Package: twinspan (via r-universe)

August 24, 2024

Title Two-Way Indicator Species Analysis

Version 0.9-3

Depends R (>= 3.5.0)

Suggests knitr, markdown

VignetteBuilder utils, knitr

Description Classification of biological communities based on splitting first axis of Correspondence Analysis for the current subset of the data, and finding species that best indicate the splits. The method is particularly popular in vegetation science.

Imports vegan

License MIT + file LICENCE

Encoding UTF-8

RoxygenNote 7.1.2

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

RemoteUrl https://github.com/jarioksa/twinspan

RemoteRef HEAD

RemoteSha 4760e1bba217994d7f5dfe043cc1225857e0a410

Contents

---

ahti   *Vegetation of Lichen-rich Pine Forests in Finland*

Description

Vegetation data set of 170 quadrats and 115 species with visually estimated cover percentage values (Oksanen & Ahti, 1982).

Usage

```
data("ahti")
```

Details

The taxon names are given with 4+4 acronyms of the binomial name. The complete scientific names are listed in [vignette](#) "ahtinames"; perhaps the easiest way of reading the vignette is to use [browseVignettes](#) (but see Examples below). Only understorey plants are included in the current file: trees and shrubs, also as seedlings, are not included, although they are listed in the original article.

Oksanen & Ahti (1982) allocated the quadrats into vegetation classes that they called 'noda'. The name of quadrat is based on the original nodum followed by the original number of the quadrat. In northern Finland (nortern boreal zone, roughly corresponding to the are of intensive reindeer grazing) five noda were separated: Ster for *Stereocaulon*, Vuli for *Vaccinium uliginosum*, Vmyr for *Vaccinium myrtillus*, Call for *Calluna*, and Empe for *Empetrum*. Most Middle boreal, south boreal and hemiboreal stands were allocated into *Cetraria islandica* nodum, but tabulated separately for young and old forest stands and by vegetation zone: middle boreal (names starting with M) and southern boreal (names starting with S) zones divided into young (Myng, Syng) and old (Mold, Sold) forests. Hemiboreal forests (Hbor) were not divided by age classes. The only separate southern boreal type was *Thymus serpyllum* nodum (Thym) on the sunny slopes of eskers.

Source

Oksanen & Ahti (1982).

References

Oksanen, J. & Ahti, T. (1982) Lichen-rich pine forest vegetation in Finland. *Annales Botanici Fennici* **19,** 275–301.

Examples

```
## Read the vignette listing complete scientific names
if (interactive()) {
vignette("ahtinames", package="twinspan")
}
```

---

as.dendrogram.twinspan

Extract Species or Quadrat Dendrograms

Description

Function extracts the species or quadrat classification as a hierarchic dendrogram.

Usage

```
## S3 method for class 'twinspan'
as.dendrogram(
  object,
  what = c("quadrat", "species"),
  height = c("level", "chi", "eigen"),
  ...
)
```

Arguments

object	twinspan result object.
what	Return either a "quadrat" or "species" dendrogram.
height	Use either division levels ("level"), total Chi-squares ("chi") or eigenvalues of first axis ("eigen") of division as dendrogram heights.
...	Other parameters to functions.

Details

The dendrogram heights are levels of divisions, total Chi-squares of divisions and groups, or eigenvalues of divisions depending on argument height. Terminal groups have no eigenvalues, because they were not considered for division. For them the method uses arbitrary value that for a group of n units is proportion $(n-1)/n$ of the height of mother division. Chi-squares are evaluated also for terminal groups. There is no guarantee that eigenvalues or Chi-squares decrease in divisions, and there may be reversals where lower levels are higher than their mother groups, and the plotted trees can be messy and unreadable. Chi-squares decrease more monotonically than eigenvalues of first axis.

R has a wealth of functions to handle and display dendrograms. See dendrogram for general description. There is even stronger support in packages (for instance, **dendextend**).

The terminal groups of twinspan trees are not binary, but may have several elements (quadrats, species). In dendrogram plots, it is best to set type="triangle" for nicer looking trees.

Value

A [dendrogram](#) object.

See Also

[as.hclust.twinspan](#), [dendrogram](#).

Examples

```
## Large datasets are difficult to show in dendrograms: take only
## Northen Boreal quadrats (from 1 to 87).

data(ahti)
tw <- twinspan(ahti[1:87,])
den <- as.dendrogram(tw)
str(den, max.level = 4)
plot(den, type = "triangle", nodePar = list(lab.cex=0.6, pch=NA))
den <- as.dendrogram(tw, height="chi")
plot(den, type = "triangle", nodePar = list(lab.cex=0.6, pch=NA))
```

---

as.hclust.twinspan   *Extract twinspan Grouping as Hierarchical Cluster Tree*

Description

Function extracts classification as an [hclust](#) object. The terminal items are the final groups, but quadrats or species are not shown: [hclust](#) cannot handle polytomies that are needed to display group members. Use [as.dendrogram](#) to show the single items. The group ID number and number of items in the terminal group are used as group names and are displayed in plots.

Usage

```
## S3 method for class 'twinspan'
as.hclust(
  x,
  what = c("quadrat", "species"),
  height = c("level", "chi"),
  binname = FALSE,
  ...
)
```

Arguments

x	[twinspan](#) result object.
what	Extract "quadrat" or "species" classification tree.
height	Use either division levels ("level") or total Chi-squares of division ("chi") as heights of internal nodes in the tree.

binname	Use binary labels for classes instead of decimal numbers.
...	Other parameters to the function (ignored).

Details

Function can return either a tree showing the twinspan hierarchy or showing the heterogeneity of each group or division. In the first case, all divisions and groups at a certain level of hierarchy are at the same height, but in the latter the divisions are at the height defined by their heterogeneity. The criterion of heterogeneity is the total chi-square (also known as inertia) of the matrix that twinspan internally uses in that division (see twintotalchi). This tree gives the visual presentation of the modified method of Roleček et al. (2009).

When tree heights are based on heterogeneity, subgroups can be more heterogeneous than their parent group. These appear as reversed branches in the tree. A warning is issued for each such case.

Value

an hclust object amended with labels for internal nodes (nodelabels).

References

Roleček, J, Tichý, L., Zelený, D. & Chytrý, M. (2009). Modified TWINSPAN classification in which the hierarchy respects cluster heterogeneity. *J Veg Sci* 20: 596–602.

See Also

as.dendrogram.twinspan provides an alternative which also shows the sampling units (quadrats or species). The result is based hclust and can be handled with its support functions. plot.twinspan, image.twinspan display the tree. Function cut.twinspan cuts the tree by a level of hierarchy, and cuth by heterogeneity for original sampling units (quadrats, species).

Examples

```
data(ahti)
tw <- twinspan(ahti)
plot(as.hclust(tw, "species"))
cl <- as.hclust(tw)
## plot and 8 groups by hierarchy level
plot(cl)
rect.hclust(cl, 8)
## plot and 8 groups by heterogeneity
cl <- as.hclust(tw, height="chi")
plot(cl)
rect.hclust(cl, 8)
```

---

cut.twinspan   *Return twinspan Classification at Given Level*

Description

Returns a vector of twinspan classes at a given level of hierarchy or classes respecting group heterogeneity for quadrats or species.

Usage

```
## S3 method for class 'twinspan'
cut(x, level, what = c("quadrat", "species"), binname = FALSE, ...)

cuth(x, what = c("quadrat", "species"), ngroups, binname = FALSE)
```

Arguments

x	twinspan result.
level	Level of hierarchy for classification. If missing, the final level used in the object will be returned.
what	Return either a "quadrat" or "species" classification vector.
binname	Use binary label for classes instead of decimal number.
...	Other parameters (ignored).
ngroups	Number of groups.

Details

[twinspan](#) returns only the the classification at the final level, but any upper level classes can be found by integer divisions by 2. [twinspan](#) bases classification principally on splitting polished ordination axis. Sometimes this allocation is in conflict with indicator pseudospecies. This is called 'misclassification' (see [misclassified](#)). Function [predict.twinspan](#) returns the similar classification based on indicator pseudospecies (also for new data), and [misclassified](#) analyses the differences of these classifications.

Function cuth cuts the classification by class heterogeneity instead of level, and can be used to implement the modified method of Roleček et al. (2009). The groups are formed with decreasing heterogeneity but respecting the hierarchy. Total chi-square (also known as inertia) is used as the criterion of heterogeneity. The criterion is calculated with [twintotalchi](#) and the criterion is based on the same data matrix as internally used in [twinspan](#). The function can also be used for species classification, also with the internally used modified species matrix.

Value

A vector of class numbers for the given level of hierarchy using Twinspan identifiers. For identifiers and levels, see [summary.twinspan](#), [as.hclust.twinspan](#).

References

Roleček, J, Tichý, L., Zelený, D. & Chytrý, M. (2009). Modified TWINSPAN classification in which the hierarchy respects cluster heterogeneity. *J Veg Sci* 20: 596–602.

See Also

`predict.twinspan` gives similar classes, but based on indicator pseudospecies. `cutree` provides a similar functionality for `hclust` trees. Function `as.hclust.twinspan` generates corresponding tree presentation, and `plot.twinspan` will print that tree labelling internal nodes (divisions).

Examples

```
data(ahti)
tw <- twinspan(ahti)
cut(tw)
## traditional twinspan classification by level of hierarchy
cut(tw, level=3)
cut(tw, what = "species")
## number of groups as with level=3, but by group heterogeneity
cuth(tw, ngroups = 8)
```

---

eigenvals.twinspan   *Eigenvalues of twinspan Divisions*

Description

Function returns the eigenvalues of `twinspan` divisions.

Usage

```
## S3 method for class 'twinspan'
eigenvals(x, what = c("quadrat", "species"), ...)
```

Arguments

x	`twinspan` result object.
what	Return eigenvalues of `"quadrat"` or `"species"` divisions.
...	Other arguments (ignored).

Details

The eigenvalues are for the first correspondence analysis axis of downweighted pseudospecies data (see `twinsform`). The eigenvalues are not evaluated for final groups which are not divided further, nor if the analysis had terminated in a mother class. This leaves zeros in the vector of eigenvalues.

The eigenvalues are not additive and each is based on slightly different data due to downweighting (see `twinsform`) and all come from the first axis of respective analysis. They may have a relation

to the magnitude of differences in division, but there is no information on the total Chi-square nor on its reduction in division: the eigenvalues only describe the strength of the axis used as a device of division. Neither can eigenvalues for quadrats and species compared: data for species clustering are constructed in a completely different way than data for quadrat classification.

Eigenvalues are in general not decreasing, but they can increase when divisions proceed. Function `as.dendrogram.twinspan` can try to use division eigenvalues as dendrogram heights, but the resulting trees often have inversions and look messy or even unreadable.

Value

Vector of eigenvalues ordered by division number, with zero for divisions that were not evaluated (terminal groups, division terminated in earlier steps).

See Also

`eigenvals` in **vegan**. `summary.twinspan` displays the eigenvalues numerically, and they can be used in `as.dendrogram.twinspan`.

Examples

```
data(ahti)
tw <- twinspan(ahti)
eigenvals(tw)
```

| image.twinspan | *Image of Quadrat and Species Classification and Average Abundances in Cells* |

Description

Function draws an `image` of mean pseudospecies values in cells of final quadrat and species classifications together with corresponding cluster trees.

Usage

```
## S3 method for class 'twinspan'
image(
  x,
  leadingspecies = FALSE,
  reorder = FALSE,
  height = c("level", "chi"),
  ...
)
```

Arguments

x	[twinspan](#) result object.
leadingspecies	Show averages of leading species (most abundant, ties broken by frequency) of each cluster instead of averages of all species of the group.
reorder	Do not use the original twinspan ordering of quadrats and species, but reorder the dendrograms. Setting this TRUE will use the first correspondence analysis axis to reorder the tree which often improves the diagonal structure. With this option, it is also possible to use arguments Colv and Rowv vectors to reorder the dendrogram by their values. The length of such a vector must correspond to the number of row or column classes.
height	Use either division levels ("level") or total Chi-squares of division ("chi") as heights of internal nodes in the trees boarding the image (see [as.hclust.twinspan](#)).
...	Other arguments passed to [tabasco](#) and further to [heatmap](#).

Details

The function is based on **vegan** function [tabasco](#). Mean abundances in cells are shown by colours which can be modified by the user (see [tabasco](#)). The mean abundances are either the average values of all species of the species group (default), or the average of the most abundant species of the species group (tied abundances broken by species frequency) if leadingspecies=TRUE. The species groups are rows and quadrat groups are columns. These are labelled by their decimal group numbers with information of group sizes, or species by their name if only leading species are shown. These numbers and the order of the groups are similar as in hierarchic cluster trees produced by [as.hclust.twinspan](#) and [summary.twinspan](#). Function also adds quadrat classification tree on the top and species classification tree on the left margin (see [as.hclust.twinspan](#)). These trees show either the levels of hierarchy or the heterogeneities of divided groups depending on the argument height.

Value

Function returns invisibly the matrix of mean abundances used to produce the graph.

See Also

[tabasco](#) and [heatmap](#) for basic functionality. The default colour scheme is based on [heat.colors](#), but better shemes can be constructed (**viridis** package provides clear schemes). The dendrograms are based on [as.hclust.twinspan](#). Function [twintable](#) provides an alternative that can list all quadrats and all species in textual format.

Examples

```
data(ahti)
tw <- twinspan(ahti)
image(tw)
im <- image(tw, height = "chi", leading = TRUE)
## image returns invisibly data
head(im)
```

---

misclassified	*Report Misclassified Quadrats*

Description

[twinspan](#) bases its quadrat classification primarily on ordination axis. In some cases this is in conflict with the classification derived from indicator pseudospecies. This function identifies these cases.

Usage

```
misclassified(x, level, verbose = TRUE, binname = FALSE)
```

Arguments

x	[twinspan](#) result object.
level	Only consider misclassification down to this level.
verbose	If FALSE, returns only the index of misclassified quadrats.
binname	Use binary labels instead of decimal numbers for classes and divergent division in verbose object.

Details

The function compares the final [twinspan](#) classification (from [cut.twinspan](#)) and the classification predicted from indicator pseudospecies (from [predict.twinspan](#)). If these two differ, the quadrat is "misclassified". With verbose=FALSE, the function returns a logical vector that is TRUE for misclassified quadrats. In default, it also returns the names, twinspan classes and predicted classes of misclassified quadrats, and the division where the misclassification occurred and classifications diverged. The divisions and their numbers can be seen in [summary.twinspan](#) and [plot.twinspan](#).

Value

If verbose=TRUE, the function returns an object of class "misclassified" with following elements.

index Index of misclassified quadrats.

labels Labels (names) of quadrats.

class Final classification from [twinspan](#).

predicted Final classification from [predict.twinspan](#).

division The division where the misclassification occurred.

With verbose=FALSE, the function returns a logical vector that is TRUE for misclassified quadrats.

See Also

The basic functions are `cut.twinspan` and `predict.twinspan`. You can see the division numbers
with `summary.twinspan` (with indicator pseudospecies) and in `plot.twinspan`.

Examples

```
data(ahti)
tw <- twinspan(ahti)
misclassified(tw)
## see the ID numbers of divisions
plot(tw)
## only look at misclassifications at first two levels
misclassified(tw, level = 2)
```

---

plot.twinspan   *Plot Classification Tree*

Description

Function displays the classification tree.

Usage

```
## S3 method for class 'twinspan'
plot(
  x,
  what = c("quadrat", "species"),
  height = c("level", "chi"),
  main = "Twinspan Dendrogram",
  binname = FALSE,
  ...
)
```

Arguments

x	`twinspan` result object.
what	Plot `"quadrat"` or `"species"` classification tree.
height	Use either division levels (`"level"`) or total Chi-squares of division (`"chi"`) as heights of internal nodes in the tree.
main	Main title of the plot.
binname	Use binary labels for classes and nodes instead of decimal numbers.
...	Other parameters passed to `plot` and `ordilabel`.

Details

The internal nodes are labelled by the numbers of division. These are the same numbers as used in `summary.twinspan` and returned by `cut.twinspan` or `predict.twinspan` for the same classification level. For terminal groups the plot shows the numeric code of the group and the number of items (quadrats or species) in the group. For division number k, its daughter divisions or groups are coded $2k$ and $2k+1$. The tree is similar as a plot of `as.hclust.twinspan`, but adds numbers of internal nodes. The tree can be based either on the levels of hierarchy or on the heterogeneity of division as assessed by chi-square (or inertia) of the division (see `as.hclust.twinspan`).

See Also

`summary.twinspan` for similar textual presentation also showing the items (quadrats, species) in terminal groups. **vegan** function `scores.hclust` can extract the coordinates of internal (or terminal nodes), and `ordilabel` is used add the labels on internal nodes.

Examples

```
data(ahti)
tw <- twinspan(ahti)
plot(tw, "species")
## default plot for quadrats
plot(tw)
## plot by the heterogeneity of divisions
plot(tw, height = "chi")
```

---

predict.twinspan   *Predict Class Membership of Quadrats*

Description

Function predicts the class membership for each quadrat using the reported indicator pseudospecies and limit for the indicator score for the "positive" (right) group.

Usage

```
## S3 method for class 'twinspan'
predict(object, newdata, level, binname = FALSE, ...)
```

Arguments

object	twinspan result object.
newdata	Data used in prediction. The species will be matched by their names, and the pseudospecies are based on the `cutlevels` used in the original `twinspan` model.
level	Level of hierarchy of classification. If missing, the prediction is made to the highest level of classification.
binname	Use binary labels instead of decimal class numbers.
...	Other parameters passed to the function (ignored).

Details

The `twinspan` classification is based on splicing polarized ordination axis, and the reported indicator pseudospecies only indicate the decisions in each division, and do not necessarily give the same classification: The original classification cannot be necessarily found when giving the original data as `newdata`. In the original TWINSPAN this is called misclassification.

See Also

[cut.twinspan](#) gives the original classification, and [misclassified](#) analyses the differences of this and `predict`.

Examples

```
data(ahti)
tw <- twinspan(ahti)
predict(tw)
predict(tw, level=3)
## misclassifications: predict and twinspan differ
sum(predict(tw) != cut(tw))
## build model for 4/5 of data and predict for the removed 1/5
i <- rep(1:5, length = nrow(ahti))
i <- sample(i) # shuffle in random order
tw <- twinspan(ahti[i != 1,]) # remove i==1
predict(tw, newdata = ahti[i==1,])
```

---

summary.twinspan   *Summary of twinspan Classification*

Description

The function gives a compact summary of divisions with indicator species and items in final classification. The output gives the same essential information as the printed output of TWINSPAN batch program, but in more compact form.

Usage

```
## S3 method for class 'twinspan'
summary(object, what = c("quadrat", "species"), binname = FALSE, ...)
```

Arguments

object	[twinspan](#) result object.
what	Summarize either quadrat or species classification.
binname	Use binary labels for divisions instead of decimal numbers.
...	Other arguments (ignored).

Details

For each division, `summary` prints the eigenvalue. For quadrat divisions, it also prints the indicator pseudospecies with their signs, followed by < and the lowest indicator score for the 'positive' (right) group. If the indicator score is below this value, follow the summary to the next item at the lower level, and if the indicator score is at the limit or higher, follow to the second alternative. For division number k, the next items are either $2k$ ('negative' group) or $2k + 1$ ('positive' group). Function `plot.twinspan` displays the division numbers in a classification tree.

For terminal groups, the function gives the size of the group and lists its elements (quadrats or species).

Value

The function returns nothing. It only prints the result object in a human-readable way.

See Also

`plot.twinspan` displays the same structure visually. Function `predict.twinspan` follows the summary strcture to predict the classification with indicator pseudospecies.

Examples

```
data(ahti)
tw <- twinspan(ahti)
summary(tw)
summary(tw, "species")
```

---

totalchi	*Total Scaled Chi-square of All Divisions and Groups*

Description

Total scaled Chi-square is the same as the sum of all eigenvalues in Correspondence Analysis. It is a measure of heterogeneity of a set of data. The function finds this measure for all divisions and terminal groups in `twinspan`

Usage

```
totalchi(x)

twintotalchi(x, what = c("quadrat", "species"))
```

Arguments

x	A matrix of non-negative data for `totalchi` or a `twinspan` result object for `twintotalchi`.
what	Analyse `quadrat` or `species` classification.

Details

Function first reconstructs the data as it was internally used in twinspan using functions twin2stack and twin2specstack. The scaled Chi-square is calculated with support function totalchi that can be called independently for any matrix with non-negative data. The scaling in Chi-square means that data are standardized to unit sum before calculating the actual Chi-square. This is often called the sum of all eigenvalues in Correspondence Analysis, but no eigenvalues are evaluated in these functions, only their potential sum.

See Also

The basic functions are twin2stack and twin2specstack that construct the data matrices.

Examples

```
data(ahti)
tw <- twinspan(ahti)
twintotalchi(tw)
```

---

twin2stack	*Extract Transformed Input Data from twinspan Result*

Description

Functions extract the data twinspan used in its analysis, and allow reproducing the internal ordination and inspecting the twinspan divisions.

Usage

```
twin2stack(x, subset, downweight = FALSE)

twin2mat(x)

twin2specstack(x, subset, downweight = TRUE)
```

Arguments

x	twinspan result object.
subset	Select a subset of quadrats (twin2stack) or species (twin2specstack).
downweight	Downweight infrequent pseudospecies.

Details

Function twin2stack extracts the pseudospecies matrix, where columns are pseudospecies with their cutlevels. This is similar to the file generated with twinsform. The default is to return a binary matrix, where data entries are eiter 0 or 1. Alternatively, it is possible to extract a subset of data with downweighting allowing scrutiny of twinspan divisions. When downweighted data are ordinated with correspondence analysis (such as **vegan** functions cca, decorana with ira=1) the

first eigenvalue will match the eigenvalue in twinspan, and when a division is used as a subset, its eigenvalue will match with twinspan.

Function twin2mat extracts data file with pseudospecies transformation. Columns are original species, and entries are abundances after pseudospecies transformation. This is similar as the output from **vegan** function coverscale with similar cut levels and argument character=FALSE. These data were not analysed in twinspan, but these are the data tabulated with twintable.

Function twin2specstack returns similar data as used in species classification in twinspan. In this matrix, species are rows and columns are "pseudocluster" preferences. The preference of each species in each terminal group and internal division is estimated as proportion of its abundance (in pseudospecies scale, see twin2mat) in the group and all data. If this proportion is 0.8 or higher, species is regarded as present at pseudocluster value 1, if it is 2 or higher at value 2, and if it is 6 or higher at value 3. With downweight=FALSE these data are returned. The columns are named by their division or cluster number followed a, b and c for pseudocluster levels (and including zero columns). In default, the pseudocluster values are still downweighted using species frequencies as weights, and then rows are weighted by species frequencies and columns by their totals extended to the same lowest level of classification, giving two times higher weight to higher "pseudocluster" levels b and c. When ordinated with correspondence analysis (cca, decorana with ira=1) this gives similar eigenvalue for the first axis as in twinspan, and when a division is used as a subset, similar eigenvalue as in that division.

See Also

For original data set instead of twinspan result, functions twinsform and coverscale are analogous to twin2stack and twin2mat.

Examples

```
data(ahti)
dim(ahti)
range(ahti)
tw <- twinspan(ahti)
x <- twin2mat(tw)
dim(x)
range(x)
colnames(x)
x <- twin2stack(tw)
dim(x)
range(x)
colnames(x)
## Inspect division 4
x <- twin2stack(tw, subset = cut(tw, 2) == 4, downweight = TRUE)
## need vegan for correspondence analysis
if (suppressPackageStartupMessages(require("vegan"))) {
cca(x)
}
## species classification
x <- twin2specstack(tw)
if (suppressPackageStartupMessages(require("vegan"))) {
cca(x)
}
```

---

twinid   *Get Twinspan Classs Identiers for Clustering Object*

Description

Twinspan returns the classification topology as a single integer vector. These functions find similar classification identifiers for each sampling unit, or `cut` that vector for a lower number of classes.

Usage

```
twinid(hclust, ...)

## S3 method for class 'twinid'
cut(x, level, binname = FALSE, ...)
```

Arguments

hclust	Cluster Analysis result compatible with [hclust](#).
...	Other parameters to functions (ignored).
x	Vector of classification IDs from `twinind`.
level	Level of hierarchy of classification. If missing, level used in the object will be returned.
binname	Use binary labels instead of decimal class numbers.

Details

Twinspan expresses the topology of cluster tree as an integer. When a cluster z is split into two, its daughters will be $2z$ and $2z + 1$, and its parent cluster is found with integer division $z/2$. The classification vector only stores the topology of the trees, and has no information on heights.

[twinspan](#) will not split small clusters and only proceeds to a defined depth of divisions. In contrast, `twinind` proceeds to each terminal unit (leaf, sampling unit, quadrat) and these will all have unique identifiers. With `cut` function you can restrict the identifiers to certain level of classification similarly as in twinspan (see [cut.twinspan](#)).

Value

Vector of class `"twinid"` giving [twinspan](#) id of each sampling unit.

Warning

If the classification is deep and has many (> 30) levels of hierarchy, the identifiers can exceed the integer maximum in R, and leaves may have non-unique identifiers, and may not recover the correct topoloty. However, they may still be unique beyond this limit, but the user should check this after getting a warning.

Examples

```
data(ahti)
cl <- hclust(dist(ahti, "manhattan"), "average")
(id <- twinid(cl))
cut(id, 6)
table(cut(id, 6))
```

---

twinsform   *Transform Data for Correspondence Analysis like twinspan*

Description

Function transforms data so that Correspondence Analysis gives the same result as in twinspan
divisions.

Usage

```
twinsform(x, cutlevels = c(0, 2, 5, 10, 20), subset, downweight = TRUE)
```

Arguments

x	Input (community) data.
cutlevels	Cut levels used to split quantitative data into binary pseudospecies.
subset	Logical vector or indices that select a subset of quadrats (sampling units).
downweight	Downweight result similarly as in decorana. Downweighting is needed to replicate the process in twinspan, but it can be left out when we only want to have a stacked data set for other uses.

Details

In twinspan, quantitative species data are split into binary (0/1) pseudospecies by cutlevels. All
these pseudospecies are stacked as columns in a new data set. Rare pseudospecies that occur at
lower frequency than 0.2 are downweighted within twinspan. This reduces the weight of rare
species or rare abundance levels in correspondence analysis, but downweighting is optional in this
function.

When the downweighted data are analysed with correspondence analysis (e.g., cca, decorana with
option ira=1), these will give the same first eigenvalue and ordination as in twinspan. When a
subset of a twinspan class is used, correspondence analysis of subdivision of the class can be
obtained.

Value

A stacked matrix of optionally downweighted pseudospecies.

See Also

downweight in **vegan**: this function is often used with Detrended Correspondence Analysis (decorana). However, the implementation is slightly different in TWINSPAN, and weights differ slightly. Function twin2stack extracts similar data from a twinspan result object.

Examples

```
data(ahti)
tahti <- twinsform(ahti)
colnames(tahti)
## needs vegan for correspondence analysis
if (suppressPackageStartupMessages(require("vegan"))) {
decorana(tahti, ira=1)
}
## similar first eigenvalue
eigenvals(twinspan(ahti))
```

---

twinspan   *Two-Way Indicator Species Analysis*

Description

Two-Way Indicator Analysis (TWINSPAN) is a divisive classification method that works by splitting first Correspondence Analysis into two classes, and then recursively working with each split subset. The current function is based on and uses much of the original FORTRAN code of the original TWINSPAN (Hill 1979). twinspan is the main function of this package, but it works silently and prints very little information: you must use separate support functions to extract various aspects of the result.

Usage

```
twinspan(
  x,
  cutlevels = c(0, 2, 5, 10, 20),
  indmax = 7,
  groupmin = 5,
  levmax = 6,
  lind,
  lwgt,
  noind
)
```

Arguments

x   Input data, usually a species community data set where columns give the species and rows the sampling units.

cutlevels	Cut levels used to split quantitative data into binary pseudospecies. Max of 9 cutlevels can be used.
indmax	Maximum number of indicators for division (15 or less).
groupmin	Minimum group size for division (2 or larger).
levmax	Maximum depth of levels of divisions (15 or less).
lind	Weights for levels of pseudospecies. For example indicator potentials c(1, 0, 0,1, 0) signify that pseudospecies at levels 1 and 4 can be used as indicators, but that those at other levels cannot. In the default case, all species are available.
lwgt	Weights for the levels of pseudospecies. For example weights c(1, 2, 2, 2) signify that pseudospecies corresponding to 3 higher cut levels are to be given twice the weight of pseudospecies at the lowest level.
noind	Numbers (indices) of species that you wish to omit from list of potential indicators. Species omitted from this list are used in the calculation, but cannot appear as indicators.

Details

twinspan may not print anything when it runs, but it will return its result that you should save for later use. The functions that reproduce most of the traditional printout are `summary.twinspan` and `twintable`. The summary prints the history of divisions with eigenvalues, signed indicator pseudospecies and the threshold of indicator scores for division, and for terminal groups it prints the group size and group members (quadrats, species). Function `twintable` prints the classified community table. In addition, `plot.twinspan` shows the dendrogram corresponding to the summary with division numbers and sizes and id number of terminal groups. Function `image.twinspan` provides a graphical overview of major structure of classification as a prelude to `twintable`. With function `as.dendrogram.twinspan` it is possible to construct a `dendrogram` of complete classification down to final units (quadrats, species), and `as.hclust.twinspan` constructs an `hclust` tree down to final groups.

The `twinspan` function performs the classic TWINSPAN with fixed levels of hierarchy, but with other functions in this package it is also possible to perform the modified method of Roleček et al. (2009): the topography of the tree is the same, but the division heights are based on the hetergeneity of the group, and the groups are extracted in the order of their heterogeneity. The measure of the heterogeneity is total standardized chi-square (or inertia) of the divided group. This is based on the same matrix as used internally in twinspan code (see support functions `twintotalchi`, `twin2stack` and `twin2specstack`).

The classification at any level of division can be extracted with `cut.twinspan`, and the most heterogenous groups with `cuth`. Function `predict.twinspan` provides a similar classification vector, but based on indicator pseudospecies, and can be used also with new data that was not used in twinspan. These two classifications are often in conflict, and `misclassified` will detect those cases and the divisions where the two classifications diverged. Function `eigenvals` extracts the eigenvalues of divisions, and `twintotalchi` finds the "sum of all eigenvalues" or the standardized chi-square of each division or final group.

Function `twinsform` transforms the data similarly as twinspan and can be used to reproduce the results of any single division. Functions `twin2mat` and `twin2stack` extract the internal data matrices in standard R format from the twinspan result.

Value

Function returns an object of class `"twinspan"`, with following items:

call Function call.

cutlevels Defined cutlevels. These will be used in `predict.twinspan`.

levelmax Maximum level depth of divisions. The divisions will end when this depth is achieved.

nspecies Number of species.

nquadrat Number of quadrats.

idat Pseudospecies data in the internal format used in twinspan. Functions `twin2mat` and `twin2stack` can change this into more usable format.

quadrat Results for quadrats (described below).

species Results for species (described below).

The results of the analysis are stored in items `quadrat` and `species` with similar structure, but `species` has only items `iclass`, `eig`, `labels` and `index` of the following:

iclass ID numbers of final classes at the lowest level of hierarchy. These can be extracted with `cut.twinspan` which also can transform these to any higher level of hierarchy.

eig Eigenvalues of divisions.

labels Name labels of units (species, quadrats).

index An index to order the units in twinspan displays, e.g., in `twintable`.

indicators A matrix of dimensions maximum number of indicators × maximum number of divisions giving the signed indices of indicator species for the division. These are shown in labels in `summary.twinspan` and used by `predict.twinspan`.

positivelimit Lowest value of indicator score for positive group in a division.

indlabels Labels of pseudospecies.

pseudo2species Index from pseudospecies to the corresponding species.

Method

TWINSPAN is very complicated and has several obscure details, and it will not be explained in details in this manual, but you should consult the source code or literature sources. Hill (1979) is the most authoritative source, but may be difficult to find. Kent & Coker (1991) do a great job in explaining the method, including many obscure details.

A strong simplification (but often sufficient to understand the basic principles) is that TWINSPAN is a divisive clustering based on splitting first correspondence analysis axis, and applying the same method recursively for resulting classes. The same method is used first for quadrats and then for species. In addition, it finds the species abundance levels (called 'pseudospecies') that best indicate these divisions facilitating ecologist's understanding of classes. Species classification is performed so that it best corresponds to the previous quadrat classification also with species composition.

The following details are more technical. The analysis starts with splitting species abundance data into discrete abundance levels called pseudospecies. With these the function constructs a stacked binary matrix with values of 0 and 1, where value 1 means that species occurs at given threshold (called 'cut level') in a quadrat. Then the pseudospecies (abundance levels) that occur in fewer

than 1/5 of quadrats are downweighted so that their presences (values 1) are reduced linearly towards minimimum value of 0.01 according to their frequencies. This will reduce their impact in correspondence analysis which is regarded as being sensitive to rare species. Each cut level of the species is downweighted independently. The first axis of correspondence analysis is found for the downweighted data. This initial step can be reproduced with the help of functions `twinsform` or `twin2stack`. However, the division is not based on this step only. Next the method finds the best indicator pseudospecies for the division. Further, it polarizes the ordination by using indicator scores for all species to find the final classes for quadrats. It does not mechanically just split the axis in the middle, but it finds the cutpoint so that indicator scores from the indicator pseudospecies and final split are as concordant as possible. Then the analysis is repeated for both resulting groups, including downweighting within the subset of quadrats.

After quadrat classification, TWINSPAN constructs species data which are completely different from the data used in quadrat classification. Species values depend on their ability to discriminate quadrat classes at any level of classification. Then species are classified in the same way and with the same code as the quadrats. In this way species classification is concordant with quadrat classification, and good indicators of quadrat classes are grouped together. The species classification can be reproduced with function `twin2specstack` which also provides a more detailed description of the data structure used at this stage.

Function `twinspan` performs only the classical TWINSPAN, but with support functions the modified method of Roleček et al. (2009) can be performed (see `cuth`, `as.hclust.twinspan`).

References

Hill, M.O. (1979). *TWINSPAN - a FORTRAN program for arranging multivariate data in an ordered two-way table by classification of individuals and attributes.* Cornell Univ., Dept of Ecology and Systematics.

Kent, M. & Coker, P. (1992) *Vegetation description and analysis: A practical approach.* John Wiley & Sons.

Roleček, J, Tichý, L., Zelený, D. & Chytrý, M. (2009). Modified TWINSPAN classification in which the hierarchy respects cluster heterogeneity. *J Veg Sci* 20: 596–602.

Examples

```
data(ahti)
## default cut levels
(tw <- twinspan(ahti))
## visual look at the divisions and group numbers
plot(tw)
## Braun-Blanquet scale
(twb <- twinspan(ahti, cutlevels = c(0, 0.1, 1, 5, 25, 50, 75)))
plot(twb)
## compare confusion
table(cut(tw, level=3), cut(twb, level=3))
## modified method of Roleček et al. (2009)
plot(twb, height="chi", main = "Rolecek tree")
## compare against the default by hierarchy levels
table(cuth(twb, ngroups=8), cut(twb, level=3))
```

---

twintable	*Community Table Ordered by Twinspan Classification*

Description

Prints a community table of pseudospecies ordered by [twinspan](#) classification.

Usage

```
twintable(object, maxspp, goodspecies, subset)
```

Arguments

object	[twinspan](#) result object.
maxspp	Maximum number of most abundant species displayed. The abundance is estimated with pseudospecies cut levels. The default is to show all species.
goodspecies	Select "good species" for tabulation. These are either species that were used as indicator pseudospecies (`"indicator"`), or most abundant species in each final species group breaking ties with frequency (`"leading"`), or `"both"` (default). The abundance is estimated after pseudospecies transformation for all quadrats and cannot be used together with maxspp.
subset	Select a subset of quadrats.

Details

Function prints a compact community table of pseudospecies values. The table is ordered by clustering both species and quadrats similarly as in [summary.twinspan](#) or in plot of [as.dendrogram.twinspan](#). The classification of each quadrat and species is shown by a sequence of `0` and `1` indicating division of each level. This string is binary presentation of the decimal class number without the leading `1`.

Only one character is used for each abundance, and the table is very compact. However, large tables can be divided over several pages or screen windows. The width of the displayed table is controlled by R option width (see [options](#)). It is possible to select only a subset of the quadrats for tabulation giving narrower tables. The number of species can be reduced by setting the maximum number of most abundant species, or alternatively, by restricting tabulation only to "good species" which are the most abundant species of each species group (ties broken by species frequency), or species used as indicators, or both.

Value

Function returns invisibly the data that [vegemite](#) used. This data can also be exported or used as any other data set.

Examples

```
data(ahti)
tw <- twinspan(ahti)
## complete table would be large, but we subset
twintable(tw, subset = cut(tw, 2) == 4, goodspecies = "both")
```

Index