Package: Nematode (via r-universe)

November 3, 2026

Title Ecological Indices Calculator for Nematode Communities

Version 0.3.2

Date 2026-10-04

Description Provides a computational toolkit for analyzing nematode communities in ecological studies. Includes methods to quantify nematode-based ecological indicators such as metabolic footprints, energy flow metrics, and community structure. These tools support assessments of soil health, ecosystem functioning, and trophic interactions, standardizing the use of nematodes as bioindicators.

License GPL (>= 3)

Depends R (>= 3.5)

Imports dplyr (>= 1.1.4), purrr (>= 1.0.4), stats, stringdist (>= 0.9.15), utils, vegan (>= 2.7-1)

BuildManual no

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

RoxygenNote 7.3.3

Repository https://h-xuanjiu.r-universe.dev

Date/Publication 2026-10-04 14:37:54 UTC

RemoteUrl https://github.com/h-xuanjiu/nematode

RemoteRef HEAD

RemoteSha bb97869eb6088ff35d8418a21b5ce415fcf5f923

Contents

cal.BI . . . 2
cal.CI . . . 3
cal.EI . . . 4
cal.H . . . 5
cal.J . . . 6
cal.MI . . . 7
cal.NCR . . . 8
cal.PPI . . . 9
cal.SI . . . 10
cal.Simpson . . . 11
cal.SRI . . . 12
cal.TD . . . 13
cal.WI . . . 14
check_nematode_genus . . . 15
cp_rel_abundance . . . 16
diet_rel_abundance . . . 18
Ecological.Indices . . . 19
fuzzy_genus_match . . . 21
NEF . . . 22
nematode.ave.mass . . . 24
nematode.info . . . 25
NMF . . . 26
num_species . . . 27
rel_abundance . . . 28
runNMDS . . . 29
runPCoA . . . 32
runSimper . . . 35
summary.NMDS . . . 36

Index **38**

`cal.BI` *Calculate Basic Index (BI)*

Description

This function calculates the Basic Index (BI) for ecological communities.

Usage

```
cal.BI(data, ...)

## S3 method for class 'data.frame'
cal.BI(data, ...)

## S3 method for class 'matrix'
```

```
cal.BI(data, ...)

## Default S3 method:
cal.BI(data, ...)
```

Arguments

data	data.frame or matrix. The nematode abundance table where rows represent samples and columns represent nematode genera. Each element indicates the count of a specific nematode genus in the corresponding sample. Row names must be sample names, and column names must be nematode genus names.
...	Additional arguments (currently unused).

Value

A data frame with two columns:

Sample.ID	Character vector of sample identifiers (from row names of data)
BI	Basic Index for each sample

Examples

```
# Example with a data frame
df <- data.frame(
  Cephalobus = c(10, NA, 15),
  Caenorhabditis = c(5, 10, NA),
  Pratylenchus = c(8, 12, 10),
  row.names = c("A", "B", "C")
)
cal.BI(data = df)
```

---

cal.CI *Calculate Channel Index (CI)*

Description

This function calculates the Channel Index (CI) for ecological communities.

Usage

```
cal.CI(data, ...)

## S3 method for class 'data.frame'
cal.CI(data, ...)

## S3 method for class 'matrix'
cal.CI(data, ...)

## Default S3 method:
cal.CI(data, ...)
```

Arguments

data	data.frame or matrix. The nematode abundance table where rows represent samples and columns represent nematode genera. Each element indicates the count of a specific nematode genus in the corresponding sample. Row names must be sample names, and column names must be nematode genus names.
...	Additional arguments (currently unused).

Value

A data frame with two columns:

Sample.ID	Character vector of sample identifiers (from row names of data)
CI	Channel Index for each sample

Examples

```
# Example with a data frame
df <- data.frame(
  Cephalobus = c(10, NA, 15),
  Caenorhabditis = c(5, 10, NA),
  Pratylenchus = c(8, 12, 10),
  row.names = c("A", "B", "C")
)
cal.CI(data = df)
```

---

cal.EI *Calculate Enrichment Index (EI)*

Description

This function calculates the Enrichment Index (EI) for ecological communities.

Usage

```
cal.EI(data, ...)

## S3 method for class 'data.frame'
cal.EI(data, ...)

## S3 method for class 'matrix'
cal.EI(data, ...)

## Default S3 method:
cal.EI(data, ...)
```

Arguments

data	data.frame or matrix. The nematode abundance table where rows represent samples and columns represent nematode genera. Each element indicates the count of a specific nematode genus in the corresponding sample. Row names must be sample names, and column names must be nematode genus names.
...	Additional arguments (currently unused).

Value

A data frame with two columns:

Sample.ID	Character vector of sample identifiers (from row names of data)
EI	Enrichment Index for each sample

Examples

```
# Example with a data frame
df <- data.frame(
  Cephalobus = c(10, NA, 15),
  Caenorhabditis = c(5, 10, NA),
  Pratylenchus = c(8, 12, 10),
  row.names = c("A", "B", "C")
)
cal.EI(data = df)
```

---

cal.H *Calculate Shannon-Wiener Index (H)*

Description

This function calculates the Shannon-Wiener Index (H) for ecological communities.

Usage

```
cal.H(data, ...)

## S3 method for class 'data.frame'
cal.H(data, ...)

## S3 method for class 'matrix'
cal.H(data, ...)

## Default S3 method:
cal.H(data, ...)
```

Arguments

data	data.frame or matrix. The nematode abundance table where rows represent samples and columns represent nematode genera. Each element indicates the count of a specific nematode genus in the corresponding sample. Row names must be sample names, and column names must be nematode genus names.
...	Additional arguments (currently unused).

Value

A data frame with two columns:

Sample.ID	Character vector of sample identifiers (from row names of data)
H	Shannon-Wiener Index for each sample

Examples

```
# Example with a data frame
df <- data.frame(
  Cephalobus = c(10, NA, 15),
  Caenorhabditis = c(5, 10, NA),
  Pratylenchus = c(8, 12, 10),
  row.names = c("A", "B", "C")
)
cal.H(data = df)
```

---

cal.J *Calculate Pielou's Evenness Index (J)*

Description

This function calculates the Pielou's Evenness Index (J) for ecological communities.

Usage

```
cal.J(data, ...)

## S3 method for class 'data.frame'
cal.J(data, ...)

## S3 method for class 'matrix'
cal.J(data, ...)

## Default S3 method:
cal.J(data, ...)
```

Arguments

data	data.frame or matrix. The nematode abundance table where rows represent samples and columns represent nematode genera. Each element indicates the count of a specific nematode genus in the corresponding sample. Row names must be sample names, and column names must be nematode genus names.
...	Additional arguments (currently unused).

Value

A data frame with two columns:

Sample.ID	Character vector of sample identifiers (from row names of data)
J	Pielou's Evenness Index for each sample

Examples

```
# Example with a data frame
df <- data.frame(
  Cephalobus = c(10, NA, 15),
  Caenorhabditis = c(5, 10, NA),
  Pratylenchus = c(8, 12, 10),
  row.names = c("A", "B", "C")
)
cal.J(data = df)
```

---

cal.MI *Calculate Maturity Index (MI)*

Description

This function calculates the Maturity Index (MI) for ecological communities.

Usage

```
cal.MI(data, ...)

## S3 method for class 'data.frame'
cal.MI(data, ...)

## S3 method for class 'matrix'
cal.MI(data, ...)

## Default S3 method:
cal.MI(data, ...)
```

Arguments

data	data.frame or matrix. The nematode abundance table where rows represent samples and columns represent nematode genera. Each element indicates the count of a specific nematode genus in the corresponding sample. Row names must be sample names, and column names must be nematode genus names.
...	Additional arguments (currently unused).

Value

A data frame with two columns:

Sample.ID	Character vector of sample identifiers (from row names of data)
MI	Maturity Index for each sample

Examples

```
# Example with a data frame
df <- data.frame(
  Cephalobus = c(10, NA, 15),
  Caenorhabditis = c(5, 10, NA),
  Pratylenchus = c(8, 12, 10),
  row.names = c("A", "B", "C")
)
cal.MI(data = df)
```

---

cal.NCR *Calculate Nematode Channel Ratio (NCR)*

Description

This function calculates the Nematode Channel Ratio (NCR) for ecological communities.

Usage

```
cal.NCR(data, ...)

## S3 method for class 'data.frame'
cal.NCR(data, ...)

## S3 method for class 'matrix'
cal.NCR(data, ...)

## Default S3 method:
cal.NCR(data, ...)
```

Arguments

data	data.frame or matrix. The nematode abundance table where rows represent samples and columns represent nematode genera. Each element indicates the count of a specific nematode genus in the corresponding sample. Row names must be sample names, and column names must be nematode genus names.
...	Additional arguments (currently unused).

Value

A data frame with two columns:

Sample.ID	Character vector of sample identifiers (from row names of data)
NCR	Nematode Channel Ratio for each sample

Examples

```
# Example with a data frame
df <- data.frame(
  Cephalobus = c(10, NA, 15),
  Caenorhabditis = c(5, 10, NA),
  Pratylenchus = c(8, 12, 10),
  row.names = c("A", "B", "C")
)
cal.NCR(data = df)
```

---

cal.PPI *Calculate Plant Parasite Index (PPI)*

Description

This function calculates the Plant Parasite Index (PPI) for ecological communities.

Usage

```
cal.PPI(data, ...)

## S3 method for class 'data.frame'
cal.PPI(data, ...)

## S3 method for class 'matrix'
cal.PPI(data, ...)

## Default S3 method:
cal.PPI(data, ...)
```

Arguments

data	data.frame or matrix. The nematode abundance table where rows represent samples and columns represent nematode genera. Each element indicates the count of a specific nematode genus in the corresponding sample. Row names must be sample names, and column names must be nematode genus names.
...	Additional arguments (currently unused).

Value

A data frame with two columns:

Sample.ID	Character vector of sample identifiers (from row names of data)
PPI	Plant Parasite Index for each sample

Examples

```
# Example with a data frame
df <- data.frame(
  Cephalobus = c(10, NA, 15),
  Caenorhabditis = c(5, 10, NA),
  Pratylenchus = c(8, 12, 10),
  row.names = c("A", "B", "C")
)
cal.PPI(data = df)
```

---

cal.SI *Calculate Structure Index (SI)*

Description

This function calculates the Structure Index (SI) for ecological communities.

Usage

```
cal.SI(data, ...)

## S3 method for class 'data.frame'
cal.SI(data, ...)

## S3 method for class 'matrix'
cal.SI(data, ...)

## Default S3 method:
cal.SI(data, ...)
```

Arguments

data	data.frame or matrix. The nematode abundance table where rows represent samples and columns represent nematode genera. Each element indicates the count of a specific nematode genus in the corresponding sample. Row names must be sample names, and column names must be nematode genus names.
...	Additional arguments (currently unused).

Value

A data frame with two columns:

Sample.ID	Character vector of sample identifiers (from row names of data)
SI	Structure Index for each sample

Examples

```
# Example with a data frame
df <- data.frame(
  Cephalobus = c(10, NA, 15),
  Caenorhabditis = c(5, 10, NA),
  Pratylenchus = c(8, 12, 10),
  row.names = c("A", "B", "C")
)
cal.SI(data = df)
```

---

cal.Simpson *Calculate Simpson Index*

Description

This function calculates the Simpson Index for ecological communities.

Usage

```
cal.Simpson(data, ...)

## S3 method for class 'data.frame'
cal.Simpson(data, ...)

## S3 method for class 'matrix'
cal.Simpson(data, ...)

## Default S3 method:
cal.Simpson(data, ...)
```

Arguments

data	data.frame or matrix. The nematode abundance table where rows represent samples and columns represent nematode genera. Each element indicates the count of a specific nematode genus in the corresponding sample. Row names must be sample names, and column names must be nematode genus names.
...	Additional arguments (currently unused).

Value

A data frame with two columns:

Sample.ID	Character vector of sample identifiers (from row names of data)
Simpson	Simpson's Index for each sample

Examples

```
# Example with a data frame
df <- data.frame(
  Cephalobus = c(10, NA, 15),
  Caenorhabditis = c(5, 10, NA),
  Pratylenchus = c(8, 12, 10),
  row.names = c("A", "B", "C")
)
cal.Simpson(data = df)
```

---

cal.SRI *Calculate Species Richness Index (SRI)*

Description

This function calculates the Species Richness Index (SRI) for ecological communities.

Usage

```
cal.SRI(data, method = "Margalef", ...)

## S3 method for class 'data.frame'
cal.SRI(data, method = "Margalef", ...)

## S3 method for class 'matrix'
cal.SRI(data, method = "Margalef", ...)

## Default S3 method:
cal.SRI(data, method = "Margalef", ...)
```

Arguments

data	data.frame or matrix. The nematode abundance table where rows represent samples and columns represent nematode genera. Each element indicates the count of a specific nematode genus in the corresponding sample. Row names must be sample names, and column names must be nematode genus names.
method	The method used to calculate the Species Richness Index. Default is "Margalef". Supported methods are "Margalef" and "Menhinick". Only one method can be specified. • "Margalef": Margalef's Richness Index, calculated as $(S-1)/\ln(N)$. • "Menhinick": Menhinick's Richness Index, calculated as $S/\sqrt{N}$.
...	Additional arguments (currently unused).

Value

A data frame with two columns:

Sample.ID	Character vector of sample identifiers (from row names of data)
SRI	Species Richness Index for each sample

Examples

```
# Example with a data frame
df <- data.frame(
  Cephalobus = c(10, NA, 15),
  Caenorhabditis = c(5, 10, NA),
  Pratylenchus = c(8, 12, 10),
  row.names = c("A", "B", "C")
)
cal.SRI(data = df, method = "Margalef")
```

---

cal.TD *Calculate Trophic Diversity (TD) Index*

Description

This function calculates the Trophic Diversity (TD) Index for ecological communities.

Usage

```
cal.TD(data, ...)

## S3 method for class 'data.frame'
cal.TD(data, ...)

## S3 method for class 'matrix'
cal.TD(data, ...)

## Default S3 method:
cal.TD(data, ...)
```

Arguments

data	data.frame or matrix. The nematode abundance table where rows represent samples and columns represent nematode genera. Each element indicates the count of a specific nematode genus in the corresponding sample. Row names must be sample names, and column names must be nematode genus names.
...	Additional arguments (currently unused).

Value

A data frame with two columns:

Sample.ID	Character vector of sample identifiers (from row names of data)
TD	Trophic Diversity index for each sample

Examples

```
# Example with a data frame
df <- data.frame(
  Cephalobus = c(10, NA, 15),
  Caenorhabditis = c(5, 10, NA),
  Pratylenchus = c(8, 12, 10),
  row.names = c("A", "B", "C")
)
cal.TD(data = df)
```

---

cal.WI *Calculate Wasilewska Index (WI)*

Description

This function calculates the Wasilewska Index (WI) for ecological communities.

Usage

```
cal.WI(data, ...)

## S3 method for class 'data.frame'
cal.WI(data, ...)

## S3 method for class 'matrix'
cal.WI(data, ...)

## Default S3 method:
cal.WI(data, ...)
```

Arguments

data	data.frame or matrix. The nematode abundance table where rows represent samples and columns represent nematode genera. Each element indicates the count of a specific nematode genus in the corresponding sample. Row names must be sample names, and column names must be nematode genus names.
...	Additional arguments (currently unused).

Value

A data frame with two columns:

Sample.ID	Character vector of sample identifiers (from row names of data)
WI	Wasilewska Index for each sample

Examples

```
# Example with a data frame
df <- data.frame(
  Cephalobus = c(10, NA, 15),
  Caenorhabditis = c(5, 10, NA),
  Pratylenchus = c(8, 12, 10),
  row.names = c("A", "B", "C")
)
cal.WI(data = df)
```

---

check_nematode_genus *Check validity of nematode genus names against reference database*

Description

This generic function validates nematode genus names by checking their existence in a reference database (nematode.info). It supports multiple input types and provides flexible output formats.

Usage

```
check_nematode_genus(Query.genus, Query.col = NULL, show.details = TRUE, ...)

## S3 method for class 'character'
check_nematode_genus(Query.genus, Query.col = NULL, show.details = TRUE, ...)

## S3 method for class 'data.frame'
check_nematode_genus(Query.genus, Query.col, show.details = TRUE, ...)

## Default S3 method:
check_nematode_genus(Query.genus, Query.col = NULL, show.details = TRUE, ...)
```

Arguments

Query.genus	Input to check: can be character vector or data.frame
Query.col	When input is data.frame, specifies column name containing genus names (ignored for character input)
show.details	Logical controlling output format: • TRUE: returns data.frame with query, existence status, and full reference info • FALSE: returns only invalid/missing genus names
...	Additional arguments (currently unused).

Value

Output varies by input type and show.details:

- For character vector input:
  - show.details = TRUE: data.frame with query, existence, and reference data
  - show.details = FALSE: character vector of invalid genera
- For data.frame input: same as character input for the specified column
- For unsupported types: error message

Examples

```
# Check character vector
check_nematode_genus(c("Caenorhabditis", "Wrong"))

# Check data.frame column
df <- data.frame(genus = c("Meloidogyne", "XXX"))
check_nematode_genus(Query.genus = df, Query.col = "genus")
```

---

cp_rel_abundance *Calculate CP Relative or Absolute Abundance*

Description

This function calculates the relative or absolute abundance of nematodes in different CP (Colonizer-Persister) groups for each sample. The CP groups range from CP1 (colonizers, r-strategists) to CP5 (persisters, K-strategists). Genera without CP classification are grouped as No_CP.

Usage

```
cp_rel_abundance(data, total.abundance = NULL, relative = TRUE, ...)

## S3 method for class 'data.frame'
cp_rel_abundance(data, total.abundance = NULL, relative = TRUE, ...)
```

```
## S3 method for class 'matrix'
cp_rel_abundance(data, total.abundance = NULL, relative = TRUE, ...)

## Default S3 method:
cp_rel_abundance(data, total.abundance = NULL, relative = TRUE, ...)
```

Arguments

data	data.frame or matrix. The nematode abundance table where rows represent samples and columns represent nematode genera. Each element indicates the count of a specific nematode genus in the corresponding sample. Row names must be sample names, and column names must be nematode genus names.
total.abundance	data.frame. A data frame with sample names as row names and a single column containing the total nematode abundance for each sample. This parameter is required when relative is set to FALSE. Default is NULL.
relative	Logical. If TRUE (default), the function calculates relative abundance (does not require total.abundance). If FALSE, the function calculates absolute abundance (requires total.abundance).
...	Additional arguments (currently unused).

Value

A data frame with seven columns:

Sample.ID	Character vector of sample identifiers (from row names of data)
CP1	Relative or absolute abundance of CP1 group (colonizers, r-strategists)
CP2	Relative or absolute abundance of CP2 group
CP3	Relative or absolute abundance of CP3 group
CP4	Relative or absolute abundance of CP4 group
CP5	Relative or absolute abundance of CP5 group (persisters, K-strategists)
No_CP	Relative or absolute abundance of genera without CP classification (only present if such genera exist)

Examples

```
# Example with a data frame
df <- data.frame(
  Cephalobus = c(10, NA, 15),
  Caenorhabditis = c(5, 10, NA),
  Pratylenchus = c(8, 12, 10),
  row.names = c("A", "B", "C")
)
abundance <- data.frame(
  abundance = c(100, 150, 120),
  row.names = c("A", "B", "C")
)
cp_rel_abundance(df, abundance, relative = FALSE)
```

```
# Example with a matrix
mat <- matrix(c(10, NA, 15, 5, 10, NA, 8, 12, 10), nrow = 3, byrow = TRUE)
colnames(mat) <- c("Cephalobus", "Caenorhabditis", "Pratylenchus")
row.names(mat) <- c("A", "B", "C")
cp_rel_abundance(mat)
```

`diet_rel_abundance` *Calculate Diet Relative or Absolute Abundance*

Description

This function calculates the relative or absolute abundance of four feeding types of nematodes in each sample. The feeding types include bacterial feeders (Ba), fungus feeders (Fu), plant feeders (Pp), and omnivores/predators (Op).

Usage

```
diet_rel_abundance(data, total.abundance = NULL, relative = TRUE, ...)

## S3 method for class 'data.frame'
diet_rel_abundance(data, total.abundance = NULL, relative = TRUE, ...)

## S3 method for class 'matrix'
diet_rel_abundance(data, total.abundance = NULL, relative = TRUE, ...)

## Default S3 method:
diet_rel_abundance(data, total.abundance = NULL, relative = TRUE, ...)
```

Arguments

`data`	`data.frame` or `matrix`. The nematode abundance table where rows represent samples and columns represent nematode genera. Each element indicates the count of a specific nematode genus in the corresponding sample. Row names must be sample names, and column names must be nematode genus names.
`total.abundance`	`data.frame`. A data frame with sample names as row names and a single column containing the total nematode abundance for each sample. This parameter is required when `relative` is set to `FALSE`. Default is `NULL`.
`relative`	`Logical`. If `TRUE` (default), the function calculates relative abundance (does not require `total.abundance`). If `FALSE`, the function calculates absolute abundance (requires `total.abundance`).
`...`	Additional arguments (currently unused).

Value

A data frame with five columns:

Sample.ID	Character vector of sample identifiers (from row names of data)
Ba	Relative or absolute abundance of bacterial feeders
Fu	Relative or absolute abundance of fungus feeders
Pp	Relative or absolute abundance of plant feeders
Op	Relative or absolute abundance of omnivores/predators

Examples

```
# Example with a data frame
df <- data.frame(
  Cephalobus = c(10, NA, 15),
  Caenorhabditis = c(5, 10, NA),
  Pratylenchus = c(8, 12, 10),
  row.names = c("A", "B", "C")
)
abundance <- data.frame(
  abundance = c(100, 150, 120),
  row.names = c("A", "B", "C")
)
diet_rel_abundance(df, abundance, relative = FALSE)

# Example with a matrix
mat <- matrix(c(10, NA, 15, 5, 10, NA, 8, 12, 10), nrow = 3, byrow = TRUE)
colnames(mat) <- c("Cephalobus", "Caenorhabditis", "Pratylenchus")
row.names(mat) <- c("A", "B", "C")
diet_rel_abundance(mat)
```

Ecological.Indices *Calculate Ecological Indices of Nematodes*

Description

This function calculates various ecological indices based on the provided nematode genus abundance data. It supports a range of indices, including taxonomic diversity, Shannon diversity index, Pielou's evenness index, Simpson's index, and more. Users can specify which indices to calculate or use the default option to calculate all supported indices.

Usage

```
Ecological.Indices(data, indices = "All", method = NULL, ...)

## S3 method for class 'data.frame'
Ecological.Indices(data, indices = "All", method = NULL, ...)
```

```
## S3 method for class 'matrix'
Ecological.Indices(data, indices = "All", method = NULL, ...)

## Default S3 method:
Ecological.Indices(data, indices = "All", method = NULL, ...)
```

Arguments

`data`	`data.frame` or `matrix`. The nematode abundance table where rows represent samples and columns represent nematode genera. Each element indicates the count of a specific nematode genus in the corresponding sample. Row names must be sample names, and column names must be nematode genus names.
`indices`	A character vector specifying the ecological indices to be calculated. The following indices are supported: • "TD" - Trophic Diversity • "H" - Shannon-Wiener Index • "J" - Pielou's Evenness Index • "Simpson" - Simpson Index • "WI" - Wasilewska Index • "MI" - Maturity Index • "PPI" - Plant Parasite Index • "SRI" - Species Richness Index • "NCR" - Nematode Channel Ratio • "CI" - Channel Index • "BI" - Basic Index • "EI" - Enrichment Index • "SI" - Structure Index Additionally, specifying `All` will calculate all supported indices. `All` is the default value.
`method`	The method to use for calculating the Species Richness Index. Default is `NULL`, which uses the default method `Margalef`. Options include: • `"Margalef"`: Margalef's Richness Index, calculated as $(S-1)/\ln(N)$. • `"Menhinick"`: Menhinick's Richness Index, calculated as $S/\sqrt{N}$.
`...`	Additional arguments (currently unused).

Value

A data frame containing the calculated indices. The data frame includes a `Sample.ID` column and additional columns for each requested index.

Examples

```
# Example with a data frame
df <- data.frame(
  Cephalobus = c(10, NA, 15),
  Caenorhabditis = c(5, 10, NA),
```

```
  Pratylenchus = c(8, 12, 10),
  row.names = c("A", "B", "C")
)
Ecological.Indices(data = df, indices = "All", method = "Menhinick")
```

`fuzzy_genus_match` *Fuzzy Matching of Nematode Genus Names*

Description

This function performs fuzzy matching of nematode genus names against a reference database using Levenshtein distance (edit distance) with case insensitivity.

Usage

```
fuzzy_genus_match(Query.genus, max_dist = 2, ...)
```

Arguments

`Query.genus`	A `character vector` of genus names to be matched against the reference
`max_dist`	Maximum allowed Levenshtein distance for matches (default = 2)
`...`	Additional parameters (currently unused)

Value

A data frame containing:

- Query.genus - Original query genus name
- CorrectName - Matched genus name from reference
- Distance - Edit distance between query and match
- Additional columns - All columns from nematode.info for matched records

Examples

```
fuzzy_genus_match(c("Harterta", "Meloidogyne"))
```

NEF *Nematode Energy Footprints (NEF) Calculation*

Description

Nematode Energy Footprints (NEF) Calculation

Usage

```
NEF(data, abundance, AE = list(Ba = 0.6, Fu = 0.38, Pp = 0.25, Op = 0.5), ...)

## S3 method for class 'data.frame'
NEF(data, abundance, AE = list(Ba = 0.6, Fu = 0.38, Pp = 0.25, Op = 0.5), ...)

## S3 method for class 'matrix'
NEF(data, abundance, AE = list(Ba = 0.6, Fu = 0.38, Pp = 0.25, Op = 0.5), ...)

## Default S3 method:
NEF(data, abundance, AE = list(Ba = 0.6, Fu = 0.38, Pp = 0.25, Op = 0.5), ...)
```

Arguments

data	A data.frame or matrix containing nematode genus abundance data. Rows represent samples, and columns represent genera.
abundance	A data.frame containing abundance information for the samples. It must match the row names of the input data.
AE	A named list specifying the assimilation efficiencies for nematode feeding groups. Must contain the following elements: • Ba - Assimilation efficiency for bacterial feeders (default: 0.6) • Fu - Assimilation efficiency for fungal feeders (default: 0.38) • Pp - Assimilation efficiency for plant feeders (default: 0.25) • Op - Assimilation efficiency for omnivores/predators (default: 0.5)
...	Additional arguments (currently unused).

Value

A list object of class "NEF" containing the following components:

data A list with original input data:

- data - Original genus abundance data.frame or matrix of nematode genera
- Abundance - Total abundance data used for calculations

Energy.flux A list containing energy flow calculations:

- Energy.flux: Data frame of energy flows ($\mu g\ C\ 100 g^{-1}\ dry\ soil$) per feeding group. Columns:

  - Sample.ID - Sample identifier
  - BaEF - Bacterial feeders energy flows
  - FuEF - Fungal feeders energy flows
  - PpEF - Plant feeders energy flows
  - OpEF - Omnivores/Predators energy flows
  - TNEF - Total energy flows of nematodes
- C.flux.node: Data frame of Biomass ($\mu g\ C\ 100 g^{-1}\ dry\ soil$) per feeding group. Columns:
  - Sample.ID - Sample identifier
  - Ba - Bacterial feeders biomass
  - Fu - Fungal feeders biomass
  - Pp - Plant feeders biomass
  - Op - Omnivores/Predators biomass
- C.flux.path: Data frame of energy flows ($\mu g\ C\ 100 g^{-1}\ dry\ soil\ day^{-1}$). Columns:
  - Sample.ID - Sample identifier
  - R.to.Ba - Carbon flux from Resources to bacterial feeders
  - R.to.Fu - Carbon flux from Resources to fungal feeders
  - R.to.Pp - Carbon flux from Resources to plant feeders
  - Ba.to.Op - Carbon flux from bacterial to omnivorous channels
  - Fu.to.Op - Carbon flux from fungal to omnivorous channels
  - Pp.to.Op - Carbon flux from plant to omnivorous channels
- U: Data frame of ecosystem stability indices. Columns:
  - Sample.ID - Sample identifier
  - U - Energy flow uniformity index

Examples

```
data <- data.frame(
  Cephalobus = c(10, 20, 30),
  Eucephalobus = c(5, 10, 12),
  Acrobeloides = c(1, 2, 3),
  Caenorhabditis = c(5, 8, 15),
  Aphelenchus = c(5, 13, 11),
  Leptonchus = c(3, 10, 15),
  Pratylenchus = c(9, 2, 15),
  Tylenchus = c(5, 0, 15),
  Mesodorylaimus = c(7, 10, 18),
  Discolaimus = c(1, 10, 25),
  row.names = c("Sample1", "Sample2", "Sample3")
)
abundance <- data.frame(
  Abundance = c(100, 200, 300),
  row.names = c("Sample1", "Sample2", "Sample3")
)
result <- NEF(data, abundance)
print(result)
```

`nematode.ave.mass` *Nematode Genus and Family Average Body Mass*

Description

A dataset containing the average dry body mass (in micrograms) of nematode genera and families, compiled from morphological measurements and allometric scaling. Essential for metabolic rate calculations and size-spectrum analyses in soil ecology.

Usage

```
nematode.ave.mass
```

Format

A data frame with 1094 rows and 4 variables:

Genus Nematode genus name (character), taxonomically validated against Nemaplex database

Family Corresponding taxonomic family (character)

Genus.Average.Mass Mean dry mass per genus (numeric, μg)

Family.Average.Mass Mean dry mass per family (numeric, μg)

Source

Nemaplex.UCDavis.edu; Revision Date: 02/02/2026; Accessed 03/18/2026

- Website: http://nemaplex.ucdavis.edu/

See Also

Use `nematode.info` for complementary trait data.

Examples

```
# Load data
data(nematode.ave.mass)

# Find mass range within a family (e.g. Rhabditidae)
rhabditidae <- subset(nematode.ave.mass, Family == "Rhabditidae")
range(rhabditidae$Genus.Average.Mass, na.rm = TRUE)

# Convert to biomass (example: 100 individuals of Acanthopharynx)
100 * subset(nematode.ave.mass, Genus == "Acanthopharynx")$Genus.Average.Mass
```

`nematode.info` *Nematode Taxonomic and Functional Traits*

Description

A dataset containing taxonomic classification and functional traits of nematode genera, including feeding habits and ecological group (c-p value).

Usage

```
nematode.info
```

Format

A data frame with 2524 rows and 4 variables:

Genus Nematode genus name (character), e.g. "Parascaris", "Heterakis"

Family Taxonomic family name (character), e.g. "Ascarididae", "Heterakidae"

Feeding_habit Feeding behavior category (character), e.g. "Bacterial feeders", "Omnivores"

CP_group Colonizer-Persister group (numeric, 1-5)

Details

This dataset is particularly useful for:

- Ecological studies of soil nematode communities
- Trophic network analysis
- Calculating nematode maturity indices (e.g. MI, PPI)

Source

Nemaplex.UCDavis.edu; Revision Date: 03/18/2026; Accessed 03/18/2026

- Website: http://nemaplex.ucdavis.edu/

Examples

```
# Load the data
data(nematode.info)

# Count nematodes by feeding habit
table(nematode.info$Feeding_habit)

# Find all genera in Ascarididae family
subset(nematode.info, Family == "Ascarididae")
```

NMF *Nematode Metabolic Footprints (NMF) Calculation*

Description

This function calculates various Nematode Metabolic Footprints (NMF) based on the input data and abundance information. It supports multiple types of NMF calculations and can handle data in different formats (data.frame or matrix).

Usage

```
NMF(data, abundance, type = "All", ...)

## S3 method for class 'data.frame'
NMF(data, abundance, type = "All", ...)

## S3 method for class 'matrix'
NMF(data, abundance, type = "All", ...)

## Default S3 method:
NMF(data, abundance, type = "all", ...)
```

Arguments

data	`data.frame` or `matrix`. The nematode abundance table where rows represent samples and columns represent nematode genera. Each element indicates the count of a specific nematode genus in the corresponding sample. Row names must be sample names, and column names must be nematode genus names.
abundance	`data.frame`. A data frame with sample names as row names and a single column containing the total nematode abundance for each sample.
type	Character vector specifying the type(s) of NMF to calculate. `All` is the default value. Valid options include: • "BaMF" - Bacterial Feeders Nematode Metabolic Footprints • "FuMF" - Fungal Feeders Nematode Metabolic Footprints • "PpMF" - Plant Feeders Nematode Metabolic Footprints • "OpMF" - Omnivore/Predator Nematode Metabolic Footprints • "Fe" - Enrichment Footprints (CP group <= 2) • "Fs" - Structure Footprints (CP group > 2) • "TNMF" - Total Nematode Metabolic Footprints • "FMF" - Functional Metabolic Footprints (product of Fe and Fs) • "All" - Calculate all the above types
...	Additional arguments (currently unused).

Value

A data.frame containing the calculated NMF values for each sample. The columns represent different NMF types, and the rows correspond to samples.

Examples

```
data <- data.frame(
  Cephalobus = c(10, 20, 30),
  Eucephalobus = c(5, 10, 12),
  Acrobeloides = c(1, 2, 3),
  Caenorhabditis = c(5, 8, 15),
  Aphelenchus = c(5, 13, 11),
  Leptonchus = c(3, 10, 15),
  Pratylenchus = c(9, 2, 15),
  Tylenchus = c(5, 0, 15),
  Mesodorylaimus = c(7, 10, 18),
  Discolaimus = c(1, 10, 25),
  row.names = c("Sample1", "Sample2", "Sample3")
)
abundance <- data.frame(
  Abundance = c(100, 200, 300),
  row.names = c("Sample1", "Sample2", "Sample3")
)
result <- NMF(data, abundance, type = "All")
print(result)
```

---

num_species *Calculate Number of Species*

Description

This function calculates the number of nematode species present in each sample. It counts the number of non-zero and non-empty nematode species for each sample.

Usage

```
num_species(data, ...)

## S3 method for class 'data.frame'
num_species(data, ...)

## S3 method for class 'matrix'
num_species(data, ...)

## Default S3 method:
num_species(data, ...)
```

Arguments

data	data.frame or matrix. The nematode abundance table where rows represent samples and columns represent nematodes. Each element indicates the count of a specific nematode in the corresponding sample. Row names must be sample names.
...	Additional arguments (currently unused).

Value

A data.frame with two columns:

Sample.ID	Character vector of sample identifiers (from row names)
NumSpecies	Number of non-zero nematode species in each sample

Examples

```
# Example with a data frame
df <- data.frame(
  Species1 = c(10, NA, 15),
  Species2 = c(5, 10, NA),
  Species3 = c(8, 12, 10),
  row.names = c("A", "B", "C")
)
num_species(df)

# Example with a matrix
mat <- matrix(c(10, NA, 15, 5, 10, NA, 8, 12, 10), nrow = 3, byrow = TRUE)
colnames(mat) <- c("Species1", "Species2", "Species3")
row.names(mat) <- c("A", "B", "C")
num_species(mat)
```

---

rel_abundance *Calculate the Relative Abundance of Nematodes*

Description

This function calculates the relative abundance of nematodes for each sample. The relative abundance is defined as the proportion of each nematode's count to the total count of all nematodes in a sample.

Usage

```
rel_abundance(data, ...)

## S3 method for class 'data.frame'
rel_abundance(data, ...)
```

```
## S3 method for class 'matrix'
rel_abundance(data, ...)

## Default S3 method:
rel_abundance(data, ...)
```

Arguments

data	data.frame or matrix. The nematode abundance table where rows represent samples and columns represent nematodes. Each element indicates the count of a specific nematode in the corresponding sample.
...	Additional arguments (currently unused).

Value

A data.frame or matrix (matching the input type) containing the relative abundance of each nematode in each sample.

Examples

```
# Example with a data frame
df <- data.frame(
  Species1 = c(10, NA, 15),
  Species2 = c(5, 10, NA),
  Species3 = c(8, 12, 10),
  row.names = c("A", "B", "C")
)
rel_abundance(df)

# Example with a matrix
mat <- matrix(c(10, NA, 15, 5, 10, NA, 8, 12, 10), nrow = 3, byrow = TRUE)
colnames(mat) <- c("Species1", "Species2", "Species3")
row.names(mat) <- c("A", "B", "C")
rel_abundance(mat)
```

---

runNMDS *Non-Metric Multidimensional Scaling (NMDS) Analysis*

Description

This function performs NMDS analysis on a dataset using the specified distance metric, and optionally runs PERMANOVA (adonis2) and ANOSIM tests for group differences. It supports both data.frame and matrix inputs.

Usage

```
runNMDS(
  data,
  group,
  distance = "bray",
  k = 2,
  decostand.method = "hellinger",
  autotransform = TRUE,
  adonis2 = TRUE,
  anosim = TRUE,
  simper = TRUE,
  ...
)

## S3 method for class 'data.frame'
runNMDS(
  data,
  group,
  distance = "bray",
  k = 2,
  decostand.method = "hellinger",
  autotransform = TRUE,
  adonis2 = TRUE,
  anosim = TRUE,
  simper = TRUE,
  ...
)

## S3 method for class 'matrix'
runNMDS(
  data,
  group,
  distance = "bray",
  k = 2,
  decostand.method = "hellinger",
  autotransform = TRUE,
  adonis2 = TRUE,
  anosim = TRUE,
  simper = TRUE,
  ...
)

## Default S3 method:
runNMDS(
  data,
  group,
  distance = "bray",
  k = 2,
```

```
  decostand.method = "hellinger",
  autotransform = TRUE,
  adonis2 = TRUE,
  anosim = TRUE,
  simper = TRUE,
  ...
)
```

Arguments

data	data.frame or matrix. The nematode abundance table where rows represent samples and columns represent nematode genera. Each element indicates the count of a specific nematode genus in the corresponding sample. Row names must be sample names.
group	data.frame. A data frame with sample names as row names and a single column containing group information for each sample.
distance	Distance metric to use (default: "bray"). See metaMDS for all available options.
k	Number of dimensions for NMDS (default: 2).
decostand.method	Standardization methods for community ecology data (default: "hellinger"). Set to NULL for no transformation. See decostand for all available options.
autotransform	Logical; whether to automatically transform the data (default: TRUE). See metaMDS for details.
adonis2	Logical; whether to perform PERMANOVA test using adonis2 (default: TRUE).
anosim	Logical; whether to perform ANOSIM test using anosim (default: TRUE).
simper	Logical; whether to perform SIMPER test using simper (default: TRUE).
...	Additional arguments passed to metaMDS, decostand, adonis2, anosim, or simper.

Value

An object of class "NMDS" containing:

- data - List containing the input data and group information
- call - The function call
- NMDS - NMDS results from metaMDS
- adonis2 - PERMANOVA results (if adonis2 = TRUE)
- anosim - ANOSIM results (if anosim = TRUE)
- SIMPER - SIMPER results (if simper = TRUE)

See Also

- metaMDS for details on NMDS implementation and distance measures
- decostand for details on standardization methods
- vegdist for available distance metrics

- adonis2 for PERMANOVA
- anosim for ANOSIM
- simper for SIMPER

Examples

```
# Example with default Bray-Curtis distance
data <- data.frame(
  Cephalobus = c(10, 20, 30, 1, 6, 5),
  Eucephalobus = c(5, 10, 12, 30, 1, 6),
  Acrobeloides = c(1, 2, 3, 12, 30, 1),
  Caenorhabditis = c(5, 8, 15, 2, 3, 12),
  Aphelenchus = c(5, 13, 11, 15, 2, 3),
  Leptonchus = c(3, 10, 15, 0, 15, 11),
  Pratylenchus = c(9, 2, 15, 15, 0, 15),
  Tylenchus = c(5, 0, 15, 11, 15, 2),
  Mesodorylaimus = c(7, 10, 18, 3, 12, 30),
  Discolaimus = c(1, 10, 25, 10, 18, 3),
  row.names = c("Sample1", "Sample2", "Sample3", "Sample4", "Sample5", "Sample6")
)
group_df <- data.frame(
  group = c("A", "A", "B", "B", "C", "C"),
  row.names = c("Sample1", "Sample2", "Sample3", "Sample4", "Sample5", "Sample6")
)

nmds <- runNMDS(data, group = group_df)
```

---

runPCoA *Principal Coordinates Analysis (PCoA) Analysis*

Description

This function performs PCoA analysis on a dataset using the specified distance metric, and optionally runs PERMANOVA (adonis2) and ANOSIM tests for group differences. It supports both data.frame and matrix inputs.

Usage

```
runPCoA(
  data,
  group,
  k = 2,
  distance = "bray",
  adonis2 = TRUE,
  anosim = TRUE,
  simper = TRUE,
  ...
```

```
)

## S3 method for class 'data.frame'
runPCoA(
  data,
  group,
  k = 2,
  distance = "bray",
  adonis2 = TRUE,
  anosim = TRUE,
  simper = TRUE,
  ...
)

## S3 method for class 'matrix'
runPCoA(
  data,
  group,
  k = 2,
  distance = "bray",
  adonis2 = TRUE,
  anosim = TRUE,
  simper = TRUE,
  ...
)

## Default S3 method:
runPCoA(
  data,
  group,
  k = 2,
  distance = "bray",
  adonis2 = TRUE,
  anosim = TRUE,
  simper = TRUE,
  ...
)
```

Arguments

`data`	`data.frame` or `matrix`. The nematode abundance table where rows represent samples and columns represent nematode genera. Each element indicates the count of a specific nematode genus in the corresponding sample. Row names must be sample names.
`group`	`data.frame`. A data frame with sample names as row names and a single column containing group information for each sample.
`k`	Number of dimensions for PCoA (default: 2).
`distance`	Distance metric to use (default: "bray"). See `vegdist` for all available options.

adonis2	Logical; whether to perform PERMANOVA test using adonis2 (default: TRUE).
anosim	Logical; whether to perform ANOSIM test using anosim (default: TRUE).
simper	Logical; whether to perform SIMPER test using simper (default: TRUE).
...	Additional arguments passed to cmdscale, decostand, adonis2, anosim, or simper.

Value

An object of class "PCoA" containing:

- data - List containing the input data and group information
- call - The function call
- Points - Sample coordinates in the reduced space.
- Eigenvalues - Variance explained by each principal coordinate axis.
- adonis2 - PERMANOVA results (if adonis2 = TRUE)
- anosim - ANOSIM results (if anosim = TRUE)
- SIMPER - SIMPER results (if simper = TRUE)

See Also

- cmdscale for details on cmdscale implementation
- vegdist for available distance metrics
- adonis2 for PERMANOVA
- anosim for ANOSIM
- simper for SIMPER

Examples

```
# Example with default Bray-Curtis distance
data <- data.frame(
  Cephalobus = c(10, 20, 30, 1, 6, 5),
  Eucephalobus = c(5, 10, 12, 30, 1, 6),
  Acrobeloides = c(1, 2, 3, 12, 30, 1),
  Caenorhabditis = c(5, 8, 15, 2, 3, 12),
  Aphelenchus = c(5, 13, 11, 15, 2, 3),
  Leptonchus = c(3, 10, 15, 0, 15, 11),
  Pratylenchus = c(9, 2, 15, 15, 0, 15),
  Tylenchus = c(5, 0, 15, 11, 15, 2),
  Mesodorylaimus = c(7, 10, 18, 3, 12, 30),
  Discolaimus = c(1, 10, 25, 10, 18, 3),
  row.names = c("Sample1", "Sample2", "Sample3", "Sample4", "Sample5", "Sample6")
)
group_df <- data.frame(
  group = c("A", "A", "B", "B", "C", "C"),
  row.names = c("Sample1", "Sample2", "Sample3", "Sample4", "Sample5", "Sample6")
)

pcoa <- runPCoA(data, group = group_df)
```

runSimper *Similarity Percentages Analysis*

Description

Discriminating species between two groups using Bray-Curtis dissimilarities

Usage

```
runSimper(object, ...)

## S3 method for class 'Ordination'
runSimper(object, ...)

## Default S3 method:
runSimper(object, ...)
```

Arguments

object	An object of class "Ordination".
...	Additional arguments passed to simper.

Value

The object of class "Ordination" containing (See runNMDS for details):

- data - List containing the input data and group information
- call - The function call
- NMDS - NMDS results from metaMDS
- SIMPER - SIMPER results

Examples

```
# Example with default Bray-Curtis distance
data <- data.frame(
  Cephalobus = c(10, 20, 30, 1, 6, 5),
  Eucephalobus = c(5, 10, 12, 30, 1, 6),
  Acrobeloides = c(1, 2, 3, 12, 30, 1),
  Caenorhabditis = c(5, 8, 15, 2, 3, 12),
  Aphelenchus = c(5, 13, 11, 15, 2, 3),
  Leptonchus = c(3, 10, 15, 0, 15, 11),
  Pratylenchus = c(9, 2, 15, 15, 0, 15),
  Tylenchus = c(5, 0, 15, 11, 15, 2),
  Mesodorylaimus = c(7, 10, 18, 3, 12, 30),
  Discolaimus = c(1, 10, 25, 10, 18, 3),
  row.names = c("Sample1", "Sample2", "Sample3", "Sample4", "Sample5", "Sample6")
)
group_df <- data.frame(
```

```
  group = c("A", "A", "B", "B", "C", "C"),
  row.names = c("Sample1", "Sample2", "Sample3", "Sample4", "Sample5", "Sample6")
)
nmds <- runNMDS(data, group = group_df, simper = FALSE)

# Example
nmds_simper <- runSimper(nmds)
print(nmds_simper$SIMPER)
```

`summary.NMDS` *Summarize NMDS Results*

Description

Provides a concise summary of Non-Metric Multidimensional Scaling (NMDS) analysis results, including stress value, PERMANOVA (adonis2) and ANOSIM test statistics.

Provides a concise summary of Principal Coordinates Analysis (PCoA) analysis results, including PERMANOVA (adonis2) and ANOSIM test statistics.

Usage

```
## S3 method for class 'NMDS'
summary(object, ...)

## S3 method for class 'PCoA'
summary(object, ...)
```

Arguments

`object`	An object of class "PCoA" produced by `runPCoA` function.
`...`	Additional arguments (currently not used).

Value

A list containing:

- stress - NMDS stress value
- points - Sample coordinates in the reduced space
- adonis2 - PERMANOVA results (R2, p-value, significance)
- anosim - ANOSIM results (R statistic, p-value, significance)

A list containing:

- points - Sample coordinates in the reduced space.
- eig - Variance explained by each principal coordinate axis.
- adonis2 - PERMANOVA results (R2, p-value, significance)
- anosim - ANOSIM results (R statistic, p-value, significance)

Examples

```
# Example with default Bray-Curtis distance
data <- data.frame(
  Cephalobus = c(10, 20, 30, 1, 6, 5),
  Eucephalobus = c(5, 10, 12, 30, 1, 6),
  Acrobeloides = c(1, 2, 3, 12, 30, 1),
  Caenorhabditis = c(5, 8, 15, 2, 3, 12),
  Aphelenchus = c(5, 13, 11, 15, 2, 3),
  Leptonchus = c(3, 10, 15, 0, 15, 11),
  Pratylenchus = c(9, 2, 15, 15, 0, 15),
  Tylenchus = c(5, 0, 15, 11, 15, 2),
  Mesodorylaimus = c(7, 10, 18, 3, 12, 30),
  Discolaimus = c(1, 10, 25, 10, 18, 3),
  row.names = c("Sample1", "Sample2", "Sample3", "Sample4", "Sample5", "Sample6")
)
group_df <- data.frame(
  group = c("A", "A", "B", "B", "C", "C"),
  row.names = c("Sample1", "Sample2", "Sample3", "Sample4", "Sample5", "Sample6")
)

# Example for summary.NMDS
nmds <- runNMDS(data, group = group_df)
summary(nmds)

# Example for summary.PCoA
pcoa <- runPCoA(data, group = group_df)
summary(pcoa)
```

Index

∗ **datasets**
    nematode.ave.mass, 24
    nematode.info, 25

adonis2, *31, 32*, *34*
anosim, *31, 32*, *34*

cal.BI, 2
cal.CI, 3
cal.EI, 4
cal.H, 5
cal.J, 6
cal.MI, 7
cal.NCR, 8
cal.PPI, 9
cal.SI, 10
cal.Simpson, 11
cal.SRI, 12
cal.TD, 13
cal.WI, 14
check_nematode_genus, 15
cmdscale, *34*
cp_rel_abundance, 16

decostand, *31*, *34*
diet_rel_abundance, 18

Ecological.Indices, 19

fuzzy_genus_match, 21

metaMDS, *31*, *35*

NEF, 22
nematode.ave.mass, 24
nematode.info, *24*, 25
NMF, 26
num_species, 27

rel_abundance, 28
runNMDS, 29, *35*
runPCoA, 32, *36*
runSimper, 35

simper, *31, 32*, *34, 35*
summary.NMDS, 36
summary.PCoA *(*summary.NMDS*)*, 36

vegdist, *31*, *33, 34*