

Package ‘dynRB’

July 13, 2018

Type Package

Title Dynamic Range Boxes

Version 0.13

Date 2018-07-13

Author Manuela Schreyer <manuelalarissa.schreyer@sbg.ac.at>,
Wolfgang Trutschnig <Wolfgang.Trutschnig@sbg.ac.at>,
Robert R. Junker <Robert.Junker@sbg.ac.at>,
Jonas Kuppler <Jonas.Kuppler@uni-duesseldorf.de>,
Arne Bathke <Arne.Bathke@sbg.ac.at>

Maintainer Manuela Schreyer <manuelalarissa.schreyer@sbg.ac.at>

Description Improves the concept of multivariate range boxes, which is highly susceptible for outliers and does not consider the distribution of the data. The package uses dynamic range boxes to overcome these problems.

Imports corrplot, RColorBrewer, ggplot2, reshape2, vegan

Suggests testthat, knitr

VignetteBuilder knitr

License GPL-2

NeedsCompilation no

Repository CRAN

Date/Publication 2018-07-13 13:40:03 UTC

R topics documented:

dynRB-package	2
dynRB_Pn	3
dynRB_Vn	4
dynRB_VPa	5
finch	6
finch2	7
overview	8

Index	10
--------------	-----------

dynRB-package

Dynamic Range Boxes

Description

The package DynRB improves the concept of multivariate range boxes, which is highly susceptible for outlines and does not consider the distribution of the data. The package uses dynamic range boxes to overcome these problems.

Details

Package: dynRB
Type: Package
Version: 0.13
Date: 2018-07-13

Note

A tutorial that introduces the functions implemented in the package, explains the outputs, gives examples on how to use the output for follow-up analyses, and finally shows how to visualize the results can be found here: [Tutorial dynRB](#)

Author(s)

Manuela Schreyer <manuelalarissa.schreyer@sbg.ac.at>,
Wolfgang Trutschnig <Wolfgang.Trutschnig@sbg.ac.at>,
Robert R. Junker <Robert.Junker@sbg.ac.at> (corresponding author),
Jonas Kuppler <Jonas.Kuppler@uni-duesseldorf.de>,
Arne Bathke <Arne.Bathke@sbg.ac.at>

References

Junker RR, Kuppler J, Bathke AC, Schreyer ML, Trutschnig W (2016) Dynamic range boxes - A robust non-parametric approach to quantify size and overlap of n-dimensional hypervolumes. *Methods in Ecology and Evolution* doi: 10.1111/2041-210X.12611

Examples

```
# example function dynRB_VPa
# for reliable results use steps = 201
data(finch2)
r<-dynRB_VPa(finch2, steps = 101)
r$result
```

dynRB_Pn	<i>Pairwise overlaps for each dimension</i>
----------	---

Description

Function returns pairwise overlaps for each dimension n . Number of dynamic range boxes (steps) can be adjusted. Default: steps = 201

Usage

```
dynRB_Pn(A = A, steps = 201, correlogram = FALSE, row_col = c(2, 2))
```

Arguments

A	Data frame, where the first column is a character vector containing the objects (e.g. species) and the other columns are numeric vectors (containing measurements).
steps	Number of range boxes. Default: steps = 201
correlogram	If TRUE, the correlogram for each species is shown. If FALSE, no correlogram is shown. Default: correlogram = FALSE
row_col	Number of rows and columns of the figures (correlogram for each species). Default: row_col = c(2, 2)

Value

Data frame containing the summarized overlaps for each pair of objects and dimension.

Note

A tutorial that introduces the functions implemented in the package, explains the outputs, gives examples on how to use the output for follow-up analyses, and finally shows how to visualize the results can be found here: [Tutorial dynRB](#)

Author(s)

Manuela Schreyer <manuelalarissa.schreyer@sbg.ac.at>,
Wolfgang Trutschnig <Wolfgang.Trutschnig@sbg.ac.at>,
Robert R. Junker <Robert.Junker@sbg.ac.at> (corresponding author),
Jonas Kuppler <Jonas.Kuppler@uni-duesseldorf.de>,
Arne Bathke <Arne.Bathke@sbg.ac.at>

References

Junker RR, Kuppler J, Bathke AC, Schreyer ML, Trutschnig W (2016) Dynamic range boxes - A robust non-parametric approach to quantify size and overlap of n-dimensional hypervolumes. *Methods in Ecology and Evolution* doi: 10.1111/2041-210X.12611

Examples

```
# example function dynRB_Pn
# for reliable results use steps = 201
data(finch2)
r<-dynRB_Pn(finch2, steps = 101)
```

dynRB_Vn	<i>Relative Dynamic Range Box size per dimension and object</i>
----------	---

Description

Function returns Dynamic Range Box size of each dimension n . Number of dynamic range boxes (steps) can be adjusted. Default: steps = 201

Usage

```
dynRB_Vn(A = A, steps = 201, correlogram = FALSE, row_col = c(2, 2))
```

Arguments

A	Data frame, where the first column is a character vector and the other columns are numeric vectors.
steps	Number of range boxes. Default: steps = 201
correlogram	If TRUE, the correlogram for each species is shown. If FALSE, no correlogram is shown. Default: correlogram = FALSE
row_col	Number of rows and columns of the figures (correlogram for each species). Default: row_col = c(2, 2)

Value

Data frame containing the summarized niche length for each object and dimension.

Note

A tutorial that introduces the functions implemented in the package, explains the outputs, gives examples on how to use the output for follow-up analyses, and finally shows how to visualize the results can be found here: [Tutorial dynRB](#)

Author(s)

Manuela Schreyer <manuelalarissa.schreyer@sbg.ac.at>,
 Wolfgang Trutschnig <Wolfgang.Trutschnig@sbg.ac.at>,
 Robert R. Junker <Robert.Junker@sbg.ac.at> (corresponding author),
 Jonas Kuppler <Jonas.Kuppler@uni-duesseldorf.de>,
 Arne Bathke <Arne.Bathke@sbg.ac.at>

References

Junker RR, Kuppler J, Bathke AC, Schreyer ML, Trutschnig W (2016) Dynamic range boxes - A robust non-parametric approach to quantify size and overlap of n-dimensional hypervolumes. *Methods in Ecology and Evolution* doi: 10.1111/2041-210X.12611

Examples

```
# example function dynRB_Vn
# for reliable results use steps = 201
data(finch2)
r<-dynRB_Vn(finch2, steps = 101)
```

dynRB_VPa	<i>Size and pairwise overlap</i>
-----------	----------------------------------

Description

Function returns size and pairwise overlaps of niches or trait-spaces. Size or overlaps of dimensions can be aggregated by using either "product", "mean" or "geometric mean" as aggregation method. The results obtained by using the product are automatically printed. Number of dynamic range boxes (steps) can be adjusted. Default: steps = 201

Usage

```
dynRB_VPa(A = A, steps = 201, correlogram = FALSE, row_col = c(2, 2),
          pca.corr = FALSE, var.thres = 0.9)
```

Arguments

A	Data frame, where the first column is a character vector and the other columns are numeric vectors.
steps	Number of range boxes. Default: steps = 201
correlogram	If TRUE, the correlogram for each species is shown. If FALSE, no correlogram is shown. Default: correlogram = FALSE
row_col	Number of rows and columns of the figures (correlogram for each species). Default: row_col = c(2, 2)
pca.corr	If TRUE, a principal components analysis is performed.
var.thres	Variance predicted by the PCA-axes, if pca.corr = TRUE.

Value

Data frame containing the summarized niche overlap (and volume) for each pair of objects aggregated by all three possible choices (i.e. product, mean, geometric mean).

Note

A tutorial that introduces the functions implemented in the package, explains the outputs, gives examples on how to use the output for follow-up analyses, and finally shows how to visualize the results can be found here: [Tutorial dynRB](#)

Author(s)

Manuela Schreyer <manuelalarissa.schreyer@sbg.ac.at>,
Wolfgang Trutschnig <Wolfgang.Trutschnig@sbg.ac.at>,
Robert R. Junker <Robert.Junker@sbg.ac.at> (corresponding author),
Jonas Kuppler <Jonas.Kuppler@uni-duesseldorf.de>,
Arne Bathke <Arne.Bathke@sbg.ac.at>

References

Junker RR, Kuppler J, Bathke AC, Schreyer ML, Trutschnig W (2016) Dynamic range boxes - A robust non-parametric approach to quantify size and overlap of n-dimensional hypervolumes. *Methods in Ecology and Evolution* doi: 10.1111/2041-210X.12611

Examples

```
# example function dynRB_VPa
# for reliable results use steps = 201
data(finch2)
r<-dynRB_VPa(finch2, steps = 101, correlogram = TRUE, row_col = c(1,1))
r$result
```

finch

Data set finch

Description

To demonstrate the application of the functions for real world data, we used existing data sets on niches and trait-spaces and quantified their sizes and overlaps. The data set finch is a data set on morphological measurements of Darwin finches. The data set comprises quantitative measurements of nine traits characterizing five species of finches, each trait was measured at least in 10 individuals per species.

Usage

```
data("finch")
```

Format

A data frame with 146 observations on the following 10 variables.

Species a character vector of the Species *Geospiza heliobates*, *Geospiza prothemelas prothemelas*, *Geospiza fuliginosa parvula*, *Geospiza fortis fortis* and *Geospiza fortis platyrhyncha*

BodyL a numeric vector

WingL a numeric vector

TailL a numeric vector

BeakW a numeric vector

BeakH a numeric vector

LBeakL a numeric vector

UBeakL a numeric vector

N.UBkL a numeric vector

TarsusL a numeric vector

Note

A tutorial that introduces the functions implemented in the package, explains the outputs, gives examples on how to use the output for follow-up analyses, and finally shows how to visualize the results can be found here: [Tutorial dynRB](#)

Source

Snodgrass R and Heller E (1904) Papers from the Hopkins-Stanford Galapagos Expedition, 1898-99. XVI. Birds. Proceedings of the Washington Academy of Sciences 5: 231-372.

Examples

```
data(finch)
## quick overview
head(finch)
```

finch2

Subset of data set finch

Description

To demonstrate the application of the functions for real world data, we used existing data sets on niches and trait-spaces and quantified their sizes and overlaps. The data set `finch2` is a data set on morphological measurements of three Darwin finches. The data set comprises quantitative measurements of nine traits characterizing two species of finches, each trait was measured at least in 10 individuals per species.

Usage

```
data("finch2")
```

Format

A data frame with 103 observations on the following 10 variables.

Species a character vector of the Species *Geospiza fuliginosa parvula* and *Geospiza fortis fortis*

BodyL a numeric vector

WingL a numeric vector

Taill a numeric vector

BeakW a numeric vector

BeakH a numeric vector

LBeakL a numeric vector

UBeakL a numeric vector

N.UBkL a numeric vector

TarsusL a numeric vector

Note

A tutorial that introduces the functions implemented in the package, explains the outputs, gives examples on how to use the output for follow-up analyses, and finally shows how to visualize the results can be found here: [Tutorial dynRB](#)

Source

Snodgrass R and Heller E (1904) Papers from the Hopkins-Stanford Galapagos Expedition, 1898-99. XVI. Birds. Proceedings of the Washington Academy of Sciences 5: 231-372.

Examples

```
data(finch2)
## quick overview
head(finch2)
```

overview

Overview function

Description

This functions can be used to show the graphics generated by the functions `dynRB_Pn`, `dynRB_Vn` and `dynRB_VPa`.

Usage

```
overview(r, row_col = c(3, 3))
```


Arguments

`r` Output of the function `dynRB_Pn`, `dynRB_Vn` or `dynRB_VPa`.
`row_col` Number of rows and columns of the figures. Default: `row_col = c(3, 3)`

Note

A tutorial that introduces the functions implemented in the package, explains the outputs, gives examples on how to use the output for follow-up analyses, and finally shows how to visualize the results can be found here: [Tutorial dynRB](#)

Author(s)

Manuela Schreyer <manuelalarissa.schreyer@sbg.ac.at>,
Wolfgang Trutschnig <Wolfgang.Trutschnig@sbg.ac.at>,
Robert R. Junker <Robert.Junker@sbg.ac.at> (corresponding author),
Jonas Kuppler <Jonas.Kuppler@uni-duesseldorf.de>,
Arne Bathke <Arne.Bathke@sbg.ac.at>

Examples

```
# example for the function dynRB_Pn
# for reliable results use steps = 201
data(finch2)
r<-dynRB_Pn(finch2, steps = 101)
overview(r)
```

Index

*Topic **\textasciitildekwd1**

dynRB_Pn, [3](#)
dynRB_Vn, [4](#)
dynRB_VPa, [5](#)
overview, [8](#)

*Topic **\textasciitildekwd2**

dynRB_Pn, [3](#)
dynRB_Vn, [4](#)
dynRB_VPa, [5](#)
overview, [8](#)

*Topic **datasets**

finch, [6](#)
finch2, [7](#)

*Topic **package**

dynRB-package, [2](#)

dynRB (dynRB-package), [2](#)

dynRB-package, [2](#)

dynRB_Pn, [3](#)

dynRB_Vn, [4](#)

dynRB_VPa, [5](#)

finch, [6](#)

finch2, [7](#)

overview, [8](#)