  dir_out = "Plots")
```

Arguments

data	A df from <code>collate_dat</code>
project	Vortex project name (used to name the output)
scenario	Vortex scenario name (used to name the output)
params	Vortex parameters to be plotted, default: <code>c('PExtinct', 'Nextant', 'Het', 'Nalleles')</code>
plotpops	The populations to be included in the plot, default: <code>'all'</code>
save2disk	Whether to save the output to disk, default: TRUE
dir_out	The local path to store the output. Default: Plots

Details

Plots are ggplot objects. When `save2disk=TRUE` these are saved as `.rda` and `.pdf` files

Value

Line plot(s)

Examples

```
# Using Pacioni et al. example data. See ?pac.clas for more details.
data(pac.clas)
lineplot.st.classic <- line_plot_year(data=pac.clas, project='Pacioni_et_al',
                                     scenario='ST_Classic',
                                     params=c('PExtinct', 'Nextant', 'Het', 'Nalleles'),
                                     save2disk=FALSE)
```

line_plot_year_mid *Line plots of Vortex parameters vs years*

Description

`line_plot_year_mid` generates line plots of the selected Vortex parameters for the selected populations, from year zero to `yrmid`. The purpose of these plots is to 'zoom' in the initial phase of the simulations to better appreciate dynamics of the parameters of interest.

Usage

```
line_plot_year_mid(data, project, scenario, yrmid = 1,
                  params = c("PExtinct", "Nextant", "Het", "Nalleles"),
                  plotpops = c("all"), save2disk = TRUE, dir_out = "Plots")
```

Arguments

<code>data</code>	A df from <code>collate_dat</code>
<code>project</code>	Vortex project name (used to name the output)
<code>scenario</code>	Vortex scenario name (used to name the output)
<code>yrmid</code>	The last year to plot
<code>params</code>	Vortex parameters to be plotted, default: <code>c('PExtinct', 'Nextant', 'Het', 'Nalleles')</code>
<code>plotpops</code>	The populations to be included in the plot, default: <code>'all'</code>
<code>save2disk</code>	Whether to save the output to disk, default: <code>TRUE</code>
<code>dir_out</code>	The local path to store the output. Default: <code>Plots</code>

Details

Plots are ggplot objects. When `save2disk=TRUE` these are saved as `.rda` and `.pdf` files

Value

Line plot(s)

Examples

```
# Using Pacioni et al. example data. See ?pac.clas for more details.
data(pac.clas)
lineMidPlot.st.classic <- line_plot_year_mid(data=pac.clas,
      project='Pacioni_et_al',
      scenario='ST_Classic',
      yrmid=50,
      params=c('PExtinct', 'Nextant', 'Het', 'Nalleles'),
      save2disk=FALSE)
```

lookup_table

Summary table of simulation parameters

Description

lookup_table creates a table that summarises simulation parameters. The final table will have a line for each scenario and one column for each parameter requested with SVs.

Usage

```
lookup_table(data, project, scenario, pop = "Population 1",
  SVs = c("SV1"), save2disk = TRUE, dir_out = "ProcessedData")
```

Arguments

data	The output from collate_dat
project	Vortex project name (used to name the output)
scenario	Vortex scenario name (used to name the output)
pop	The name of the pop to be used as reference
SVs	The parameters to include in the table
save2disk	Whether to save the output to disk, default: TRUE
dir_out	The local path to store the output. Default: ProcessedData

Details

If the name of the populations were changed, the user has to indicate a population to be used as reference, otherwise lookup_table will look for a population named 'Population 1' (i.e. Vortex default name for the first population).

lookup_table reports the values of SVs at year zero. This is done because parameters may take value 'zero' if the relevant population goes extinct. There are cases where Vortex may not evaluate some parameters at year 0. This may happen, for example, when a population is empty at initialization (i.e. the initial population size is zero), or when K is set to zero at the beginning of the

simulation. The user should check the values reported and check the Vortex input files if these do not look correct.

SVs can be any variable included in the data, including GS or PS set up in Vortex.

Value

A data.frame with scenario names and parameter values

Examples

```
# Using Pacioni et al. example data. See ?pac.clas for more details.
data(pac.clas)
lkup.st.classic <- lookup_table(data=pac.clas, project='Pacioni_et_al',
                               scenario='ST_Classic', pop='Population 1',
                               SVs=c('SV1', 'SV2', 'SV3', 'SV4', 'SV5', 'SV6', 'SV7'),
                               save2disk=FALSE)
```

m_scatter	<i>Generates a matrix of scatter plots</i>
-----------	--

Description

m_scatter generates a matrix of pairwise scatter plots to graphically investigate possible associations between variables.

Usage

```
m_scatter(data, data_type = "dat", lookup = NULL, yr = 1, popn = 1,
          param = "N", vs = NA, save2disk = TRUE, fname = NULL,
          dir_out = "Plots")
```

Arguments

data	The output from collate_dat, the long format of the output from collate_run or the output from con_l_yr
data_type	The type of input data. Possible options are 'dat', 'yr' or 'run'
lookup	A table to add relevant variable matched using the scenarios names
yr	The year to be plotted
popn	The sequential number of the population (in integer)
param	The parameter to be plotted in the last row
vs	The parameters to be plotted
save2disk	Whether to save the output to disk, default: TRUE
fname	The name of the files where to save the output
dir_out	The local path to store the output. Default: Plots

Details

The output from `collate_dat` is the preferred input for this function as large datasets will require a long time to be plotted.

It may be convenient to pass the dependent variable of a regression model with `param` so that all the pairwise scatter plots of this variable will be in one line.

Value

A matrix of scatter plots

Examples

```
# Using Pacioni et al. example data. See ?pac.lhs for more details.
data(pac.lhs)
# Remove base scenario
pac.lhs.no.base <- pac.lhs[!pac.lhs$scen.name == 'ST_LHS(Base)', ]

# Get correct parameter values at year 0
lkup.ST_LHS <- lookup_table(
  data=pac.lhs.no.base, project='Pacioni_et_al',
  scenario='ST_LHS',
  pop='Population 1',
  SVs=c('SV1', 'SV2', 'SV3', 'SV4', 'SV5', 'SV6', 'SV7'),
  save2disk=FALSE)

scatter.plot <- m_scatter(
  data=pac.lhs.no.base[1:33],
  data_type='dat',
  lookup=lkup.ST_LHS,
  yr=120,
  popn=1,
  param='Nall',
  vs=c('SV1', 'SV2', 'SV3'),
  save2disk=FALSE)
```

Nadults

Calculate the harmonic mean of the total number of adults

Description

Nadults calculates, for several scenarios, the harmonic mean of the total number of adults between `yr0` and `yrt`. These can be use to calculate N_e/N ratios where relevant.

Usage

```
Nadults(data, scenarios = "all", npops_noMeta = 1,
  appendMeta = FALSE, gen = 1, yr0 = 1, yrt = 2,
  save2disk = TRUE, fname = "Nadults", dir_out = "DataAnalysis")
```

Arguments

<code>data</code>	The second element (<code>census_means</code>) of the output from <code>collate_yr</code>
<code>scenarios</code>	A vector of scenario names for which <code>Ne</code> needs to be calculated, default: <code>'all'</code>
<code>npops_noMeta</code>	The total number of populations excluding the metapopulation, default: 1
<code>appendMeta</code>	Whether to calculate data for the metapopulation, default: <code>FALSE</code>
<code>gen</code>	The generation time express in years
<code>yr0</code>	The time window to be considered (first and last year respectively)
<code>yrt</code>	The time window to be considered (first and last year respectively)
<code>save2disk</code>	Whether to save the output to disk, default: <code>TRUE</code>
<code>fname</code>	The name of the files where to save the output, default: <code>'Nadults'</code>
<code>dir_out</code>	The local path to store the output. Default: <code>DataAnalysis</code>

Details

`yrt` is adjusted by subtracting the number of years of the generation time (rounded to the nearest integer). In this way the user can provide the same `yr0`, `yrt` and `gen` to `Nadults` and `Ne` and these values are adjusted internally to correctly calculate the `Ne/N` ratios where relevant. If this behaviour is not desired, use `gen=0`.

Value

A `data.table` (`data.frame` if `data.table` is not loaded) with `Nb` values

Examples

```
# Using Pacioni et al. example data. See ?pac.yr for more details.
data(pac.yr)
NadultAll <- Nadults(data=pac.yr[[2]], scenarios='all', gen=2.54, yr0=50,
                    yrt=120, save2disk=FALSE)
```

<code>Ne</code>	<i>Calculate the effective population size (N_e)</i>
-----------------	---

Description

`Ne` calculates the effective population size (`Ne`) between `yr0` and `yrt` for several scenarios based on the loss of genetic diversity (expected heterozygosity) using the temporal approach.

Usage

```
Ne(data = NULL, scenarios = "all", gen = 1, yr0 = 1, yrt = 2,
   save2disk = TRUE, fname = "Ne", dir_out = "DataAnalysis")
```

Arguments

data	The output from collate_dat
scenarios	A vector of scenario names for which Ne needs to be calculated, default: 'all'
gen	The generation time express in years
yr0, yrt	The time window to be considered (first and last year respectively)
save2disk	Whether to save the output to disk, default: TRUE
fname	The name of the files where to save the output, default: 'Ne'
dir_out	The local path to store the output. Default: DataAnalysis

Details

yr0 is adjusted by adding the number of years of the generation time (rounded to the nearest integer). In this way the user can provide the same yr0,yrt and gen to Nadults and Ne and these values are adjusted internally to correctly calculate the Ne/N ratios where relevant. If this behaviour is not desired, use gen=0.

NOTE: When a population goes extinct, the results of the calculations are spurious (they are 0.5). This may change in future versions.

Value

A data.table (data.frame if data.table is not loaded) with Ne values

Examples

```
# Using Pacioni et al. example data. See ?pac.clas for more details.
data(pac.clas)
# Calculate Ne for all scenarios in the data. Note the odd value for scenario
# 12, consequent to the population going extinct.
NeAll <- Ne(data=pac.clas, scenarios='all', gen=2.54, yr0=50, yrt=120,
            save2disk=FALSE)
```

pac.clas

Collated results from Vortex scenarios - Pacioni et al. (2017)

Description

Subset (only 3 runs) of data from Pacioni et al. (2017) used to conduct a sensitivity analysis on demographic parameters. Vortex outputs, from the project named 'Pacioni_et_al' and (Single-Factor) sensitivity test scenario 'ST_Classic' (.stdat files), were collated with collate_dat.

Format

a data.frame of 2904 observations of 68 variables.

Source

Pacioni, C., Williams, M., Lacy RC, Spencer, P.B.S. and Wayne, A.F. (2017) Predators and genetic fitness: key threatening factors for the conservation of bettong species. Pacific Conservation Biology. DOI:10.1071/PC17002

Examples

```
data("pac.clas")
head(pac.clas)
```

pac.clas.lookup *Look-up table*

Description

Data from Pacioni et al. (2017) - sensitivity test scenario 'ST_Classic' - were used to generate a look-up table sizes using lookup_table.

Format

A data.frame with 24 observations of 8 variables.

Source

Pacioni, C., Williams, M., Lacy RC, Spencer, P.B.S. and Wayne, A.F. (2017) Predators and genetic fitness: key threatening factors for the conservation of bettong species. Pacific Conservation Biology. DOI:10.1071/PC17002

Examples

```
data("pac.clas.lookup")
head(pac.clas.lookup)
```

pac.clas.Nadults *Harmonic mean of adults and population sizes*

Description

Data from Pacioni et al. (2017) - sensitivity test scenario 'ST_Classic' - were used to calculate the harmonic mean of adults and population sizes using Nadults.

Format

A data.frame with 24 observations of 4 variables.

Source

Pacioni, C., Williams, M., Lacy RC, Spencer, P.B.S. and Wayne, A.F. (2017) Predators and genetic fitness: key threatening factors for the conservation of bettong species. Pacific Conservation Biology. DOI:10.1071/PC17002

Examples

```
data("pac.clas.Nadults")
head(pac.clas.Nadults)
```

pac.clas.Ne	<i>Effective population size</i>
-------------	----------------------------------

Description

Data from Pacioni et al. (2017) - sensitivity test scenario 'ST_Classic' - were used to calculate the effective population size sizes using Ne.

Format

A data.frame with 24 observations of 2 variables.

Source

Pacioni, C., Williams, M., Lacy RC, Spencer, P.B.S. and Wayne, A.F. (2017) Predators and genetic fitness: key threatening factors for the conservation of bettong species. Pacific Conservation Biology. DOI:10.1071/PC17002

Examples

```
data("pac.clas.Ne")
head(pac.clas.Ne)
```

pac.clas.pairw	<i>Results of pairwise comparisons of simulation scenarios</i>
----------------	--

Description

Results of pairwise comparisons of simulation scenarios included in the sensitivity test scenario 'ST_Classic' using pairwise.

Format

A named list of 12 elements. See documentation for details.

Source

Pacioni, C., Williams, M., Lacy RC, Spencer, P.B.S. and Wayne, A.F. (2017) Predators and genetic fitness: key threatening factors for the conservation of bettong species. Pacific Conservation Biology. DOI:10.1071/PC17002

Examples

```
data("pac.clas.pairw")
head(pac.clas.pairw)
```

pac.lhs

Collated results from Vortex scenarios - Pacioni et al. (2017)

Description

Data from Pacioni et al. (2017) used to conduct a sensitivity analysis on demographic parameters. Vortex outputs, from the project named 'Pacioni_et_al' and (Latin Hypercube Sampling) sensitivity test scenario 'ST_LHS' (.stdat files), were collated with collate_dat.

Format

A data.frame of 6171 observations of 68 variables.

Source

Pacioni, C., Williams, M., Lacy RC, Spencer, P.B.S. and Wayne, A.F. (2017) Predators and genetic fitness: key threatening factors for the conservation of bettong species. Pacific Conservation Biology. DOI:10.1071/PC17002

Examples

```
data("pac.lhs")
head(pac.lhs)
```

pac.run.lhs

Collated results from Vortex scenarios - Pacioni et al. (2017)

Description

Data from Pacioni et al. (2017) used to conduct a sensitivity analysis on demographic parameters. Vortex outputs, from the project named 'Pacioni_et_al' and (Latin Hypercube Sampling) sensitivity test scenario 'ST_LHS' (.run files), were collated with collate_run.

Format

A named list of two data.frames: run (153 obs, 7 var), lrun (153 obs, 8 var).

Source

Pacioni, C., Williams, M., Lacy RC, Spencer, P.B.S. and Wayne, A.F. (2017) Predators and genetic fitness: key threatening factors for the conservation of bettong species. Pacific Conservation Biology. DOI:10.1071/PC17002

Examples

```
data("pac.run.lhs")
head(pac.run.lhs)
```

pac.yr

Collated results from Vortex scenarios - Pacioni et al. (2017)

Description

Data from Pacioni et al. (2017) used to conduct a sensitivity analysis on demographic parameters. Vortex outputs, from the project named 'Pacioni_et_al' and (Single-Factor) sensitivity test scenario 'ST_Classic' (.yr files), were collated with collate_yr.

Format

A named list of two elements: all (8712 obs, 26 var), means (2904 obs, 25 var).

Source

Pacioni, C., Williams, M., Lacy RC, Spencer, P.B.S. and Wayne, A.F. (2017) Predators and genetic fitness: key threatening factors for the conservation of bettong species. Pacific Conservation Biology. DOI:10.1071/PC17002

Examples

```
data("pac.yr")
head(pac.yr)
```

pairwise

Pairwise comparisons and ranks of scenarios

Description

pairwise conducts pairwise comparisons against a baseline scenario using sensitivity coefficients and strictly standardised mean difference. It also ranks scenarios (and/or parameters when relevant) using these statistics. When yrs='max' (default), VortexR automatically sets yrs to the last year of the simulation.

Usage

```
pairwise(data, project, scenario, params = c("PExtinct", "Nextant",
      "Het", "Nalleles"), yrs = "max", ST = FALSE, type = NA,
      group.mean = FALSE, SVs = NA, save2disk = TRUE,
      dir_out = "DataAnalysis/Pairwise")
```

Arguments

<code>data</code>	A data.frame generated by <code>collate_dat</code>
<code>project</code>	The Vortex project name
<code>scenario</code>	The ST Vortex scenario name or the scenario that should be used as baseline if simulations were not conducted with the ST module
<code>params</code>	A character vector with the parameters to be compared, default: <code>c('PExtinct', 'Nextant', 'Het', 'Nalleles')</code>
<code>yrs</code>	The year(s) to be analysed, default: <code>'max'</code>
<code>ST</code>	Whether files are from sensitivity analysis (TRUE), or not (FALSE, default)
<code>type</code>	Type of ST. Possible options are: <code>'Sampled'</code> , <code>'Latin Hypercube Sampling'</code> , <code>'Factorial'</code> or <code>'Single-Factor'</code>
<code>group.mean</code>	Whether calculate the mean of the statistics (SSMD and Sensitivity Coefficient) by group. See details
<code>SVs</code>	A character vector with the parameters to be used to group scenarios, default: NA
<code>save2disk</code>	Whether to save the output to disk, default: TRUE
<code>dir_out</code>	The local path to store the output. Default: <code>DataAnalysis/Pairwise</code>

Details

Pairwise comparisons against a baseline scenario are conducted using sensitivity coefficients (SC, Drechsler et al. 1998) and strictly standardised mean difference (SSDM, Zhang 2007).

`pairwise` ranks, for each population, the scenarios (and SVs if relevant, see below) based on the absolute value of the statistics (either SC or SSMD) regardless of the sign. That is, the scenario with the absolute SC or SSMD value most different from zero will have a rank equal to '1'. The actual statistics need to be inspected to evaluate the direction of the change.

The Kendall's coefficient of concordance is calculated to test whether the order of ranked scenarios (or SVs if relevant) is statistically consistent across the chosen points in time and parameters (or SVs). For example, if 100 years were simulated, `yrs=c(50, 100)` and `params=c('Nall', 'Het')`, the consistency of ranking will be tested across the four raters (i.e. Nall at year 50, and at year 100, Het at year 50 and at year 100). Kendall's test operates a listwise deletion of missing data. However, when data in a whole column (i.e. ranks for a parameter) are missing, the column is removed before the statistic is calculated (See vignette for more information).

It is possible to evaluate the mean effect of a range of values for certain parameters on their outcome variables of interest (i.e. ranking the parameters, rather than scenarios). This is automatically done when the analysis is conducted on with `ST=TRUE`, `type='Single-Factor'` and there is more than one SV passed with the argument `SVs`. Alternatively, it is achievable with a combined use of

`group.mean=TRUE`, SVs. The first argument result in the calculations, following Conroy and Brook (2003), of the mean SC and SSMD for each group of scenarios that have different parameter values. SVs provides the names of the parameters to be considered. Parameters are then ranked accordingly (See vignette for more information).

The parameter values passed with SVs are evaluated at year=0. This is done because these parameters may take value 'zero' if the relevant populations goes extinct. There are cases where Vortex may not evaluate these parameters even at year 0. This may happen, for example, when a population is empty at initialization (i.e. the initial population size is zero), or when K is set to zero at the beginning of the simulation. The user has to make sure that the values for the parameters passed in are correct.

Note that it only makes sense to rank parameters in a ST run when the Single-Factor option is used in Vortex. This is because with Single-Factor, the parameters are modified one at the time (See vignette for more information).

Value

A list of six elements:

- A data.frame with SC values for all scenarios
- A data.frame with SSMD values
- A data.frame with p-values for SSMD values
- A data.frame with the scenario ranks based on SC and one based on SSMD
- The output of the Kendall's test

If `group.mean=TRUE` there will be six additional elements:

- A data.frame with the mean SC values for each parameter
- A data.frame with the mean SSMD values
- A data.frame with p-values calculated for the mean SSMD values
- A data.frame with the parameter ranks based on the mean SC and one based on the mean SSMD
- The output of the Kendall's test performed on the ranking of the parameters

References

- Conroy, S. D. S., and B. W. Brook. 2003. Demographic sensitivity and persistence of the threatened white- and orange-bellied frogs of Western Australia. *Population Ecology* 45:105-114.
- Drechsler, M., M. A. Burgman, and P. W. Menkhorst. 1998. Uncertainty in population dynamics and its consequences for the management of the orange-bellied parrot *Neophema chrysogaster*. *Biological Conservation* 84:269-281.
- Zhang, X. D. 2007. A pair of new statistical parameters for quality control in RNA interference high-throughput screening assays. *Genomics* 89:552-561.

Examples

```
# Using Pacioni et al. example data. See ?pac.clas for more details.
data(pac.clas)
pairw<-pairwise(data=pac.clas, project='Pacioni_et_al', scenario='ST_Classic',
               params=c('Nall', 'Het'), yrs=c(60,120), ST=TRUE,
               type='Single-Factor',
               SVs=c('SV1', 'SV2', 'SV3', 'SV4', 'SV5', 'SV6', 'SV7'),
               save2disk=FALSE)
```

Pextinct

*Cumulative probability of extinction at the end of the simulation***Description**

Pextinct calculates the cumulative probability of extinction at the by calculating the proportion of runs in which a population goes extinct for each scenario.

Usage

```
Pextinct(data, project, scenario, ST = FALSE, save2disk = TRUE,
         dir_out = "DataAnalysis/Pextinct")
```

Arguments

data	The long format of run (lrun, the second element) of the output from collate_run
project	The Vortex project name
scenario	The ST Vortex scenario name or the scenario that should be used as baseline if simulations were not conducted with the ST module
ST	Whether files are from sensitivity analysis (TRUE), or not (FALSE, default)
save2disk	Whether to save the output to disk, default: TRUE
dir_out	The local path to store the output. Default: DataAnalysis/Pairwise

Details

Pextinct then compares each scenario by calculating the strictly standardised mean difference (SSMD, Zhang 2007) and reports this statistic with its associated p values. Raw data are also reported.

Value

A list with two elements, a table (data.table) with the mean Probability of extinction and its SD, the SSMD and its associated p-value for each scenario and population, and a table (data.table) with each iteration where extinction is coded as one (and zero for no extinction)

References

Zhang, X. D. 2007. A pair of new statistical parameters for quality control in RNA interference high-throughput screening assays. *Genomics* 89:552-561.

Examples

```
# Using Pacioni et al. example data. See ?pac.run.lhs for more details.
data(pac.run.lhs)
Pext <- Pextinct(pac.run.lhs[[2]], project='Pacioni_et_al',
  scenario='ST_Classic', ST=TRUE, save2disk=FALSE,
  dir_out='DataAnalysis/Pextinct')
```

PrefixAndRepeat	<i>Return a prefixed and repeated string of character</i>
-----------------	---

Description

Return a prefixed and repeated string of character

Usage

```
PrefixAndRepeat(chars, times = 1, prefix = "")
```

Arguments

chars	A string of characters (popvalue)
times	The number of repetitions (ncolpop), default: 1
prefix	A text prefix, default: ""

pval	<i>Calculates p-values from z-values</i>
------	--

Description

pval calculates one-tailed p values from a vector that contains z-values and it is generally used internally.

Usage

```
pval(x)
```

Arguments

x	z-values
---	----------

Value

A numeric vector of length equal to length(x)

Examples

```
z <- c(1.645, 1.96, 3.09)
pval(z)
```

rRec	<i>Calculate the mean recovery rate (Pacioni et al 2017) and compare scenarios</i>
------	--

Description

rRec calculates the mean and standard deviation growth rate between the time yr0 and yrt, which was defined as 'recovery rate' by Pacioni et al (in press). The function then calculates the strictly standardised mean difference (SSMD, Zhang 2007) for each scenario, and each population contained in the data. rRec uses this statistic to compare each scenario (providing associated p-values) with a baseline scenario.

Usage

```
rRec(data, project, scenario, ST = FALSE, runs, yr0 = 1, yrt,
      save2disk = TRUE, dir_out = "DataAnalysis/rRec")
```

Arguments

data	A data.frame generated by collate_dat
project	The Vortex project name
scenario	The ST Vortex scenario name or the scenario that should be used as baseline if simulations were not conducted with the ST module
ST	Whether files are from sensitivity analysis (TRUE), or not (FALSE, default)
runs	The number of Vortex simulation runs
yr0	The time window to be considered (first and last year respectively)
yrt	The time window to be considered (first and last year respectively)
save2disk	Whether to save the output to disk, default: TRUE
dir_out	The local path to store the output. Default: DataAnalysis/Pairwise

Details

The means and standard deviations are calculated as: ****check how to insert formula in Roxy tags****
 $rRec = \frac{\sigma(N_i * M_i)}{\sigma(N_i)} \frac{(N_1 * M_1 + N_2 * M_2 + N_3 * M_3)}{(N_1 + N_2 + N_3)}$ $SD = \frac{N_1 * S_1 + N_2 * S_2 + N_3 * S_3}{(N_1 + N_2 + N_3)}$

Where M is the mean growth rate in each year, N is the sample size (number of simulation runs) and S is the standard deviation.

The baseline scenario is selected with the argument scenario. However, if the simulations were part of a sensitivity testing (as indicated by ST) then the baseline scenario is selected using the scenario with the suffix '(Base)'.

Value

A table (`data.table`) with the mean `rRec` and its `SD`, the `SSMD` and its associated `p-value` for each scenario and population

References

Zhang, X. D. 2007. A pair of new statistical parameters for quality control in RNA interference high-throughput screening assays. *Genomics* 89:552-561.

Pacioni, C., and Mayer, F. (2017). `vortexR`: an R package for post Vortex simulation analysis.

Examples

```
# Using Pacioni et al. example data. See ?pac.clas for more details.
data(pac.clas)
recov <- rRec(pac.clas, project='Pacioni_et_al', scenario='ST_Classic',
             ST=TRUE, runs=3, yr0=1, yrt=120, save2disk=FALSE,
             dir_out='DataAnalysis/rRec')
```

 se2sd

Standard Error from a vector

Description

Standard Error from a vector

Usage

```
se2sd(se, no)
```

Arguments

<code>se</code>	A standard error of a set of values
<code>no</code>	The number of values

Value

The standard deviation of the values

SSMD_matrix

*Generate a SSMD matrix with all possible pairwise comparisons***Description**

SSMD_matrix conducts pairwise comparisons for all possible pairs using strictly standardised mean difference (SSDM, Zhang 2007).

Usage

```
SSMD_matrix(data, project, scenario, params = c("PExtinct", "Nextant",
        "Het", "Nalleles"), yrs = "max", ST = FALSE, save2disk = TRUE,
        dir_out = "DataAnalysis/SSMD_matrix")
```

Arguments

data	A data.frame generated by collate_dat
project	The Vortex project name
scenario	The ST Vortex scenario name or the scenario that should be used as baseline if simulations were not conducted with the ST module
params	A character vector with the parameters to be compared, default: c('PExtinct', 'Nextant', 'Het', 'Nalleles')
yrs	The year(s) to be analysed, default: 'max'
ST	Whether files are from sensitivity analysis (TRUE), or not (FALSE, default)
save2disk	Whether to save the output to disk, default: TRUE
dir_out	The local path to store the output. Default: DataAnalysis/SSMD_matrix

Details

When yrs= 'max' (default), VortexR automatically sets yrs to the last year of the simulation .

Value

A list where each element is a matrix of SSMD (belowe the diagonal) and related p-values (above the diagonal) for each combination of 'yrs', population and 'params'

References

Zhang, X. D. 2007. A pair of new statistical parameters for quality control in RNA interference high-throughput screening assays. *Genomics* 89:552-561.

Examples

```
# Using Campbell et al. and Pacioni et al. example data.
# See ?pacioni and ?campbell for more details on example data.
require(vortexRdata)

data("pac.clas")

SSMD_matrix(data=pac.clas, project="Pacioni_et_al",
             scenario="ST_Classic",
             params = c("PExtinct", "Nextant", "Het", "Nalleles"),
             yrs = c(60, 120), ST = FALSE, save2disk = FALSE)

data(sta.main)
ssmd_mat <- SSMD_matrix(data=sta.main, project="test",
                        scenario="test",
                        params = c("PExtant", "Nextant"),
                        yrs = c(25, 50), ST = FALSE, save2disk = FALSE)
```

sta.evy5

Collated results from Vortex scenarios - Campbell et al (2016)

Description

A dataset with the results from Vortex scenarios used in Campbell et al (2016) to simulate major application of control measures in every 5 year cycle. Vortex outputs, from the project named 'Starlingv3PopBased' and the sensitivity test scenario 'MReductEvy5' (.stdat files), were collated with `collate_dat`.

Format

a `data.frame` with 1020 observations of 47 variables.

Source

Campbell et al. (2016). Assessing the economic benefits of starling detection and control to Western Australia. *Australasian Journal of Environmental Management*, 23, 81-99. DOI:[10.1080/14486563.2015.1028486](https://doi.org/10.1080/14486563.2015.1028486)

Examples

```
data("sta.evy5")
head(sta.evy5)
```

`sta.evy5.b11`*Collated results from Vortex scenarios - Campbell et al (2016)*

Description

A dataset with the results from Vortex scenarios used in Campbell et al (2016) to simulate major application of control measures in every 5 year cycle, maintaining 2011 levels of investment. Vortex outputs, from the project named 'Starlingv3PopBased' and the sensitivity test scenario 'MReduction_B11_09Evy5' (.stdat files), were collated with `collate_dat`.

Format

a `data.frame` with 1020 observations of 47 variables.

Source

Campbell et al. (2016). Assessing the economic benefits of starling detection and control to Western Australia. *Australasian Journal of Environmental Management*, 23, 81-99. DOI:[10.1080/14486563.2015.1028486](https://doi.org/10.1080/14486563.2015.1028486)

Examples

```
data("sta.evy5.b11")
head(sta.evy5.b11)
```

`sta.main`*Collated results from Vortex scenarios - Campbell et al (2016)*

Description

A dataset with the results from the main Vortex scenarios used in Campbell et al (2016). Vortex outputs, from the project named 'Starlingv3PopBased' (.dat files), were collated with `collate_dat`.

Format

a `data.frame` with 1632 observations of 44 variables.

Source

Campbell et al. (2016). Assessing the economic benefits of starling detection and control to Western Australia. *Australasian Journal of Environmental Management*, 23, 81-99. DOI:[10.1080/14486563.2015.1028486](https://doi.org/10.1080/14486563.2015.1028486)

Examples

```
data("sta.main")
head(sta.main)
```

vortexR

vortexR: an R package for Post Vortex Simulation Analysis

Description

vortexR facilitates Post Vortex Simulation Analysis (PVSA) by offering tools to collate multiple Vortex (v10) output files into one R object, generate plots and conduct basic analysis (e.g. pairwise comparisons of scenarios) and more advanced statistics such as fitting of a Generalised Linear Model (GLM) to investigate the main and the interaction effects of the variables of interest.

Details

vortexR has a number of functions that are useful during the development of a Vortex project and for its analysis after completion. vortexR makes it easy to automatise the creation of plots and computation of basic statistics to inspect the effect of changes carried out in the Vortex project. Once the project development is completed, the same framework used in vortexR during the development of the project can be refined and extended to include more advanced statistical analyses or can be easily included in Markdown documents for the creation of reports (by converting them into pdf) or update web-pages.

The use of vortexR ensures reproducibility and standardises analytical approaches in population viability analysis.

Documentations

Use `help(package = 'vortexR')` for a list of vortexR functions and their specific documentations.

A more detailed description of the package and functions can be opened with: `vignette(package='vortexR', topic='Use`

More vignettes may be come available in the future. Use `vignette(package='vortexR')` to see all the available vignettes.

Citation

If you use vortexR, please cite: Pacioni, C., and Mayer, F. (2017). vortexR: an R package for post Vortex simulation analysis. *Methods in Ecology and Evolution*.

Get in touch

Please, use <https://github.com/carlopacioni/vortexR/issues> to report any issues with vortexR. If unsure, or for feedback, contact me at: carlo.pacioni 'at' gmail.com.

Publications

Below there are listed a few publications that used vortexR.

Campbell et al. (2016). Assessing the economic benefits of starling detection and control to Western Australia. *Australasian Journal of Environmental Management*, 23, 81-99. DOI:10.1080/14486563.2015.1028486

Pacioni, C., Williams, M., Lacy RC, Spencer, P.B.S. and Wayne, A.F. (2017) Predators and genetic fitness: key threatening factors for the conservation of bettong species. Pacific Conservation Biology. DOI:10.1071/PC17002

Index

collate_dat, [2](#)
collate_one_dat, [4](#)
collate_proc_data, [4](#)
collate_run, [5](#)
collate_yr, [6](#)
CompileIter, [7](#)
conv_l_yr, [8](#)

data.table, [19](#), [20](#)
df2disk, [9](#)
dot_plot, [9](#)

fit_regression, [10](#)

get_file_paths, [13](#)
glmulti, [11](#), [12](#)

line_plot_year, [14](#)
line_plot_year_mid, [15](#)
lookup_table, [16](#)

m_scatter, [17](#)

Nadults, [18](#)
Ne, [19](#)

pac.clas, [20](#)
pac.clas.lookup, [21](#)
pac.clas.Nadults, [21](#)
pac.clas.Ne, [22](#)
pac.clas.pairw, [22](#)
pac.lhs, [23](#)
pac.run.lhs, [23](#)
pac.yr, [24](#)
pairwise, [24](#)
Pextinct, [27](#)
PrefixAndRepeat, [28](#)
pval, [28](#)

rRec, [29](#)

se2sd, [30](#)

SSMD_matrix, [31](#)
sta.evy5, [32](#)
sta.evy5.b11, [33](#)
sta.main, [33](#)

vortexR, [34](#)
vortexR-package (vortexR), [34](#)