Package 'MultiTraits'

October 1, 2025

Type Package

Title Analyzing and Visualizing Multidimensional Plant Traits

Version 0.6.0

Date 2025-10-1

Maintainer Yan He <heyan@njfu.edu.cn>

Description Implements analytical methods for multidimensional plant traits, including Competitors-Stress tolerators-Ruderals strategy analysis using leaf traits, Leaf-Height-Seed strategy analysis, Niche Periodicity Table analysis, and Trait Network analysis. Provides functions for data analysis, visualization, and network metrics calculation. Methods are based on Grime (1974) <doi:10.1038/250026a0>, Pierce et al. (2017) <doi:10.1111/1365-2435.12882>, Westoby (1998) <doi:10.1023/A:1004327224729>, Winemiller et al. (2015) <doi:10.1111/ele.12462>, He et al. (2020) <doi:10.1016/j.tree.2020.06.003>.

Imports
igraph,Hmisc,corrplot,vegan,ggrepel,ape,dplyr,ggraph,ggsci,rpart,magrittr,rlang,ggplot2,scatterplot3d

Depends R (>= 4.0.0)

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

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Author Yan He [aut, cre],
Zhaogang Liu [aut],
Jiangshan Lai [aut],
Lingfeng Mao [aut]

Repository CRAN

Date/Publication 2025-10-01 16:00:03 UTC

Contents

CSR	*Classify Plant Strategies using Pierce et al. (2017) CSR Method*

Description

This function calculates CSR (Competitor-Stress tolerator-Ruderal) ecological strategies for plant species based on three key functional traits: Leaf Area (LA), Leaf Dry Matter Content (LDMC), and Specific Leaf Area (SLA). The CSR classification system was originally developed by Grime (1974) and this implementation follows the global method by Pierce et al. (2017).

Usage

```
CSR(data)
```

Arguments

data   A data frame containing at least three numeric columns: LA, LDMC, and SLA.

- **LA**: Leaf area in mm^2.
- **LDMC**: Leaf dry matter content (%).
- **SLA**: Specific leaf area in mm^2/mg.

Details

The function implements the global CSR classification method which:

1. Transforms the three input traits using species-specific equations
2. Calculates derived traits including leaf dry weight, fresh weight, and succulence index
3. Adjusts LDMC for succulent species (>5 g dm-2)
4. Projects traits onto principal component axes from a global calibration dataset
5. Applies outlier corrections to keep values within calibrated ranges
6. Converts to proportional CSR values that sum to 100%
7. Assigns the closest matching tertiary CSR strategy type

The three strategies represent different ecological approaches:

- **C (Competitor)**: Species adapted to productive, undisturbed environments
- **S (Stress-tolerator)**: Species adapted to resource-limited environments
- **R (Ruderal)**: Species adapted to frequently disturbed environments

Value

A data frame containing the original input data plus four additional columns:

- C: Competitor strategy proportion (0-100%)
- S: Stress-tolerator strategy proportion (0-100%)
- R: Ruderal strategy proportion (0-100%)
- type: CSR strategy type classification (character) - one of 19 possible categories: "C", "C/CR", "C/CS", "CR", "C/CSR", "CS", "CR/CSR", "CS/CSR", "R/CR", "CSR", "S/CS", "R/CSR", "S/CSR", "R", "SR/CSR", "S", "R/SR", "S/SR", "SR"

Note

- All input values must be positive numbers
- NA values are not permitted and will cause the function to stop with an error

References

1. Grime, J.P. (1974). Vegetation classification by reference to strategies. Nature, 250, 26–31.
2. Pierce, S., Negreiros, D., Cerabolini, B.E.L., Kattge, J., Díaz, S., et al. (2017). A global method for calculating plant CSR ecological strategies applied across biomes world-wide. Functional Ecology, 31: 444-457.

Examples

```
LA <- c(369615.7, 11.8, 55.7, 36061.2, 22391.8, 30068.1, 31059.5, 29895.1)
LDMC <- c(25.2, 39.7, 13.3, 35.5, 33.2, 36.1, 35.2, 34.9)
SLA <- c(17.4, 6.6, 34.1, 14.5, 8.1, 12.1, 9.4, 10.9)
traits <- data.frame(LA, LDMC, SLA)
CSR(data = traits)
```

CSR_hodgson   *Classify Plant Strategies using Hodgson et al. (1999) CSR Method*

Description

This function calculates C, S, and R scores as percentages based on input plant trait data, following the approach of Hodgson et al. (1999) and its application in Caccianiga et al. (2006). Input is a dataframe with specific trait columns, and the output is a new dataframe containing calculated CSR coordinates, percentages, and assigned CSR type.

Usage

```
CSR_hodgson(data)
```

Arguments

data   A data.frame containing the following columns:

        growth_form Character vector: plant growth form, "g" for graminoid, "n" for non-graminoid.

        CH Numeric: Canopy height (mm).

        LDMC Numeric: Leaf dry matter content (percent).

        FP Numeric: Flowering period (# of months).

        LS Numeric: Lateral spread (six-point classification).

        LDW Numeric: Leaf dry weight (mg).

        SLA Numeric: Specific leaf area (mm2/mg).

        FS Numeric: Flowering start (month).

Details

Implements the Hodgson et al. (1999) method for allocating plant species into the CSR (Competitor–Stress-tolerator–Ruderal) triangle based on plant functional traits. Also assigns each species to the nearest CSR type.

This implementation:

- Uses different equations for graminoids and non-graminoids to compute raw CSR dimensions.
- Scales results to coordinate space $[-2.5, 2.5]$, then shifts to positive and converts to percentages.
- Assigns the nearest CSR type based on standard reference CSR percentages from Hodgson's scheme.

Value

A data.frame with the following columns:

- growth_form, CH, LDMC, FP, LS, LDW, SLA, FS — copied from input;
- C, S, R — calculated CSR percentages;
- type — assigned CSR type label (e.g., "C", "CSR", "S/CSR").

Note

Input data must not contain NA values in required columns. If such values are present, the function will stop with an error.

References

1. Hodgson, J.G., Wilson, P.J., Hunt, R., Grime, J.P. & Thompson, K. (1999). Allocating CSR plant functional types: a soft approach to a hard problem. Oikos, 85, 282–294.

2. Caccianiga, M., Luzzaro, A., Pierce, S., Ceriani, R.M. & Cerabolini, B. (2006). The functional basis of a primary succession resolved by CSR classification. Oikos, 112, 10–20.

Examples

```
# Example trait dataset
traits <- data.frame(
  growth_form = c("g", "g", "n", "g", "n"),
  CH = c(45.3, 169.7, 13.7, 132.7, 76.0),
  LDMC = c(33.0, 37.9, 25.9, 28.0, 15.7),
  FP = c(2, 2, 2, 1, 2),
  LS = c(3, 5, 4, 2, 5),
  LDW = c(1.9, 9.9, 2.3, 7.5, 40.2),
  SLA = c(19.0, 20.4, 15.2, 22.6, 21.8),
  FS = c(5, 5, 4, 5, 5)
)

# Run CSR classification
result <- CSR_hodgson(traits)
print(result)

# Plot CSR positions
CSR_plot(data = result)
```

CSR_plot   *Create a Ternary Plot for CSR Plant Ecological Strategies*

Description

This function creates a ternary plot to visualize plant ecological strategies based on the CSR (Competitor-Stress tolerator-Ruderal) framework developed by Grime (1974). The plot is built using ggplot2 and displays the relative proportions of C, S, and R strategies for each species or sample.

Usage

```
CSR_plot(
  data,
  point_size = 3,
  point_shape = 21,
```

```
    custom_colors = c(C = "#E60D0D", `C/CR` = "#BA0D3B", `C/CS` = "#BA3B0D", CR =
     "#7A0D7A", `C/CSR` = "#8A3B3B", CS = "#7A7A0D", `CR/CSR` = "#6B2B6B", `CS/CSR` =
      "#6B6B2B", `R/CR` = "#3B0DBA", CSR = "#545454", `S/CS` = "#3BBA0D", `R/CSR` =
     "#3B3B8A", `S/CSR` = "#3B8A3B", R = "#0D0DE6", `SR/CSR` = "#2B6B6B", S = "#0DE60D",
       `R/SR` = "#0D3BBA", `S/SR` = "#0DBA3B", SR = "#0D7A7A")
)
```

Arguments

data	A data frame containing CSR strategy data. Must include columns:

        C Numeric vector of Competitor strategy values (0-100)

        S Numeric vector of Stress-tolerator strategy values (0-100)

        R Numeric vector of Ruderal strategy values (0-100)

        type Character vector indicating the CSR strategy type/classification

point_size	Numeric value specifying the size of points in the plot. Default is 3.
point_shape	Numeric value specifying the shape of points in the plot. Default is 21 (filled circle with border).
custom_colors	Named character vector specifying custom colors for each CSR strategy type. Default includes 19 predefined colors for all possible CSR combinations.

Details

The CSR strategy framework classifies plants into three primary functional types based on their ecological strategies:

C (Competitors) Species adapted to productive, low-stress environments

S (Stress-tolerators) Species adapted to unproductive, high-stress environments

R (Ruderals) Species adapted to productive, high-disturbance environments

The ternary plot allows visualization of the relative contribution of each strategy, where each point represents a species positioned according to its C, S, and R values (which sum to 100%).

Value

A ggplot object representing a ternary plot with:

- Points colored by CSR strategy type
- Ternary coordinate system with C, S, R axes
- Legend showing strategy types and their colors
- Grid lines and arrows for better visualization

References

1. Grime, J.P. (1974). Vegetation classification by reference to strategies. Nature, 250, 26–31.
2. Hodgson, J.G., Wilson, P.J., Hunt, R., Grime, J.P. & Thompson, K. (1999). Allocating CSR plant functional types: a soft approach to a hard problem. Oikos, 85, 282–294.

3. Caccianiga, M., Luzzaro, A., Pierce, S., Ceriani, R.M. & Cerabolini, B. (2006). The functional basis of a primary succession resolved by CSR classification. Oikos, 112, 10–20.

4. Pierce, S., Negreiros, D., Cerabolini, B.E.L., Kattge, J., Díaz, S., et al. (2017). A global method for calculating plant CSR ecological strategies applied across biomes world-wide. Functional Ecology, 31: 444-457.

See Also

CSR or CSR_hodgson for calculating CSR strategies from plant functional traits

Examples

```
data(PFF)
head(PFF)
traits <- data.frame(LA=PFF$Leaf_area, LDMC=PFF$LDMC, SLA=PFF$SLA)
head(traits)
result <- CSR(data = traits)
head(result)
CSR_plot(data=result)
```

LHS	*Calculate LHS (Leaf-Height-Seed) Plant Ecological Strategy Classification*

Description

The LHS scheme uses three fundamental plant traits that reflect important trade-offs controlling plant strategies:

- **Specific Leaf Area (SLA)**: Light-capturing area deployed per unit dry mass, reflecting the trade-off between rapid resource acquisition and leaf longevity

- **Height**: Canopy height at maturity, expressing the amount of growth attempted between disturbances and competitive ability for light

- **Seed Mass**: Reflecting the trade-off between seed number and individual seed provisioning, affecting dispersal capacity and seedling survival

All three axes are log-scaled as they are approximately lognormally distributed between species. Species are classified into eight strategy types based on whether their log-transformed trait values are above (L = Large) or below (S = Small) the median values.

Usage

```
LHS(data)
```

Arguments

data   A data frame containing plant trait data with the following required columns:

> **SLA** Specific leaf area (area per unit dry mass, typically mm2/mg or m2/kg)
> **Height** Canopy height at maturity (typically in metres)
> **SeedMass** Seed mass (typically in mg or g)

> Row names should represent species names or identifiers.

Details

This function implements the LHS plant ecology strategy scheme proposed by Westoby (1998), which classifies plant species based on three key functional traits: specific leaf area (SLA), canopy height at maturity, and seed mass. The LHS scheme provides a quantitative framework for comparing plant ecological strategies worldwide.

The function performs the following operations:

1. Validates input data for required columns and checks for missing, zero, or negative values
2. Log-transforms all three traits
3. Calculates median values for each log-transformed trait
4. Classifies each species based on whether traits are above (L) or below (S) medians
5. Returns the original data with added log-transformed columns and strategy classification

Value

A data frame with the original columns plus:

log_SLA Natural logarithm of SLA

log_Height Natural logarithm of Height

log_SeedMass Natural logarithm of SeedMass

LHS_strategy Character string indicating the LHS strategy type (e.g., "S-L-S")

References

1. Westoby, M. (1998). A leaf-height-seed (LHS) plant ecology strategy scheme. Plant and Soil, 199, 213–227.
2. Yang, J., Wang, Z., Zheng, Y., & Pan, Y. (2022). Shifts in plant ecological strategies in remnant forest patches along urbanization gradients. Forest Ecology and Management, 524, 120540.

Examples

```
data(PFF)
pff <- PFF[, c("SLA", "Height", "SeedMass")]
rownames(pff) <- PFF$species
head(pff)
result <- LHS(pff)
head(result)
```

LHS_plot	*Create 3D Scatter Plot for LHS Plant Ecological Strategy Data*

Description

The LHS scheme uses three fundamental plant traits that reflect important trade-offs controlling plant ecological strategies :

- **Specific Leaf Area (SLA)**: Light-capturing area deployed per unit dry mass, reflecting the trade-off between rapid resource acquisition and leaf longevity
- **Height**: Canopy height at maturity, expressing the amount of growth attempted between disturbances and competitive ability for light
- **Seed Mass**: Reflecting the trade-off between seed number and individual seed provisioning, affecting dispersal capacity and seedling survival

All three axes are log-scaled as they are approximately lognormally distributed between species. The 3D visualisation allows researchers to explore species clustering and relationships within the LHS strategy space, facilitating the identification of functional groups and ecological patterns.

Usage

```
LHS_plot(
  data,
  group = NULL,
  show_cube = TRUE,
  colors = c("#30123BFF", "#4777EFFF", "#1BD0D5FF", "#62FC6BFF", "#D2E935FF",
    "#FE9B2DFF", "#DB3A07FF", "#7A0403FF"),
  cube_angle = 60
)
```

Arguments

data	A data frame containing LHS analysis results with the following required columns:
	log_SLA Natural logarithm of specific leaf area
	log_Height Natural logarithm of canopy height at maturity
	log_SeedMass Natural logarithm of seed mass
	Typically this would be the output from the LHS() function. Row names should represent species names or identifiers.
group	Character string specifying the column name to use for grouping points by colour. If NULL (default), all points will be plotted in the same colour. Common choices include ″LHS_strategy″ to colour by the eight LHS strategy types (e.g., "S-S-S", "L-L-L") or any other categorical variable in the dataset.
show_cube	Logical indicating whether to display the 3D cube frame around the plot. Default is TRUE, which helps with spatial orientation and depth perception.

colors	Character vector of colours to use for different groups. Should contain at least as many colours as there are levels in the grouping variable. Default provides a viridis-inspired colour palette with 8 colours suitable for the 8 LHS strategy types. If only one group is plotted, only the first colour will be used.
cube_angle	Numeric value specifying the viewing angle for the 3D plot in degrees. Default is 60. Values between 40-80 typically provide good visualisation perspectives.

Details

This function creates a three-dimensional scatter plot to visualise the LHS (Leaf-Height-Seed) plant ecological strategy scheme proposed by Westoby (1998). The plot displays species positions in the three-dimensional LHS space defined by log-transformed specific leaf area (SLA), canopy height at maturity, and seed mass.

The function creates a 3D scatter plot using the scatterplot3d package, with log-transformed trait values on each axis. When grouping is specified, the plot includes an automated legend positioned to the right of the main plot area.

The three axes represent the core dimensions of the LHS ecological strategy space:

- X-axis: log(SLA) - reflects the leaf economics spectrum from resource-conservative to resource-acquisitive strategies
- Y-axis: log(Height) - represents the plant size spectrum and competitive ability for light capture
- Z-axis: log(Seed Mass) - indicates the seed size spectrum affecting dispersal and establishment success

The function automatically handles layout management when legends are displayed, ensuring optimal use of plotting space.

Value

Invisibly returns the scatterplot3d object, which can be used for adding additional graphical elements to the plot if needed.

References

1. Westoby, M. (1998). A leaf-height-seed (LHS) plant ecology strategy scheme. Plant and Soil, 199, 213–227.
2. Yang, J., Wang, Z., Zheng, Y., & Pan, Y. (2022). Shifts in plant ecological strategies in remnant forest patches along urbanization gradients. Forest Ecology and Management, 524, 120540.

Examples

```
data(PFF)
pff <- PFF[, c("SLA", "Height", "SeedMass")]
rownames(pff) <- PFF$species
head(pff)
result <- LHS(pff)
head(result)
```

```
LHS_plot(result)
LHS_plot(result, group = "LHS_strategy", show_cube = TRUE)
```

LHS_strategy_scheme   *Create a table of Leaf-Height-Seed (LHS) strategy types*

Description

This function generates a data frame containing different plant growth strategies based on the Leaf-Height-Seed (LHS) scheme. Each strategy is described by a combination of traits and their corresponding ecological interpretation.

Usage

```
LHS_strategy_scheme()
```

Value

A data frame with two columns:

type Character vector of LHS strategy combinations (e.g., "L-L-L", "L-L-S", etc.)

strategy Character vector describing the ecological strategy for each type

References

1. Westoby, M. (1998). A leaf-height-seed (LHS) plant ecology strategy scheme. Plant and Soil, 199, 213–227.

2. Yang, J., Wang, Z., Zheng, Y., & Pan, Y. (2022). Shifts in plant ecological strategies in remnant forest patches along urbanization gradients. Forest Ecology and Management, 524, 120540.

Examples

```
LHS_strategy_scheme()
```

NPT_continuous   *Continuous Niche Classification Based on Periodic Table of Niches*

Description

This function implements a continuous niche classification scheme based on the periodic table of niches concept. It performs a hierarchical Principal Component Analysis (PCA) approach where separate PCAs are conducted on different niche dimensions, followed by a second-level PCA to integrate results across dimensions.

Usage

```
NPT_continuous(data, dimension)
```

Arguments

data
: A data frame containing species and their functional trait measurements. Each row represents a species and columns contain trait values. The data should include all traits specified in the dimension parameter.

dimension
: A named list where each element represents a niche dimension (e.g., "grow", "survive", "reproductive") and contains a character vector of column names corresponding to traits associated with that dimension. Each dimension should contain multiple functionally related traits.

Details

The function implements a two-stage hierarchical PCA approach based on the methodology described in Winemiller et al. (2015) for creating continuous niche classification schemes. This approach addresses the challenge that analysis of data sets containing many functionally unrelated measures may fail to detect patterns of covariation that determine species' ecological responses to and effects on their environments.

Stage 1: Dimensional PCA Analysis

Separate Principal Component Analysis is performed on trait data for each niche dimension using the internal `pca_first` function. This dimensional approach ensures that all niche dimensions have an equal opportunity to influence the composite niche scheme and species ordinations. For each dimension, the function:

- Performs PCA using `vegan::rda()`
- Calculates variance explained by the first two principal components
- Identifies traits with highest absolute loadings on PC1 and PC2
- Extracts species scores on both principal components

Stage 2: Integration PCA

The species scores from the first two principal components of each dimensional PCA are combined into a new data matrix (with columns named as "pc1.dimension" and "pc2.dimension"). A second PCA is then performed on this matrix to create a two-dimensional continuum integrating patterns

(strategies) within each of the niche dimensions. This creates a continuous ordination of species within niche space that can be used for comparative ecological analyses.

Methodological Advantages

This hierarchical PCA method prevents domination by any single type of trait or dimension. The approach allows all niche dimensions to have equal influence on the composite niche scheme, with gradients dominated by those dimensional components having greatest influence on community structure patterns.

Value

A list containing three elements:

PCA_first	A data frame summarizing the first-level PCA results for each dimension, including variance explained by PC1 and PC2 (as percentages), and the traits with highest absolute loadings on each principal component axis
PCA_second	A matrix containing species scores from the second-level PCA that integrates all niche dimensions into a unified ordination space
result	The complete second-level PCA result object from vegan::rda() containing detailed ordination results for further analysis

Note

Important Considerations:

- Missing values (NA) are automatically removed using na.omit()
- A message displays the number of rows removed due to NA values
- Each niche dimension should contain multiple functionally related traits
- Users should ensure traits are appropriately scaled/transformed before analysis
- The function uses scaling = FALSE in PCA, assuming pre-standardized data
- Column names in the data must exactly match trait names in dimensions list

References

1. Winemiller, K. O., Fitzgerald, D. B., Bower, L. M., & Pianka, E. R. (2015). Functional traits, convergent evolution, and periodic tables of niches. Ecology letters, 18(8), 737-751.

2. Yu, R., Huang, J., Xu, Y., Ding, Y., & Zang, R. (2020). Plant functional niches in forests across four climatic zones: Exploring the periodic table of niches based on plant functional traits. Frontiers in Plant Science, 11, 841.

Examples

```
data(PFF)
PFF[,4:21] <- log(PFF[,4:21])
traits_dimension <- list(
  grow = c("SLA","SRL","Leaf_Nmass","Root_Nmass"),
  survive = c("Height","Leaf_CN","Root_CN"),
  reproductive = c("SeedMass","FltDate","FltDur")
)
```

```
result <- NPT_continuous(data = PFF, dimension = traits_dimension)
result
```

NPT_continuous_plot   *Plot Continuous Niche Classification Results*

Description

This function creates a biplot visualization of the continuous niche classification results from the hierarchical Principal Component Analysis. It displays species ordination in niche space with optional grouping and shows the contribution of different niche dimensions as arrows.

Usage

```
NPT_continuous_plot(pca_obj, group = NULL, default_fill = "#1373D3")
```

Arguments

pca_obj   The PCA result object from `NPT_continuous()$result`, which should be a vegan rda object containing the second-level PCA results that integrate all niche dimensions into a unified ordination space.

group   Optional vector specifying group membership for each species/sample. If provided, points will be colored by group. If NULL (default), all points will have the same color. Length should match the number of rows in the original data.

default_fill   Character string specifying the default fill color when no grouping is applied. Default is "#1373D3" (blue).

Details

The function creates a standard PCA biplot where:

- Points represent species positioned in the integrated niche space
- Red arrows represent the niche dimensions (from first-level PCAs) and their relative contribution to the ordination axes
- Arrow length indicates the strength of correlation with the ordination axes
- Arrow direction shows the gradient direction in niche space
- Dashed reference lines at x=0 and y=0 help interpret the ordination

The plot helps interpret:

- Species clustering patterns in niche space
- Which niche dimensions drive the main gradients
- Relationships between different niche dimensions
- Group differences in niche occupation (when groups are specified)

Value

A ggplot2 object containing the biplot visualization with:

- Points representing species/samples in the ordination space

- Arrows showing the direction and magnitude of niche dimension contributions

- Labels for niche dimensions (arrows)

- Variance explained by PC1 and PC2 in axis labels

- Optional color coding by groups if provided

Note

- The function requires the result object from `NPT_continuous()`

- A larger plotting window is recommended for better visualization

- Arrow labels show dimension names (e.g., "pc1.grow", "pc2.survive")

- The function uses `max.overlaps = Inf` to show all labels

- Group colors are automatically assigned if groups are provided

References

1. Winemiller, K. O., Fitzgerald, D. B., Bower, L. M., & Pianka, E. R. (2015). Functional traits, convergent evolution, and periodic tables of niches. Ecology letters, 18(8), 737-751.

2. Yu, R., Huang, J., Xu, Y., Ding, Y., & Zang, R. (2020). Plant functional niches in forests across four climatic zones: Exploring the periodic table of niches based on plant functional traits. Frontiers in Plant Science, 11, 841.

Examples

```
data(PFF)
PFF[,4:21] <- log(PFF[,4:21])
PFF <- na.omit(PFF)
traits_dimension <- list(
  grow = c("SLA","SRL","Leaf_Nmass","Root_Nmass"),
  survive = c("Height","Leaf_CN","Root_CN"),
  reproductive = c("SeedMass","FltDate","FltDur")
)
npt_result <- NPT_continuous(data = PFF, dimension = traits_dimension)
NPT_continuous_plot(npt_result$result)
NPT_continuous_plot(npt_result$result, PFF$family)
```

NPT_discrete   *Discrete Niche Classification Based on Periodic Table of Niches*

Description

This function implements a discrete niche classification scheme based on the periodic table of niches concept. It performs Principal Component Analysis (PCA) on functional traits grouped by niche dimensions, followed by clustering to create hierarchical niche classifications.

Usage

```
NPT_discrete(data, dimension, clustering_method = "CART", k_max = 6)
```

Arguments

data	A data frame containing species and their functional trait measurements. Each row represents a species and columns contain trait values.
dimension	A named list where each element represents a niche dimension (e.g., "Growth", "Survival", "Reproduction") and contains a character vector of column names corresponding to traits associated with that dimension.
clustering_method	
	Character string specifying the clustering method to use. Options are:

- "CART" - Classification and Regression Trees (default)
- "kmeans" - K-means clustering with automatic k selection

k_max	Integer specifying the maximum number of clusters allowed for k-means clustering. Default is 6. This parameter is only used when clustering_method = "kmeans". The optimal k value will be selected using the elbow method, constrained by this maximum value.

Details

The function implements the methodology described in Winemiller et al. (2015) for creating discrete niche classification schemes. The approach follows three main steps:

Step 1: PCA Analysis by Dimension

Separate Principal Component Analysis is performed on trait data for each niche dimension. This dimensional approach prevents functionally unrelated traits from masking important ecological patterns. The first two principal components are retained for each dimension.

Step 2: Clustering Methods

Two clustering approaches are available:

- **CART**: Uses the Euclidean distance from the origin in PCA space as the response variable and original trait values as predictors to create regression trees. The tree is pruned using cross-validation.

- **k-means**: Performs clustering on the two-dimensional PCA space with automatic optimal k selection using the elbow method. The optimal k is constrained by the k_max parameter (default maximum = 6).

Step 3: Hierarchical Niche Classification

The function combines clustering results from all niche dimensions to create a comprehensive niche classification scheme. Each species receives a niche code representing its cluster membership across all dimensions (e.g., "1,2,1" for cluster 1 in dimension 1, cluster 2 in dimension 2, and cluster 1 in dimension 3).

The function also calculates niche occupancy statistics, comparing the number of realized niches to the total number of potential niche combinations.

Value

A list containing two elements:

niche_classification

A data frame with species names, cluster assignments for each dimension, and comprehensive niche codes

summary   A summary data frame showing unique niche codes, the number of species in each niche, and lists of species names

Note

- Missing values should be handled prior to using this function.

- The function prints diagnostic information during execution, including variance explained by PCs and clustering results.

- For k-means clustering, setting a lower k_max value will force simpler clustering solutions, while higher values allow for more complex niche subdivisions.

- **Randomness warning:** The kmeans method involves random initialization of cluster centers, so results may vary between runs. For reproducibility, set a random seed using set.seed() before running this function. CART results may also differ slightly if predictor splitting involves tie-breaking.

References

1. Winemiller, K. O., Fitzgerald, D. B., Bower, L. M., & Pianka, E. R. (2015). Functional traits, convergent evolution, and periodic tables of niches. Ecology letters, 18(8), 737-751.

2. Pianka, E. R., Vitt, L. J., Pelegrin, N., Fitzgerald, D. B., & Winemiller, K. O. (2017). Toward a periodic table of niches, or exploring the lizard niche hypervolume. The American Naturalist, 190(5), 601-616.

Examples

```
## Not run:
# Load and prepare data
data(PFF)
rownames(PFF) <- PFF$species
```

```
PFF_traits <- PFF[, c("SLA", "SRL", "Leaf_Nmass", "Root_Nmass","Height",
                      "Leaf_CN", "Root_CN","SeedMass", "FltDate", "FltDur")]
# Perform log transformation of data and remove missing values
PFF_traits <- log(na.omit(PFF_traits))
head(PFF_traits)
# Define trait dimensions
dimension <- list(Grow = c("SLA", "SRL", "Leaf_Nmass", "Root_Nmass"),
                  Survive = c("Height", "Leaf_CN", "Root_CN"),
                  Reproductive = c("SeedMass", "FltDate", "FltDur"))

set.seed(123)
discrete_result <- NPT_discrete(data = PFF_traits, dimension = dimension)
head(discrete_result$niche_classification)

## End(Not run)
```

NPT_discrete_plot   *Visualize Discrete Niche Classification Tree*

Description

This function creates a hierarchical tree visualization of the discrete niche classification results generated by `NPT_discrete`. The visualization displays the niche classification structure as a directed tree graph, where each level represents a niche dimension and nodes represent different clusters within each dimension.

Usage

```
NPT_discrete_plot(data, point_size = 2.5)
```

Arguments

data   A data frame containing niche classification results, typically the `niche_classification` output from `NPT_discrete`. Must contain:

- `species` - Character column with species names
- `niche_code` - Character column with comma-separated niche codes (e.g., "1,2,1")
- Additional columns representing cluster assignments for each dimension (used as layer names in the visualization)

point_size   Numeric value specifying the size of nodes in the tree plot. Default is 2.5.

Details

The function constructs a hierarchical tree visualization where:

- **Root node**: Represents the starting point of the classification
- **Level 1 nodes**: Show clusters in the first niche dimension

- **Level 2 nodes**: Show clusters in the second niche dimension, connected to their parent clusters from level 1

- **Subsequent levels**: Continue the hierarchical pattern for additional dimensions

Visualization Features:

- Different colors for each niche dimension level using NPG color palette

- Directed edges showing the hierarchical relationships

- Node labels indicating cluster numbers

- Legend showing niche levels

- Automatic detection of the number of dimensions from niche codes

Tree Construction Algorithm:

The function automatically determines the number of niche dimensions by parsing the first niche code, then recursively builds the tree structure:

1. Parse niche codes to extract individual dimension values

2. Create nodes for each unique cluster combination

3. Establish parent-child relationships between levels

4. Generate the final graph structure for visualization

Value

A `ggplot` object representing the niche classification tree. The plot can be further customized using standard `ggplot2` functions.

Note

- The function assumes that niche codes are comma-separated strings

- Layer names are automatically extracted from column names (excluding "species" and "niche_code")

- If the number of layer names doesn't match dimensions, generic names (Layer1, Layer2, etc.) are used

References

1. Winemiller, K. O., Fitzgerald, D. B., Bower, L. M., & Pianka, E. R. (2015). Functional traits, convergent evolution, and periodic tables of niches. Ecology letters, 18(8), 737-751.

2. Pianka, E. R., Vitt, L. J., Pelegrin, N., Fitzgerald, D. B., & Winemiller, K. O. (2017). Toward a periodic table of niches, or exploring the lizard niche hypervolume. The American Naturalist, 190(5), 601-616.

Examples

```
## Not run:
# Load and prepare data
data(PFF)
rownames(PFF) <- PFF$species
PFF_traits <- PFF[, c("SLA", "SRL", "Leaf_Nmass", "Root_Nmass","Height",
                      "Leaf_CN", "Root_CN","SeedMass", "FltDate", "FltDur")]
# Perform log transformation of data and remove missing values
PFF_traits <- log(na.omit(PFF_traits))
head(PFF_traits)
# Define trait dimensions
dimension <- list(Grow = c("SLA", "SRL", "Leaf_Nmass", "Root_Nmass"),
                  Survive = c("Height", "Leaf_CN", "Root_CN"),
                  Reproductive = c("SeedMass", "FltDate", "FltDur"))

set.seed(123)
discrete_result <- NPT_discrete(data = PFF_traits, dimension = dimension)
NPT_discrete_plot(discrete_result$niche_classification)


## End(Not run)
```

PFF	*Plant Functional Traits Dataset from Ponderosa Pine Forests Flora (PFF)*

Description

A dataset containing functional traits for 133 plant species commonly found in southwestern USA Pinus ponderosa var. scopulorum P. & C. Lawson (ponderosa pine) forests.

Usage

```
PFF
```

Format

A data frame with 137 rows and 21 variables:

species Plant species name

genus Plant genus name

family Plant family name

Height Canopy height (cm)

Leaf_area Leaf area (mm^2)

LDMC Leaf dry matter content (%)

SLA Specific leaf area (mm^2/mg)

SRL Specific root length (m/g)

SeedMass Seed mass (mg)

FltDate Mean flowering date (Julian day)

FltDur Mean flowering duration (days)

k_value Decomposition decay constant, where proportion of original mass remaining = exp(- k-value*0.926)

Leaf_Cmass Leaf carbon content (% dry mass)

Leaf_Nmass Leaf nitrogen content (% dry mass)

Leaf_CN Leaf carbon to nitrogen ratio

Leaf_Pmass Leaf phosphorus content (% dry mass)

Leaf_NP Leaf nitrogen to phosphorus ratio

Leaf_CP Leaf carbon to phosphorus ratio

Root_Cmass Root carbon content (% dry mass)

Root_Nmass Root nitrogen content (% dry mass)

Root_CN Root carbon to nitrogen ratio

Details

This dataset contains measurements of a core set of functional traits that reflect aspects of each species' ability to disperse, establish, acquire water and nutrients, and photosynthesize. Traits include specific leaf area (SLA), height, seed mass, specific root length (SRL), leaf and fine root nitrogen concentration, leaf phosphorus concentration, and leaf dry matter content (LDMC). Julian flowering date and flowering duration were also obtained for each species. Leaf litter decomposition rates were measured on 103 species.

Source

Laughlin, D. C., Leppert, J. J., Moore, M. M., & Sieg, C. H. (2010). A multi-trait test of the leaf-height-seed plant strategy scheme with 133 species from a pine forest flora. Functional Ecology, 24(3), 485-700.

Examples

```
data(PFF)
head(PFF)
```

PFF_tree   *Phylogenetic Tree of 133 Plant Species from Pine Forest Flora*

Description

A phylogenetic tree containing 133 plant species from a pine forest flora study. The tree includes both tip and node labels, is rooted, and contains branch lengths.

Usage

```
PFF_tree
```

Format

A phylo object (phylogenetic tree) with the following structure:

tips 133 terminal nodes representing plant species

nodes 121 internal nodes with taxonomic labels

tip.label Species names including Hymenopappus mexicanus, Heliomeris multiflora, Bahia dissecta, etc.

node.label Taxonomic group names including Spermatophyta, eudicotyledons, Pentapetalae, etc.

edge.length Branch lengths indicating evolutionary distances

root Tree is rooted

Details

This phylogenetic tree represents the evolutionary relationships among 133 plant species found in pine forest ecosystems. The tree structure includes:

- 133 terminal taxa (tip labels with species names)
- 121 internal nodes with higher-level taxonomic classifications
- Branch lengths representing evolutionary time or genetic distance
- Root indicating the common ancestor of all included species

The tree was constructed as part of a comprehensive study examining plant functional strategies in pine forest communities, specifically testing the leaf-height-seed (LHS) plant strategy scheme proposed in functional ecology.

Source

Laughlin, D. C., Leppert, J. J., Moore, M. M., & Sieg, C. H. (2010). A multi-trait test of the leaf-height-seed plant strategy scheme with 133 species from a pine forest flora. Functional Ecology, 24(3), 485-700.

Examples

```
data(PFF_tree)
print(PFF_tree)
```

---

phylo_correlation   *Phylogenetically Corrected Correlation Analysis*

Description

This function calculates phylogenetically independent correlations between traits using phylogenetic independent contrasts (PICs). It accounts for phylogenetic relationships when computing correlations, which is important when analyzing trait data from related species.

Usage

```
phylo_correlation(traits_matrix, phylo_tree, method = "pearson")
```

Arguments

traits_matrix	A numeric matrix or data frame where rows represent species and columns represent traits. Row names should contain species names that match the tip labels in the phylogenetic tree.
phylo_tree	A phylogenetic tree object of class "phylo" (from the ape package). The tree should contain the same species as in the traits matrix.
method	Character string specifying the correlation method to use. Options are "pearson" (default) or "spearman".

Details

The function performs the following steps:

1. Checks if the phylogenetic tree is binary and converts it if necessary

2. Matches species between the trait matrix and phylogenetic tree

3. For each pair of traits, calculates phylogenetic independent contrasts (PICs)

4. Computes correlations between PICs instead of raw trait values

5. Handles missing data by using only complete cases for each trait pair

The phylogenetic independent contrasts method removes the effects of phylogenetic relatedness, allowing for proper statistical inference about trait correlations. This is crucial when analyzing data from related species, as standard correlation methods may be biased due to phylogenetic non-independence.

Value

A list containing two matrices:

r	A symmetric correlation matrix with phylogenetically corrected correlation coefficients
P	A symmetric matrix of p-values corresponding to the correlation tests

Note

- The function requires at least 3 species with complete data for each trait pair
- Non-binary trees are automatically converted to binary using `multi2di()`
- Species present in the tree but missing from the trait matrix will generate warnings
- The function handles missing values by performing pairwise complete case analysis

References

Felsenstein, J. (1985). Phylogenies and the comparative method. The American Naturalist, 125(1), 1-15.

Harvey, P. H., & Pagel, M. D. (1991). The comparative method in evolutionary biology. Oxford University Press.

Examples

```
data(PFF)
data(PFF_tree)
rownames(PFF) <- PFF$species
traits <- PFF[,4:21]
head(traits)
phylo_correlation(traits, PFF_tree, method = "pearson")
```

PTN   *Generate Plant Trait Network*

Description

This function creates a network graph from a plant trait correlation matrix, applying thresholds for correlation strength and significance. It supports both standard correlations and phylogenetic independent contrasts.

Usage

```
PTN(
  traits_matrix,
  rThres = 0.2,
  pThres = 0.05,
  method = "pearson",
  phylo_correction = FALSE,
  phylo_tree = NULL
)
```

Arguments

`traits_matrix`	A numeric matrix or data frame where each column represents a plant trait and each row represents a sample/species. Row names should contain species names when using phylogenetic correction.
`rThres`	Numeric, threshold for correlation coefficient, default is 0.2. Correlations with absolute values below this threshold are set to zero.
`pThres`	Numeric, threshold for p-value, default is 0.05. Only correlations with p-values below this threshold are included in the network.
`method`	Character, specifies the correlation method to use: "pearson" (default) or "spearman".
`phylo_correction`	
	Logical, whether to apply phylogenetic correction using phylogenetic independent contrasts, default is FALSE.
`phylo_tree`	A phylo object from the ape package containing the phylogenetic tree. Required when phylo_correction = TRUE. Species names in the tree must match row names in traits_matrix.

Details

The function performs the following steps:

1. Validates input parameters and phylogenetic tree compatibility (if applicable).

2. Calculates correlation coefficients and p-values using either standard correlation or phylogenetic independent contrasts.

3. Applies correlation coefficient and p-value thresholds to filter relationships.

4. Adjusts p-values using False Discovery Rate (FDR) correction.

5. Constructs an unweighted undirected graph from the filtered correlation matrix.

6. Removes self-loops and isolated nodes from the graph.

7. Adds correlation coefficients as edge attributes.

When phylo_correction = TRUE, the function:

- Matches species names between traits_matrix and phylo_tree
- Calculates phylogenetic independent contrasts for each trait
- Computes correlations between contrasts to control for phylogenetic relatedness

Value

Returns an igraph object representing the trait network with the following attributes:

- Edge attribute 'correlation': original correlation values (positive or negative)
- Edge attribute 'weight': absolute correlation values used for network analysis
- Vertices represent traits that pass the correlation and significance thresholds

References

1. He, N., Li, Y., Liu, C., et al. (2020). Plant trait networks: improved resolution of the dimen-sionality of adaptation. Trends in Ecology & Evolution, 35(10), 908-918.

2. Li, Y., Liu, C., Sack, L., Xu, L., Li, M., Zhang, J., & He, N. (2022). Leaf trait network architecture shifts with species-richness and climate across forests at continental scale. Ecology Letters, 25(6), 1442-1457.

Examples

```
# Example 1: Standard trait network analysis
data(PFF)
rownames(PFF) <- PFF$species
PFF_traits <- PFF[, c("Height", "Leaf_area","LDMC","SLA","SRL","SeedMass","FltDate",
                      "FltDur","Leaf_Cmass","Leaf_Nmass","Leaf_CN","Leaf_Pmass",
                      "Leaf_NP","Leaf_CP","Root_Cmass","Root_Nmass","Root_CN")]
PFF_traits <- na.omit(PFF_traits)
head(PFF_traits)

ptn_result <- PTN(traits_matrix = PFF_traits, rThres = 0.2, pThres = 0.05, method = "pearson")
ptn_result

# Example 2: Phylogenetically corrected trait network analysis
data(PFF_tree)

# Trait network with phylogenetic correction
ptn_phylo_result <- PTN(traits_matrix = PFF_traits,
                        rThres = 0.2,
                        pThres = 0.05,
                        method = "pearson",
                        phylo_correction = TRUE,
                        phylo_tree = PFF_tree)
ptn_phylo_result
```

PTN_corr   *Calculate and Visualize Plant Trait Correlation Network*

Description

This function calculates correlation coefficients for given plant traits and generates a correlation network plot. It supports both standard correlation analysis and phylogenetically corrected correlation analysis using phylogenetic independent contrasts.

Usage

```
PTN_corr(
  traits_matrix,
```

```
    rThres = 0.2,
    pThres = 0.05,
    method = "pearson",
    phylo_correction = FALSE,
    phylo_tree = NULL
)
```

Arguments

`traits_matrix`	A numeric matrix or data frame where each column represents a plant trait and each row represents a sample (species). Row names should contain species names when using phylogenetic correction.
`rThres`	Numeric, threshold for correlation coefficient, default is 0.2. Only correlations with absolute values above this threshold will be displayed in the plot. Must be between 0 and 1.
`pThres`	Numeric, threshold for p-value, default is 0.05. Only correlations with p-values below this threshold will be displayed in the plot. Must be between 0 and 1.
`method`	Character, specifies the correlation method to use: "pearson" (default) or "spearman".
`phylo_correction`	
	Logical, whether to apply phylogenetic correction using phylogenetic independent contrasts. Default is FALSE.
`phylo_tree`	A phylo object (from the ape package) containing the phylogenetic tree. Required when phylo_correction = TRUE. Species names in the tree must match row names in traits_matrix.

Details

The function performs the following steps:

1. Validates input parameters and data structure
2. If phylogenetic correction is requested, matches species between traits_matrix and phylo_tree
3. Calculates correlation coefficients using either standard correlation or phylogenetic independent contrasts
4. Adjusts p-values using the False Discovery Rate (FDR) method
5. Creates a correlation network plot using hierarchical clustering for trait ordering
6. Marks non-significant correlations with red crosses based on both correlation and p-value thresholds

When phylo_correction = TRUE, the function uses phylogenetic independent contrasts to account for phylogenetic relationships among species, which helps control for the non-independence of species data due to shared evolutionary history.

Value

Returns a correlation network plot object created by corrplot. The plot displays correlations as circles, with positive correlations in blue and negative correlations in red. Non-significant correlations (based on thresholds) are marked with red crosses.

Note

- Missing values (NA) should be handled before using this function
- When using phylogenetic correction, ensure species names are consistent between traits_matrix row names and phylo_tree tip labels
- The function requires the Hmisc, corrplot, and (optionally) ape packages

References

1. Felsenstein, J. (1985). Phylogenies and the comparative method. The American Naturalist, 125(1), 1-15.

2. He, N., Li, Y., Liu, C., et al. (2020). Plant trait networks: improved resolution of the dimensionality of adaptation. Trends in Ecology & Evolution, 35(10), 908-918.

3. Li, Y., Liu, C., Sack, L., Xu, L., Li, M., Zhang, J., & He, N. (2022). Leaf trait network architecture shifts with species-richness and climate across forests at continental scale. Ecology Letters, 25(6), 1442-1457.

Examples

```
data(PFF)
rownames(PFF) <- PFF$species
PFF_traits <- PFF[, c("Height", "Leaf_area","LDMC","SLA","SRL","SeedMass","FltDate",
                      "FltDur","Leaf_Cmass","Leaf_Nmass","Leaf_CN","Leaf_Pmass",
                      "Leaf_NP","Leaf_CP","Root_Cmass","Root_Nmass","Root_CN")]
PFF_traits <- na.omit(PFF_traits)
head(PFF_traits)
PTN_corr(traits_matrix = PFF_traits, rThres = 0.2, pThres = 0.05, method = "pearson")

data(PFF_tree)
PTN_corr(traits_matrix = PFF_traits, rThres = 0.2, pThres = 0.05, method = "pearson",
        phylo_correction = TRUE, phylo_tree = PFF_tree)
```

PTN_metrics   *Calculate Node and Global Metrics for Trait Networks*

Description

This function computes various node and global metrics for a trait network graph.

Usage

```
PTN_metrics(graph)
```

Arguments

graph   An igraph object representing the trait network, typically generated by the PTN function.

Value

A list containing two data frames:

node   A data frame with node-level metrics including degree, closeness, betweenness, and local clustering coefficient.

global   A data frame with global metrics including edge density, diameter, average path length, average clustering coefficient, and modularity.

References

1. He, N., Li, Y., Liu, C., et al. (2020). Plant trait networks: improved resolution of the dimensionality of adaptation. Trends in Ecology & Evolution, 35(10), 908-918.

2. Li, Y., Liu, C., Sack, L., Xu, L., Li, M., Zhang, J., & He, N. (2022). Leaf trait network architecture shifts with species-richness and climate across forests at continental scale. Ecology Letters, 25(6), 1442-1457.

Examples

```
data(PFF)
rownames(PFF) <- PFF$species
PFF_traits <- PFF[, c("Height", "Leaf_area","LDMC","SLA","SRL","SeedMass","FltDate",
                      "FltDur","Leaf_Cmass","Leaf_Nmass","Leaf_CN","Leaf_Pmass",
                      "Leaf_NP","Leaf_CP","Root_Cmass","Root_Nmass","Root_CN")]
PFF_traits <- na.omit(PFF_traits)
head(PFF_traits)
ptn_result <- PTN(traits_matrix = PFF_traits, rThres = 0.2, pThres = 0.05)
PTN_metrics(ptn_result)

data(PFF_tree)
ptn_phylo_result <- PTN(traits_matrix = PFF_traits,
                        rThres = 0.2,
                        pThres = 0.05,
                        method = "pearson",
                        phylo_correction = TRUE,
                        phylo_tree = PFF_tree)
PTN_metrics(ptn_phylo_result)
```

PTN_plot   *Plot Trait Network Graph*

Description

This function visualizes the trait network graph generated by the PTN function.

Usage

```
PTN_plot(graph, style = 1, vertex.size = 20, vertex.label.cex = 0.6)
```

Arguments

graph   An igraph object representing the trait network.

style   A numeric value that determines the plotting style (default is 1).

vertex.size   Numeric value for the size of vertices in the plot (default is 20).

vertex.label.cex
                            Numeric value for the scaling factor of vertex labels (default is 0.6).

Details

The function uses the `cluster_edge_betweenness` algorithm to identify communities in the graph
and assigns community membership to vertices. It offers two plotting styles:

- Style 1: Plots the community structure.
- Style 2: Plots the graph in a circular layout with vertex colors representing communities. The
  vertex size and label size can be customized using vertex.size and vertex.label.cex parameters
  respectively.

Value

An object of class `igraph`. This function generates a visualization of the trait network graph. When
style = 1, it displays a community structure plot. When style = 2, it displays a circular layout
plot where vertex colors represent community membership, edge thickness represents correlation
strength, and edge color represents the sign of the correlation (black for positive, red for negative).

References

1. He, N., Li, Y., Liu, C., et al. (2020). Plant trait networks: improved resolution of the dimen-
   sionality of adaptation. Trends in Ecology & Evolution, 35(10), 908-918.

2. Li, Y., Liu, C., Sack, L., Xu, L., Li, M., Zhang, J., & He, N. (2022). Leaf trait network ar-
   chitecture shifts with species-richness and climate across forests at continental scale. Ecology
   Letters, 25(6), 1442-1457.

Examples

```
data(PFF)
PFF_traits <- PFF[, c("Height", "Leaf_area","LDMC","SLA","SRL","SeedMass","FltDate",
                      "FltDur","Leaf_Cmass","Leaf_Nmass","Leaf_CN","Leaf_Pmass",
                      "Leaf_NP","Leaf_CP","Root_Cmass","Root_Nmass","Root_CN")]
PFF_traits <- na.omit(PFF_traits)
head(PFF_traits)
ptn_result <- PTN(traits_matrix = PFF_traits, rThres = 0.2, pThres = 0.05)
PTN_plot(ptn_result, style = 1, vertex.size = 20, vertex.label.cex = 0.6)
PTN_plot(ptn_result, style = 2, vertex.size = 20, vertex.label.cex = 0.6)
```

Index