

# Package ‘MultiTraits’

May 7, 2026

**Type** Package

**Title** Analyzing and Visualizing Multidimensional Plant Traits

**Version** 1.0.0

**Date** 2026-3-22

**Maintainer** Yan He <hey@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 He et al. (2026) <[doi:10.1002/ecog.08026](https://doi.org/10.1002/ecog.08026)>.

## Imports

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

**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** 2026-03-22 06:20:08 UTC

## Contents

average_interlayer_clustering_coefficient . . . . .	2
average_interlayer_path_length . . . . .	4
crosslayer_module_assoc . . . . .	6
cross_layer_groups . . . . .	7
CSR . . . . .	9
CSR_hodgson . . . . .	11
CSR_plot . . . . .	12
forest_invader_traits . . . . .	14
forest_invader_tree . . . . .	17
interlayer_closeness . . . . .	18
interlayer_clustering_coefficient . . . . .	19
interlayer_degree . . . . .	21
interlayer_edge_density . . . . .	23
LHS . . . . .	24
LHS_plot . . . . .	26
LHS_strategy_scheme . . . . .	28
NPT_continuous . . . . .	29
NPT_continuous_plot . . . . .	31
NPT_discrete . . . . .	33
NPT_discrete_plot . . . . .	35
PFF . . . . .	37
PFF_tree . . . . .	39
phylo_correlation . . . . .	40
PTMN . . . . .	41
PTMN_metrics . . . . .	43
PTMN_plot . . . . .	45
PTN . . . . .	48
PTN_corr . . . . .	50
PTN_metrics . . . . .	52
PTN_plot . . . . .	53
<b>Index</b>	<b>55</b>

---

average\_interlayer\_clustering\_coefficient

*Average Interlayer Clustering Coefficient for Plant Trait Multilayer Networks*

---

### Description

Computes the Average Interlayer Clustering Coefficient (AICC) for a plant trait multilayer network (PTMN). The AICC is the mean interlayer clustering coefficient across all nodes in the network, capturing the overall prevalence of tightly knit cross-layer modules. High AICC values indicate that integrated, modular cross-layer structures are prevalent, reflecting strong phenotypic integration and modularity, which can allow flexible responses to multidimensional environmental pressures.

**Usage**

```
average_interlayer_clustering_coefficient(data)
```

**Arguments**

**data** A PTMN object created by the PTMN() function, containing the multilayer network structure with nodes representing plant traits and edges representing trait relationships within and across functional layers.

**Details**

Calculate Average Interlayer Clustering Coefficient

The Average Interlayer Clustering Coefficient is calculated as:

$$AICC = \frac{1}{n} \sum_{i=1}^n ICC_i$$

where  $n$  is the total number of nodes and  $ICC_i$  denotes the interlayer clustering coefficient of node  $v_i$ . This parameter is particularly useful for understanding multidimensional plant adaptations and coordinated responses across different plant organs and functional systems.

In the context of plant ecology, higher AICC values suggest that plants have evolved integrated trait networks that facilitate coordinated responses to environmental challenges. For example, non-native woody species have been shown to exhibit higher AICC values compared to native species, potentially contributing to their invasion success.

**Value**

A numeric value representing the average interlayer clustering coefficient. Values range from 0 to 1, where higher values indicate stronger cross-layer integration and more tightly connected functional modules.

**See Also**

[PTMN](#) for constructing plant trait multilayer networks

**Examples**

```
## Not run:
data(forest_invader_tree)
data(forest_invader_traits)
traits <- forest_invader_traits[, 6:73]
layers <- list(
  shoot_dynamics = c("LeafDuration", "LeafFall150", "LeafRate_max",
                    "Chl_shade50", "LAgain", "FallDuration",
                    "LeafOut", "Chl_sun50", "EmergenceDuration",
                    "LeafTurnover"),
  leaf_structure = c("PA_leaf", "Mass_leaf", "Lifespan_leaf",
                    "Thick_leaf", "SLA", "Lobe", "LDMC",
                    "Stomate_size", "Stomate_index"),
```

```

leaf_metabolism = c("J_max", "Vc_max", "Asat_area", "CC_mass",
                    "LSP", "AQY", "CC_area", "Rd_area",
                    "Asat_mass", "WUE", "Rd_mass", "PNUE"),
leaf_chemistry = c("N_area", "Chl_area", "DNA", "Phenolics",
                  "Cellulose", "N_mass", "N_litter", "Chl_ab",
                  "Chl_mass", "N_res", "C_litter", "C_area",
                  "C_mass", "Ash", "Lignin", "Solubles",
                  "Decomp_leaf", "Hemi"),
root = c("NPP_root", "SS_root", "SRL", "RTD", "RDMC",
         "NSC_root", "Decomp_root", "Starch_root",
         "C_root", "N_root", "Lignin_root"),
stem = c("Latewood_diam", "Metaxylem_diam", "Earlywood_diam",
         "NSC_stem", "Vessel_freq", "SS_stem", "Cond_stem",
         "Starch_stem")
)
graph <- PTMN(traits, layers_list = layers, method = "pearson")
average_interlayer_clustering_coefficient(graph)

## End(Not run)

```

---

average\_interlayer\_path\_length

*Calculate Average Interlayer Path Length (AIPL) for Plant Trait Multilayer Networks*

---

## Description

This function calculates the Average Interlayer Path Length (AIPL) for a Plant Trait Multilayer Network (PTMN). AIPL describes the average path length between pairs of nodes from different network layers and reflects how efficiently information or resources can be transmitted across the network. Lower AIPL values indicate tighter, more efficient cross-layer connections, supporting the rapid transmission of environmental cues and resources among network layers and thereby enhancing the overall adaptability of the system.

## Usage

```
average_interlayer_path_length(data)
```

## Arguments

data	A data frame containing the PTMN edge list with required columns: <code>node.from</code> , <code>layer.from</code> , <code>node.to</code> , and <code>layer.to</code> . This data frame represents the network structure with both intralayer and interlayer connections.
------	---

## Details

The AIPL is calculated using the formula from multilayer network theory:

$$AIPL = \frac{1}{\sum_{\alpha=1}^{M-1} \sum_{\beta=\alpha+1}^M \sum_{i \in \alpha} \sum_{j \in \beta} 1_{d_{ij} < \infty}} \sum_{\alpha=1}^{M-1} \sum_{\beta=\alpha+1}^M \sum_{i \in \alpha} \sum_{j \in \beta} d_{ij}$$

where  $d_{ij}$  is the shortest path length from node  $i$  (in layer  $\alpha$ ) to node  $j$  (in layer  $\beta$ ) and  $1_{d_{ij} < \infty}$  is an indicator function of node reachability.

The function:

1. Creates an undirected graph from the edge list
2. Calculates shortest path distances between all node pairs
3. Considers only node pairs from different layers (interlayer connections)
4. Includes only finite path lengths (reachable pairs)
5. Returns the mean of all valid interlayer path lengths

## Value

A numeric value representing the average interlayer path length across all reachable node pairs between different layers. Returns NaN if no interlayer paths exist or all interlayer node pairs are unreachable.

## See Also

[PTMN](#) for constructing plant trait multilayer networks

## Examples

```
## Not run:
data(forest_invader_tree)
data(forest_invader_traits)
traits <- forest_invader_traits[, 6:73]
layers <- list(
  shoot_dynamics = c("LeafDuration", "LeafFall50", "LeafRate_max",
                    "Chl_shade50", "LAgain", "FallDuration",
                    "LeafOut", "Chl_sun50", "EmergeDuration",
                    "LeafTurnover"),
  leaf_structure = c("PA_leaf", "Mass_leaf", "Lifespan_leaf",
                    "Thick_leaf", "SLA", "Lobe", "LDMC",
                    "Stomate_size", "Stomate_index"),
  leaf_metabolism = c("J_max", "Vc_max", "Asat_area", "CC_mass",
                     "LSP", "AQY", "CC_area", "Rd_area",
                     "Asat_mass", "WUE", "Rd_mass", "PNUE"),
  leaf_chemistry = c("N_area", "Chl_area", "DNA", "Phenolics",
                    "Cellulose", "N_mass", "N_litter", "Chl_ab",
                    "Chl_mass", "N_res", "C_litter", "C_area",
                    "C_mass", "Ash", "Lignin", "Solubles",
                    "Decomp_leaf", "Hemi"),
```

```

root = c("NPP_root", "SS_root", "SRL", "RTD", "RDMC",
         "NSC_root", "Decomp_root", "Starch_root",
         "C_root", "N_root", "Lignin_root"),
stem = c("Latewood_diam", "Metaxylem_diam", "Earlywood_diam",
         "NSC_stem", "Vessel_freq", "SS_stem", "Cond_stem",
         "Starch_stem")
)
graph <- PTMN(traits, layers_list = layers, method = "pearson")
average_interlayer_path_length(graph)

## End(Not run)

```

---

crosslayer\_module\_assoc

*Calculate Module Interlayer Association (MIA) for Plant Trait Multi-layer Networks*

---

## Description

This function computes the Module Interlayer Association (MIA), a key topological parameter for Plant Trait Multilayer Networks (PTMNs). MIA quantifies the proportion of interlayer edges within each network module, reflecting the degree of cross-layer integration within modules. High MIA values suggest the formation of tightly connected cross-layer modules that facilitate coordinated adaptation across different plant organs and functional systems.

## Usage

```
crosslayer_module_assoc(data)
```

## Arguments

data	A data frame containing the PTMN edge information with columns: <code>node.from</code> , <code>node.to</code> , <code>layer.from</code> , and <code>layer.to</code> . This is typically the output from the PTMN function.
------	--

## Details

The function uses the Girvan-Newman algorithm to detect network modules, then calculates the proportion of interlayer edges within each module that spans multiple layers. The MIA is computed as the average of interlayer association values across all cross-layer modules. This metric is particularly useful for understanding how plant traits coordinate across different organs (roots, stems, leaves) and functional systems in response to environmental pressures.

In the context of plant invasion ecology, higher MIA values have been associated with greater invasion success, as they may indicate more efficient coordination among functional systems.

**Value**

A numeric value representing the average Module Interlayer Association (MIA). Returns NA if no interlayer modules are found. Values range from 0 to 1, where higher values indicate stronger cross-layer integration within modules.

**See Also**

[PTMN](#) for constructing plant trait multilayer networks

**Examples**

```
## Not run:
data(forest_invader_tree)
data(forest_invader_traits)
traits <- forest_invader_traits[, 6:73]
layers <- list(
  shoot_dynamics = c("LeafDuration", "LeafFall50", "LeafRate_max",
    "Chl_shade50", "LAgain", "FallDuration",
    "LeafOut", "Chl_sun50", "EmergeDuration",
    "LeafTurnover"),
  leaf_structure = c("PA_leaf", "Mass_leaf", "Lifespan_leaf",
    "Thick_leaf", "SLA", "Lobe", "LDMC",
    "Stomate_size", "Stomate_index"),
  leaf_metabolism = c("J_max", "Vc_max", "Asat_area", "CC_mass",
    "LSP", "AQY", "CC_area", "Rd_area",
    "Asat_mass", "WUE", "Rd_mass", "PNUE"),
  leaf_chemistry = c("N_area", "Chl_area", "DNA", "Phenolics",
    "Cellulose", "N_mass", "N_litter", "Chl_ab",
    "Chl_mass", "N_res", "C_litter", "C_area",
    "C_mass", "Ash", "Lignin", "Solubles",
    "Decomp_leaf", "Hemi"),
  root = c("NPP_root", "SS_root", "SRL", "RTD", "RDMC",
    "NSC_root", "Decomp_root", "Starch_root",
    "C_root", "N_root", "Lignin_root"),
  stem = c("Latewood_diam", "Metaxylem_diam", "Earlywood_diam",
    "NSC_stem", "Vessel_freq", "SS_stem", "Cond_stem",
    "Starch_stem")
)
graph <- PTMN(traits, layers_list = layers, method = "pearson")
crosslayer_module_assoc(graph)

## End(Not run)
```

## Description

This function identifies network modules that span multiple functional layers in Plant Trait Multilayer Networks (PTMNs). Cross-layer modules represent coordinated trait groups that integrate multiple plant organs or functional systems, which are crucial for understanding multidimensional plant adaptations and phenotypic integration.

## Usage

```
cross_layer_groups(data)
```

## Arguments

data	A data frame containing the PTMN edge information with columns: <code>node.from</code> , <code>node.to</code> , <code>layer.from</code> , and <code>layer.to</code> . This is typically the output from the PTMN function.
------	--

## Details

The function uses the Girvan-Newman community detection algorithm to identify network modules, then filters to retain only those modules that contain nodes from multiple functional layers. Cross-layer modules are particularly important in plant ecology as they represent coordinated trait groups that facilitate integrated responses across different organs and functional systems.

In the context of plant invasion ecology, species with more cross-layer modules may exhibit enhanced phenotypic integration and coordination, potentially contributing to invasion success.

## Value

A named list where each element represents a cross-layer module. Each list element contains the node names (trait names) that belong to that cross-layer module. Only modules spanning multiple layers are returned. Returns an empty list if no cross-layer modules are detected.

## See Also

[PTMN](#) for constructing plant trait multilayer networks

## Examples

```
## Not run:
data(forest_invader_tree)
data(forest_invader_traits)
traits <- forest_invader_traits[, 6:73]
layers <- list(
  shoot_dynamics = c("LeafDuration", "LeafFall150", "LeafRate_max",
                    "Chl_shade50", "LAgain", "FallDuration",
                    "LeafOut", "Chl_sun50", "EmergeDuration",
                    "LeafTurnover"),
  leaf_structure = c("PA_leaf", "Mass_leaf", "Lifespan_leaf",
                    "Thick_leaf", "SLA", "Lobe", "LDMC",
                    "Stomate_size", "Stomate_index"),
  leaf_metabolism = c("J_max", "Vc_max", "Asat_area", "CC_mass",
```

```

      "LSP", "AQY", "CC_area", "Rd_area",
      "Asat_mass", "WUE", "Rd_mass", "PNUE"),
leaf_chemistry = c("N_area", "Chl_area", "DNA", "Phenolics",
                  "Cellulose", "N_mass", "N_litter", "Chl_ab",
                  "Chl_mass", "N_res", "C_litter", "C_area",
                  "C_mass", "Ash", "Lignin", "Solubles",
                  "Decomp_leaf", "Hemi"),
root = c("NPP_root", "SS_root", "SRL", "RTD", "RDMC",
        "NSC_root", "Decomp_root", "Starch_root",
        "C_root", "N_root", "Lignin_root"),
stem = c("Latewood_diam", "Metaxylem_diam", "Earlywood_diam",
        "NSC_stem", "Vessel_freq", "SS_stem", "Cond_stem",
        "Starch_stem")
)
graph <- PTMN(traits, layers_list = layers, method = "pearson")
cross_layer_groups(graph)

## End(Not run)

```

---

 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<sup>2</sup>.
- **LDMC**: Leaf dry matter content (%).
- **SLA**: Specific leaf area in mm<sup>2</sup>/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<sup>-2</sup>)

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	<p>A data.frame containing the following columns:</p> <p><b>growth_form</b> Character vector: plant growth form, "g" for graminoid, "n" for non-graminoid.</p> <p><b>CH</b> Numeric: Canopy height (mm).</p> <p><b>LDMC</b> Numeric: Leaf dry matter content (percent).</p> <p><b>FP</b> Numeric: Flowering period (# of months).</p> <p><b>LS</b> Numeric: Lateral spread (six-point classification).</p> <p><b>LDW</b> Numeric: Leaf dry weight (mg).</p> <p><b>SLA</b> Numeric: Specific leaf area (mm<sup>2</sup>/mg).</p> <p><b>FS</b> Numeric: Flowering start (month).</p>
------	---

### 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: <b>C</b> Numeric vector of Competitor strategy values (0-100) <b>S</b> Numeric vector of Stress-tolerator strategy values (0-100) <b>R</b> Numeric vector of Ruderal strategy values (0-100) <b>type</b> 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)
```

---

forest\_invader\_traits *Forest Woody Invader Functional Traits Dataset*

---

### Description

A comprehensive functional trait dataset comparing native and invasive woody species in North American deciduous forests. This dataset includes 76 species (44 native, 32 non-native) with 68 continuous functional traits across leaf, stem, root, and whole-plant characteristics. The data has been processed using probabilistic PCA (PPCA) for missing value imputation and includes log-transformed variables for right-skewed distributions.

### Usage

```
forest_invader_traits
```

### Format

A data frame with 76 rows (species) and 73 columns:

**species** Species name (character)

**family** Taxonomic family (character)

**growthform** Growth form classification: shrub, tree, or vine (character)

**nativity** Binary coding: 0 = native, 1 = non-native (numeric)

**status** Invasion status: "native", "naturalized", or "invasive" (character)

**Mass\_leaf** Leaf dry mass (standardized, log-transformed)

**SLA** Specific leaf area (standardized)  
**LDMC** Leaf dry matter content (standardized)  
**Lobe** Leaf shape complexity (standardized, log-transformed)  
**PA\_leaf** Leaf perimeter-area ratio (standardized, log-transformed)  
**Thick\_leaf** Leaf thickness (standardized, log-transformed)  
**Stomate\_size** Stomatal guard cell length (standardized)  
**Stomate\_index** Stomatal density index (standardized, log-transformed)  
**Rd\_area, Rd\_mass** Leaf dark respiration per area and mass (standardized, log-transformed)  
**Asat\_area, Asat\_mass** Light-saturated photosynthetic rate per area and mass (standardized, log-transformed)  
**Vc\_max** Maximum carboxylation rate (standardized, log-transformed)  
**J\_max** Maximum electron transport rate (standardized, log-transformed)  
**AQY** Apparent quantum yield (standardized, log-transformed)  
**LSP** Light saturation point (standardized, log-transformed)  
**PNUE** Photosynthetic nitrogen use efficiency (standardized, log-transformed)  
**WUE** Water use efficiency (standardized, log-transformed)  
**CC\_area, CC\_mass** Leaf construction cost per area and mass (standardized, log-transformed)  
**N\_mass, N\_area, N\_litter** Leaf nitrogen content (standardized, log-transformed)  
**N\_res** Nitrogen resorption efficiency (standardized)  
**C\_mass, C\_area, C\_litter** Leaf carbon content (standardized, log-transformed)  
**Solubles, Hemi, Lignin, Cellulose** Leaf chemical composition (standardized, log-transformed)  
**Ash** Leaf ash content (standardized, log-transformed)  
**Phenolics** Leaf phenolic content (standardized, log-transformed)  
**Chl\_area, Chl\_mass** Chlorophyll content per area and mass (standardized, log-transformed)  
**Chl\_ab** Chlorophyll a:b ratio (standardized, log-transformed)  
**Decomp\_leaf** Leaf decomposition rate (standardized, log-transformed)  
**DNA** Nuclear DNA content (standardized, log-transformed)  
**SRL** Specific root length (standardized, log-transformed)  
**RTD, RDMC** Root tissue density and dry matter content (standardized, log-transformed)  
**N\_root, C\_root** Root nitrogen and carbon content (standardized, log-transformed)  
**Lignin\_root** Root lignin content (standardized, log-transformed)  
**Decomp\_root** Root decomposition rate (standardized, log-transformed)  
**LAgain** Annual leaf area gain (standardized, log-transformed)  
**LeafRate\_max** Maximum leaf production rate (standardized, log-transformed)  
**NPP\_root** Annual root production (standardized, log-transformed)  
**SS\_root, SS\_stem** Soluble sugar content in roots and stems (standardized, log-transformed)  
**Starch\_root, Starch\_stem** Starch content in roots and stems (standardized, log-transformed)

**NSC\_root, NSC\_stem** Non-structural carbohydrate content (standardized, log-transformed)

**Vessel\_freq** Xylem vessel frequency (standardized, log-transformed)

**Metaxylem\_diam, Earlywood\_diam, Latewood\_diam** Xylem vessel diameters (standardized, log-transformed)

**Cond\_stem** Stem hydraulic conductance (standardized, log-transformed)

**LeafFall50** Median date of 50% leaf senescence (standardized)

**Lifespan\_leaf** Mean leaf lifespan (standardized, log-transformed)

**EmergeDuration** Duration of spring leaf emergence (standardized)

**LeafDuration** Annual duration of live leaves (standardized, log-transformed)

**LeafTurnover** Annual leaf turnover rate (standardized)

**FallDuration** Duration of autumn senescence (standardized)

**Chl\_sun50, Chl\_shade50** Date of 50% chlorophyll loss in sun/shade leaves (standardized)

**LeafOut** Date of spring leaf emergence (standardized)

## Details

This dataset represents measurements from a common garden experiment conducted at Syracuse University from 2006-2018. All continuous traits have been standardized to zero mean and unit variance, with 37 right-skewed traits log-transformed prior to standardization. Missing values (29% of the original dataset) were imputed using probabilistic PCA with 2 principal components.

The study identified a unique "fast but steady" trait syndrome in invasive species, combining high metabolic rates with extended leaf duration and robust leaf construction. Two traits - annual leaf duration and nuclear DNA content - distinguished native from invasive species with 93% accuracy.

## Source

Common garden experiment, Syracuse University, New York (43°39'N, 76°99'W). Species grown under 20% light transmittance shade cloth simulating forest understory conditions.

## References

Fridley, J. D., Bauerle, T. L., Craddock, A., Ebert, A. R., Frank, D. A., Heberling, J. M., Hinman, E. D., Jo, I., Martinez, K. A., Smith, M. S., Woolhiser, L. J., & Yin, J. (2022). Fast but steady: An integrated leaf-stem-root trait syndrome for woody forest invaders. *Ecology Letters*, 25(4), 900-912. doi:10.1111/ele.13967

## Examples

```
data(forest_invader_traits)
head(forest_invader_traits)
```

---

forest\_invader\_tree    *Phylogenetic tree for forest invader trait analysis*

---

### Description

A phylogenetic tree object containing 76 woody plant species (44 native and 32 non-native) used in the analysis of functional trait syndromes for forest invaders in North American deciduous forests. The tree is based on species-level phylogeny with aged branch lengths and includes native, naturalized, and invasive species classifications.

### Usage

forest\_invader\_tree

### Format

A phylo object with:

**edge** A 142 x 2 matrix defining the tree topology

**edge.length** A numeric vector of 142 branch lengths

**Nnode** Number of internal nodes (67)

**node.label** Character vector with node labels including taxonomic groups

**tip.label** Character vector of 76 species names

### Details

The phylogenetic tree was constructed using correlation matrix of phylogenetic distances calculated with an aged tree based on the species-level phylogeny. The tree includes representatives from major angiosperm families and is used for phylogenetically independent contrasts in trait analysis. Branch lengths reflect evolutionary time, and the tree structure allows for proper phylogenetic corrections in comparative analyses of functional traits between native and invasive species.

Species include shrubs and vines from deciduous forests, with invasive species representing those managed as high-impact invaders in northeastern United States, and naturalized species representing non-native species that have established but are not considered highly invasive.

### Source

Tree construction based on Jo et al. (2016) phylogeny with additional species placement using updated familial/generic phylogenies.

### References

Fridley, J. D., Bauerle, T. L., Craddock, A., Ebert, A. R., Frank, D. A., Heberling, J. M., Hinman, E. D., Jo, I., Martinez, K. A., Smith, M. S., Woolhiser, L. J., & Yin, J. (2022). Fast but steady: An integrated leaf-stem-root trait syndrome for woody forest invaders. *Ecology Letters*, 25(4), 900-912. doi:10.1111/ele.13967

**Examples**

```
data(forest_invader_tree)
print(forest_invader_tree)
```

---

interlayer\_closeness *Calculate Interlayer Closeness for Plant Trait Multilayer Network Nodes*

---

**Description**

Computes the interlayer closeness (IC) for each node in a plant trait multilayer network (PTMN). Interlayer closeness measures how efficiently a node can reach nodes in other layers along the shortest possible path, serving as an indicator of its capacity for cross-layer information or resource transfers. Nodes with high IC can efficiently influence the entire multilayer network and play a central role in rapid, coordinated system-wide responses in dynamic environments, thus facilitating swift phenotypic adjustments.

**Usage**

```
interlayer_closeness(data)
```

**Arguments**

`data` A data frame containing network edge information with required columns: `node.from`, `layer.from`, `node.to`, `layer.to`

**Details**

The interlayer closeness is calculated using the formula:

$$IC_i = \frac{n_{inter}^i}{\sum_{\beta \neq \alpha} \sum_{j \in \beta} d_{ij}}$$

where  $d_{ij}$  is the shortest path length from node  $v_i$  to node  $v_j$  in another layer and  $n_{inter}^i$  is the total number of reachable nodes in other layers.

This parameter is part of a comprehensive set of quantitative metrics for plant trait multilayer networks (PTMNs) that facilitate the effective identification of hub traits and key cross-layer functional modules. These metrics are essential for understanding the coordinated adaptation of plant traits across functional levels.

**Value**

A data frame with two columns:

`node` Character vector of node names  
`interlayer_closeness` Numeric vector of interlayer closeness values, sorted in descending order

**See Also**

[PTMN](#) for constructing plant trait multilayer networks

**Examples**

```
## Not run:
data(forest_invader_tree)
data(forest_invader_traits)
traits <- forest_invader_traits[, 6:73]
layers <- list(
  shoot_dynamics = c("LeafDuration", "LeafFall50", "LeafRate_max",
    "Chl_shade50", "LAgain", "FallDuration",
    "LeafOut", "Chl_sun50", "EmergeDuration",
    "LeafTurnover"),
  leaf_structure = c("PA_leaf", "Mass_leaf", "Lifespan_leaf",
    "Thick_leaf", "SLA", "Lobe", "LDMC",
    "Stomate_size", "Stomate_index"),
  leaf_metabolism = c("J_max", "Vc_max", "Asat_area", "CC_mass",
    "LSP", "AQY", "CC_area", "Rd_area",
    "Asat_mass", "WUE", "Rd_mass", "PNUE"),
  leaf_chemistry = c("N_area", "Chl_area", "DNA", "Phenolics",
    "Cellulose", "N_mass", "N_litter", "Chl_ab",
    "Chl_mass", "N_res", "C_litter", "C_area",
    "C_mass", "Ash", "Lignin", "Solubles",
    "Decomp_leaf", "Hemi"),
  root = c("NPP_root", "SS_root", "SRL", "RTD", "RDMC",
    "NSC_root", "Decomp_root", "Starch_root",
    "C_root", "N_root", "Lignin_root"),
  stem = c("Latewood_diam", "Metaxylem_diam", "Earlywood_diam",
    "NSC_stem", "Vessel_freq", "SS_stem", "Cond_stem",
    "Starch_stem")
)
graph <- PTMN(traits, layers_list = layers, method = "pearson")
interlayer_closeness(graph)

## End(Not run)
```

---

```
interlayer_clustering_coefficient
```

*Calculate Interlayer Clustering Coefficient for Plant Trait Multilayer Networks (PTMNs)*

---

**Description**

This function calculates the interlayer clustering coefficient (ICC) for each node in a plant trait multilayer network. The ICC measures interconnection among a node's interlayer neighbors, indicating the node's capacity to form cooperative, cross-layer functional modules within the local network.

**Usage**

```
interlayer_clustering_coefficient(data)
```

**Arguments**

**data** A data frame containing multilayer network edge information with required columns: `node.from`, `layer.from`, `node.to`, and `layer.to`. This is typically the output from the [PTMN](#) function.

**Details**

The interlayer clustering coefficient is defined as:

$$ICC_i = \frac{2E_i}{IK_i(K_i - 1)}$$

where  $E_i$  is the number of actual edges among the node's interlayer neighbors and  $IK_i$  is the interlayer degree.

For nodes with an interlayer degree of zero or one, ICC is defined as zero. Nodes with high ICC act as "organizers" of highly integrated, cross-functional modules, enhancing system-level responsiveness and phenotypic coordination.

The function constructs PTMNs as undirected networks, reflecting the reciprocal nature of trait relationships in plants.

**Value**

A data frame with the following columns:

**node** Character. The node identifier (trait name)

**layer** Character. The layer identifier (functional category)

**interlayer\_degree** Numeric. Number of interlayer connections for the node

**actual\_connections** Numeric. Actual number of connections between interlayer neighbors

**potential\_connections** Numeric. Maximum possible connections between interlayer neighbors

**interlayer\_clustering\_coefficient** Numeric. The ICC value calculated as `actual_connections / potential_connections`

The results are ordered by decreasing interlayer clustering coefficient values.

**See Also**

[PTMN](#) for constructing plant trait multilayer networks

**Examples**

```
## Not run:
data(forest_invader_tree)
data(forest_invader_traits)
traits <- forest_invader_traits[, 6:73]
layers <- list(
```

```

shoot_dynamics = c("LeafDuration", "LeafFall50", "LeafRate_max",
                  "Chl_shade50", "LAgain", "FallDuration",
                  "LeafOut", "Chl_sun50", "EmergeDuration",
                  "LeafTurnover"),
leaf_structure = c("PA_leaf", "Mass_leaf", "Lifespan_leaf",
                  "Thick_leaf", "SLA", "Lobe", "LDMC",
                  "Stomate_size", "Stomate_index"),
leaf_metabolism = c("J_max", "Vc_max", "Asat_area", "CC_mass",
                   "LSP", "AQY", "CC_area", "Rd_area",
                   "Asat_mass", "WUE", "Rd_mass", "PNUE"),
leaf_chemistry = c("N_area", "Chl_area", "DNA", "Phenolics",
                  "Cellulose", "N_mass", "N_litter", "Chl_ab",
                  "Chl_mass", "N_res", "C_litter", "C_area",
                  "C_mass", "Ash", "Lignin", "Solubles",
                  "Decomp_leaf", "Hemi"),
root = c("NPP_root", "SS_root", "SRL", "RTD", "RDMC",
         "NSC_root", "Decomp_root", "Starch_root",
         "C_root", "N_root", "Lignin_root"),
stem = c("Latewood_diam", "Metaxylem_diam", "Earlywood_diam",
         "NSC_stem", "Vessel_freq", "SS_stem", "Cond_stem",
         "Starch_stem")
)
graph <- PTMN(traits, layers_list = layers, method = "pearson")
interlayer_clustering_coefficient(graph)

## End(Not run)

```

---

interlayer_degree	<i>Calculate Interlayer Degree for Nodes in Plant Trait Multilayer Networks</i>
-------------------	---

---

### Description

This function calculates the interlayer degree for each node in a Plant Trait Multilayer Network (PTMN). The interlayer degree represents the number of connections a node has to nodes in different layers, which is crucial for understanding cross-layer coordination and functional integration across plant organs.

### Usage

```
interlayer_degree(data)
```

### Arguments

data	A data frame containing multilayer network edge information with columns: node.from, layer.from, node.to, layer.to. This represents the extended edge list from a PTMN construction.
------	--

## Details

In PTMNs, interlayer edges represent interactions between traits in different layers, illustrating the integration of multiple functional systems. Nodes with higher interlayer degrees indicate traits that are more integrated across plant organs or functional systems, suggesting coordinated adaptation mechanisms.

The function counts both incoming and outgoing interlayer connections for each node, as PTMNs are constructed as undirected networks reflecting the reciprocal nature of trait relationships in plants.

## Value

A data frame with three columns:

node	Character. The name of the trait node
layer	Character. The layer (organ or functional system) containing the node
interlayer_degree	Numeric. The number of interlayer connections for each node

The results are ordered by interlayer degree in descending order.

## See Also

[PTMN](#) for constructing plant trait multilayer networks

## Examples

```
## Not run:
data(forest_invader_tree)
data(forest_invader_traits)
traits <- forest_invader_traits[, 6:73]
layers <- list(
  shoot_dynamics = c("LeafDuration", "LeafFall50", "LeafRate_max",
                    "Chl_shade50", "LAgain", "FallDuration",
                    "LeafOut", "Chl_sun50", "EmergeDuration",
                    "LeafTurnover"),
  leaf_structure = c("PA_leaf", "Mass_leaf", "Lifespan_leaf",
                    "Thick_leaf", "SLA", "Lobe", "LDMC",
                    "Stomate_size", "Stomate_index"),
  leaf_metabolism = c("J_max", "Vc_max", "Asat_area", "CC_mass",
                     "LSP", "AQY", "CC_area", "Rd_area",
                     "Asat_mass", "WUE", "Rd_mass", "PNUE"),
  leaf_chemistry = c("N_area", "Chl_area", "DNA", "Phenolics",
                    "Cellulose", "N_mass", "N_litter", "Chl_ab",
                    "Chl_mass", "N_res", "C_litter", "C_area",
                    "C_mass", "Ash", "Lignin", "Solubles",
                    "Decomp_leaf", "Hemi"),
  root = c("NPP_root", "SS_root", "SRL", "RTD", "RDMC",
           "NSC_root", "Decomp_root", "Starch_root",
           "C_root", "N_root", "Lignin_root"),
  stem = c("Latewood_diam", "Metaxylem_diam", "Earlywood_diam",
           "NSC_stem", "Vessel_freq", "SS_stem", "Cond_stem",
           "Starch_stem")
```

```

)
graph <- PTMN(traits, layers_list = layers, method = "pearson")
interlayer_degree(graph)

## End(Not run)

```

---

interlayer\_edge\_density

*Calculate Interlayer Edge Density for Plant Trait Multilayer Networks*

---

### Description

This function calculates the interlayer edge density (IED) for a plant trait multilayer network (PTMN). The IED measures the ratio of observed to maximally possible interlayer edges, indicating the extent of direct connections among functional layers. A higher IED indicates greater interdependency among layers, reflecting stronger overall network integration.

### Usage

```
interlayer_edge_density(data)
```

### Arguments

data	A data frame containing edge information for the multilayer network. The data frame must contain at least four columns: <code>layer.from</code> , <code>layer.to</code> , <code>node.from</code> , and <code>node.to</code> , representing the source layer, target layer, source node, and target node for each edge, respectively.
------	--

### Details

The interlayer edge density is calculated using the formula:

$$IED = \frac{\sum_{\alpha=1}^{M-1} \sum_{\beta=\alpha+1}^M L_{inter}^{\alpha,\beta}}{\sum_{\alpha=1}^{M-1} \sum_{\beta=\alpha+1}^M n_{\alpha} \cdot n_{\beta}}$$

where  $M$  is the total number of layers,  $L_{inter}^{\alpha,\beta}$  is the number of observed interlayer edges between layers  $\alpha$  and  $\beta$ , and  $n_{\alpha}$  and  $n_{\beta}$  are their respective node counts.

This parameter is one of the specially designed topological parameters for quantifying PTMN topology, facilitating the effective identification of hub traits and key cross-layer functional modules essential for understanding coordinated adaptation of plant traits across functional levels.

### Value

A numeric value representing the interlayer edge density, calculated as the ratio of actual interlayer edges to the maximum possible number of interlayer edges between all layer pairs.

**See Also**

[PTMN](#) for constructing plant trait multilayer networks

**Examples**

```
## Not run:
data(forest_invader_tree)
data(forest_invader_traits)
traits <- forest_invader_traits[, 6:73]
layers <- list(
  shoot_dynamics = c("LeafDuration", "LeafFall50", "LeafRate_max",
                    "Chl_shade50", "LAgain", "FallDuration",
                    "LeafOut", "Chl_sun50", "EmergeDuration",
                    "LeafTurnover"),
  leaf_structure = c("PA_leaf", "Mass_leaf", "Lifespan_leaf",
                    "Thick_leaf", "SLA", "Lobe", "LDMC",
                    "Stomate_size", "Stomate_index"),
  leaf_metabolism = c("J_max", "Vc_max", "Asat_area", "CC_mass",
                     "LSP", "AQY", "CC_area", "Rd_area",
                     "Asat_mass", "WUE", "Rd_mass", "PNUE"),
  leaf_chemistry = c("N_area", "Chl_area", "DNA", "Phenolics",
                    "Cellulose", "N_mass", "N_litter", "Chl_ab",
                    "Chl_mass", "N_res", "C_litter", "C_area",
                    "C_mass", "Ash", "Lignin", "Solubles",
                    "Decomp_leaf", "Hemi"),
  root = c("NPP_root", "SS_root", "SRL", "RTD", "RDMC",
           "NSC_root", "Decomp_root", "Starch_root",
           "C_root", "N_root", "Lignin_root"),
  stem = c("Latewood_diam", "Metaxylem_diam", "Earlywood_diam",
           "NSC_stem", "Vessel_freq", "SS_stem", "Cond_stem",
           "Starch_stem")
)
graph <- PTMN(traits, layers_list = layers, method = "pearson")
interlayer_edge_density(graph)

## End(Not run)
```

---

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 mm<sup>2</sup>/mg or m<sup>2</sup>/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	<p>A data frame containing LHS analysis results with the following required columns:</p> <p><b>log_SLA</b> Natural logarithm of specific leaf area</p> <p><b>log_Height</b> Natural logarithm of canopy height at maturity</p> <p><b>log_SeedMass</b> Natural logarithm of seed mass</p> <p>Typically this would be the output from the <code>LHS()</code> function. Row names should represent species names or identifiers.</p>
group	<p>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.</p>
show_cube	<p>Logical indicating whether to display the 3D cube frame around the plot. Default is TRUE, which helps with spatial orientation and depth perception.</p>
colors	<p>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.</p>
cube_angle	<p>Numeric value specifying the viewing angle for the 3D plot in degrees. Default is 60. Values between 40-80 typically provide good visualisation perspectives.</p>

**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:  $\log(\text{SLA})$  - reflects the leaf economics spectrum from resource-conservative to resource-acquisitive strategies
- Y-axis:  $\log(\text{Height})$  - represents the plant size spectrum and competitive ability for light capture
- Z-axis:  $\log(\text{Seed Mass})$  - 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 <code>vegan::rda()</code> 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 <code>NPT_continuous()</code> \$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: <ul style="list-style-type: none"> <li>"CART" - Classification and Regression Trees (default)</li> <li>"kmeans" - K-means clustering with automatic k selection</li> </ul>
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:

<code>niche_classification</code>	A data frame with species names, cluster assignments for each dimension, and comprehensive niche codes
<code>summary</code>	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

<code>data</code>	A data frame containing niche classification results, typically the <code>niche_classification</code> output from <code>NPT_discrete</code> . Must contain: <ul style="list-style-type: none"> <li>• <code>species</code> - Character column with species names</li> <li>• <code>niche_code</code> - Character column with comma-separated niche codes (e.g., "1,2,1")</li> <li>• Additional columns representing cluster assignments for each dimension (used as layer names in the visualization)</li> </ul>
<code>point_size</code>	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<sup>2</sup>)  
**LDMC** Leaf dry matter content (%)  
**SLA** Specific leaf area (mm<sup>2</sup>/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")
```

**Description**

Constructs a phylogenetic trait multilayer network by calculating correlations between traits while accounting for phylogenetic relationships. The function creates a network where nodes represent traits organized into different layers, and edges represent significant correlations between traits.

**Usage**

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

**Arguments**

<code>traits_matrix</code>	A data frame or matrix containing trait values where rows represent species and columns represent traits. Row names should match the tip labels in the phylogenetic tree when phylogenetic correction is used.
<code>layers_list</code>	A named list where each element contains the names of traits belonging to that layer. Names of the list elements represent layer names.
<code>rThres</code>	Numeric. Correlation threshold for filtering weak correlations. Correlations with absolute values below this threshold are set to zero. Default is 0.2.
<code>pThres</code>	Numeric. P-value threshold for statistical significance after FDR correction. Correlations with p-values above this threshold are excluded. Default is 0.05.
<code>method</code>	Character. Correlation method to use. Either "pearson" or "spearman". Default is "pearson".
<code>phylo_correction</code>	Logical. Whether to apply phylogenetic correction using phylogenetic independent contrasts. Default is FALSE.
<code>phylo_tree</code>	A phylo object from the ape package. Required when <code>phylo_correction = TRUE</code> . Should contain all species present in <code>traits_matrix</code> .

**Details**

The function performs the following steps:

1. Validates input parameters and data consistency
2. Calculates correlation matrix using standard methods or phylogenetic independent contrasts
3. Filters correlations based on correlation and p-value thresholds
4. Applies FDR correction to p-values
5. Constructs an igraph network object
6. Maps traits to their respective layers
7. Returns edge list with layer information

When phylogenetic correction is enabled, the function uses phylogenetic independent contrasts to account for the non-independence of species data due to shared evolutionary history.

**Value**

A data frame containing the network edges with the following columns:

- `node.from`: Name of the source trait
- `layer.from`: Layer name of the source trait
- `node.to`: Name of the target trait
- `layer.to`: Layer name of the target trait
- `correlation`: Correlation coefficient between the traits

Returns an empty data frame with the same structure if no significant correlations are found.

**Examples**

```

## Not run:
# Load example data
data(forest_invader_tree)
data(forest_invader_traits)
traits <- forest_invader_traits[, 6:73]

# Define trait layers
layers <- list(
  shoot_dynamics = c("LeafDuration", "LeafFall50", "LeafRate_max",
                    "Chl_shade50", "LAgain", "FallDuration",
                    "LeafOut", "Chl_sun50", "EmergeDuration",
                    "LeafTurnover"),
  leaf_structure = c("PA_leaf", "Mass_leaf", "Lifespan_leaf",
                    "Thick_leaf", "SLA", "Lobe", "LDMC",
                    "Stomate_size", "Stomate_index"),
  leaf_metabolism = c("J_max", "Vc_max", "Asat_area", "CC_mass",
                     "LSP", "AQY", "CC_area", "Rd_area",
                     "Asat_mass", "WUE", "Rd_mass", "PNUE"),
  leaf_chemistry = c("N_area", "Chl_area", "DNA", "Phenolics",
                     "Cellulose", "N_mass", "N_litter", "Chl_ab",
                     "Chl_mass", "N_res", "C_litter", "C_area",
                     "C_mass", "Ash", "Lignin", "Solubles",
                     "Decomp_leaf", "Hemi"),
  root = c("NPP_root", "SS_root", "SRL", "RTD", "RDMC",
           "NSC_root", "Decomp_root", "Starch_root",
           "C_root", "N_root", "Lignin_root"),
  stem = c("Latewood_diam", "Metaxylem_diam", "Earlywood_diam",
           "NSC_stem", "Vessel_freq", "SS_stem", "Cond_stem",
           "Starch_stem")
)
# Run PTMN analysis without phylogenetic correction
PTMN(traits, layers_list = layers, method = "pearson")

# Run PTMN analysis with phylogenetic correction
PTMN(traits, layers_list = layers, method = "pearson",
     phylo_correction = TRUE, phylo_tree = forest_invader_tree)

## End(Not run)

```

**Description**

This function computes comprehensive network metrics for plant trait multilayer networks, including both node-level and global network properties. It quantifies the structural characteristics and connectivity patterns that reveal coordinated adaptation mechanisms across multiple plant organs and functional systems.

**Usage**

```
PTMN_metrics(graph)
```

**Arguments**

**graph** A multilayer network object created by the [PTMN](#) function, representing trait relationships across different plant organs or functional systems.

**Details**

The PTMN framework addresses limitations of traditional single-layer plant trait networks by capturing trait relationships across multiple plant organs and functional systems. These metrics quantify network topology using specially designed parameters that facilitate identification of hub traits and key cross-layer functional modules, essential for understanding coordinated adaptation of plant traits.

Node-level metrics help identify traits that serve as integration points across different plant systems, while global metrics characterize the overall network structure and connectivity patterns. Higher interlayer connectivity may indicate greater functional integration and potentially enhanced adaptive capacity.

**Value**

A list containing two data frames:

**node** Node-level metrics including:

- **node**: Name of the trait node
- **layer**: Layer (organ/functional system) containing the node
- **interlayer\_degree**: Number of connections a node has across layers
- **interlayer\_closeness**: Measure of how close a node is to all other nodes across layers
- **interlayer\_clustering\_coefficient**: Local clustering coefficient considering interlayer connections

**global** Global network metrics including:

- **Interlayer\_edge\_density**: Proportion of possible interlayer connections that are realized
- **Average\_interlayer\_path\_length**: Average shortest path length between nodes across layers
- **Average\_interlayer\_clustering\_coefficient**: Network-wide clustering coefficient
- **Module\_interlayer\_association**: Degree of modular organization across layers

**See Also**

[PTMN](#) for constructing plant trait multilayer networks

**Examples**

```

## Not run:
data(forest_invader_tree)
data(forest_invader_traits)
traits <- forest_invader_traits[, 6:73]
layers <- list(
  shoot_dynamics = c("LeafDuration", "LeafFall50", "LeafRate_max",
    "Chl_shade50", "LAgain", "FallDuration",
    "LeafOut", "Chl_sun50", "EmergeDuration",
    "LeafTurnover"),
  leaf_structure = c("PA_leaf", "Mass_leaf", "Lifespan_leaf",
    "Thick_leaf", "SLA", "Lobe", "LDMC",
    "Stomate_size", "Stomate_index"),
  leaf_metabolism = c("J_max", "Vc_max", "Asat_area", "CC_mass",
    "LSP", "AQY", "CC_area", "Rd_area",
    "Asat_mass", "WUE", "Rd_mass", "PNUE"),
  leaf_chemistry = c("N_area", "Chl_area", "DNA", "Phenolics",
    "Cellulose", "N_mass", "N_litter", "Chl_ab",
    "Chl_mass", "N_res", "C_litter", "C_area",
    "C_mass", "Ash", "Lignin", "Solubles",
    "Decomp_leaf", "Hemi"),
  root = c("NPP_root", "SS_root", "SRL", "RTD", "RDMC",
    "NSC_root", "Decomp_root", "Starch_root",
    "C_root", "N_root", "Lignin_root"),
  stem = c("Latewood_diam", "Metaxylem_diam", "Earlywood_diam",
    "NSC_stem", "Vessel_freq", "SS_stem", "Cond_stem",
    "Starch_stem")
)
graph <- PTMN(traits, layers_list = layers, method = "pearson")
PTMN_metrics(graph)

## End(Not run)

```

PTMN\_plot

*Plot Plant Trait Multilayer Networks (PTMN)***Description**

This function creates visualizations of plant trait multilayer networks (PTMNs) constructed using the PTMN framework. PTMNs systematically integrate multilayer network theory with plant functional trait analysis, enabling quantitative assessments of trait relationships across plant organs and functional systems.

**Usage**

```

PTMN_plot(
  data,
  style = 1,

```

```

vertex.size = 15,
show.labels = TRUE,
vertex.label.cex = 0.7,
vertex.label.dist = 0.2,
edge.width = 2,
vertex.label.font = 2,
node.alpha = 1,
module.alpha = 0.3,
show.legend = TRUE,
legend.pos = "bottomright",
legend.cex = 1,
legend.pt.size = 2.5,
x.intersp = 2,
y.intersp = 2,
title.font = 2,
title.cex = 1.2
)

```

### Arguments

<code>data</code>	A data frame containing the PTMN edge list with columns: <code>node.from</code> , <code>node.to</code> , <code>layer.from</code> , <code>layer.to</code> . This should be the output from the PTMN function.
<code>style</code>	Integer specifying the layout style. Options are: <ul style="list-style-type: none"> <li>• 1 - Standard network layout with cross-layer module highlighting (default)</li> <li>• 2 - Circular layout with nodes arranged by layer</li> </ul>
<code>vertex.size</code>	Numeric value controlling the size of network nodes (vertices). Default is 15.
<code>show.labels</code>	Logical indicating whether to display node labels. Default is TRUE.
<code>vertex.label.cex</code>	Numeric value controlling the size of vertex labels. Default is 0.7.
<code>vertex.label.dist</code>	Numeric value controlling the distance of labels from vertices. Default is 0.2.
<code>edge.width</code>	Numeric value controlling the width of network edges. Default is 2.
<code>vertex.label.font</code>	Integer specifying the font style for vertex labels (1=plain, 2=bold, 3=italic, 4=bold italic). Default is 2.
<code>node.alpha</code>	Numeric value between 0 and 1 controlling the transparency of nodes. Default is 1 (opaque).
<code>module.alpha</code>	Numeric value between 0 and 1 controlling the transparency of module highlighting. Default is 0.3.
<code>show.legend</code>	Logical indicating whether to display the layer legend. Default is TRUE.
<code>legend.pos</code>	Character string specifying legend position. Options include "bottomright", "bottomleft", "topright", "topleft". Default is "bottomright".
<code>legend.cex</code>	Numeric value controlling the size of legend text. Default is 1.
<code>legend.pt.size</code>	Numeric value controlling the size of legend symbols. Default is 2.5.

x.intersp	Numeric value controlling horizontal spacing in legend. Default is 2.
y.intersp	Numeric value controlling vertical spacing in legend. Default is 2.
title.font	Integer specifying the font style for legend title. Default is 2 (bold).
title.cex	Numeric value controlling the size of legend title. Default is 1.2.

### Details

The PTMN visualization distinguishes between intralayer edges (black lines) showing relationships between traits within the same organ or functional system, and interlayer edges (red lines) representing interactions between traits in different layers. Each layer corresponds to specific functional or structural units such as individual plant organs (leaves, stems, roots) or functional systems.

In style 1, cross-layer modules are highlighted with shaded areas, representing tightly connected functional groups that span multiple layers. In style 2, the circular layout arranges nodes by layer in a circular pattern, making layer organization more visually apparent.

Node colors are automatically assigned based on layer membership using the "ggsci::nrc\_npg" color palette, ensuring visual distinction between different functional layers.

### Value

No return value. This function is called for its side effect of creating a network plot.

### See Also

[PTMN](#) for constructing plant trait multilayer networks

### Examples

```
## Not run:
data(forest_invader_tree)
data(forest_invader_traits)
traits <- forest_invader_traits[, 6:73]
layers <- list(
  shoot_dynamics = c("LeafDuration", "LeafFall150", "LeafRate_max",
    "Chl_shade50", "LAgain", "FallDuration",
    "LeafOut", "Chl_sun50", "EmergeDuration",
    "LeafTurnover"),
  leaf_structure = c("PA_leaf", "Mass_leaf", "Lifespan_leaf",
    "Thick_leaf", "SLA", "Lobe", "LDMC",
    "Stomate_size", "Stomate_index"),
  leaf_metabolism = c("J_max", "Vc_max", "Asat_area", "CC_mass",
    "LSP", "AQY", "CC_area", "Rd_area",
    "Asat_mass", "WUE", "Rd_mass", "PNUE"),
  leaf_chemistry = c("N_area", "Chl_area", "DNA", "Phenolics",
    "Cellulose", "N_mass", "N_litter", "Chl_ab",
    "Chl_mass", "N_res", "C_litter", "C_area",
    "C_mass", "Ash", "Lignin", "Solubles",
    "Decomp_leaf", "Hemi"),
  root = c("NPP_root", "SS_root", "SRL", "RTD", "RDMC",
    "NSC_root", "Decomp_root", "Starch_root",
    "C_root", "N_root", "Lignin_root"),
```

```

    stem = c("Latewood_diam", "Metaxylem_diam", "Earlywood_diam",
             "NSC_stem", "Vessel_freq", "SS_stem", "Cond_stem",
             "Starch_stem")
  )
graph <- PTMN(traits, layers_list = layers, method = "pearson")
PTMN_plot(graph, style = 1, vertex.size = 8,
           vertex.label.cex = 0.5, edge.width = 2,
           show.legend = FALSE)

## End(Not run)

```

---

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

<code>traits_matrix</code>	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.
<code>rThres</code>	Numeric, threshold for correlation coefficient, default is 0.2. Correlations with absolute values below this threshold are set to zero.
<code>pThres</code>	Numeric, threshold for p-value, default is 0.05. Only correlations with p-values below this threshold are included in the network.
<code>method</code>	Character, specifies the correlation method to use: "pearson" (default) or "spearman".
<code>phylo_correction</code>	Logical, whether to apply phylogenetic correction using phylogenetic independent contrasts, default is FALSE.
<code>phylo_tree</code>	A phylo object from the ape package containing the phylogenetic tree. Required when <code>phylo_correction = TRUE</code> . Species names in the tree must match row names in <code>traits_matrix</code> .

## 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 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

```
# 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**

<code>graph</code>	An igraph object representing the trait network.
<code>style</code>	A numeric value that determines the plotting style (default is 1).
<code>vertex.size</code>	Numeric value for the size of vertices in the plot (default is 20).
<code>vertex.label.cex</code>	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 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)
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

- \* **datasets**
  - forest\_invader\_traits, 14
  - forest\_invader\_tree, 17
  - PFF, 37
  - PFF\_tree, 39
  
- average\_interlayer\_clustering\_coefficient, 2
- average\_interlayer\_path\_length, 4
  
- cross\_layer\_groups, 7
- crosslayer\_module\_assoc, 6
- CSR, 9, 14
- CSR\_hodgson, 11, 14
- CSR\_plot, 12
  
- forest\_invader\_traits, 14
- forest\_invader\_tree, 17
  
- interlayer\_closeness, 18
- interlayer\_clustering\_coefficient, 19
- interlayer\_degree, 21
- interlayer\_edge\_density, 23
  
- LHS, 24
- LHS\_plot, 26
- LHS\_strategy\_scheme, 28
  
- NPT\_continuous, 29
- NPT\_continuous\_plot, 31
- NPT\_discrete, 33, 35
- NPT\_discrete\_plot, 35
  
- PFF, 37
- PFF\_tree, 39
- phylo\_correlation, 40
- PTMN, 3, 5, 7, 8, 19, 20, 22, 24, 41, 44, 47
- PTMN\_metrics, 43
- PTMN\_plot, 45
- PTN, 48
- PTN\_corr, 50
- PTN\_metrics, 52
- PTN\_plot, 53