Package ‘pfwim’

May 9, 2026

Title Food Web Construction for Paleo Communities

Version 1.0.2

Description R tools for reconstructing paleo food webs from species traits and size rules.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Depends R (>= 3.5)

LazyData true

Imports dplyr, stats, tibble, tidyr, tidyselect

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

Config/testthat/edition 3

VignetteBuilder knitr

URL `https://beckslab.github.io/pfwim/`,
`https://github.com/BecksLab/pfwim`

NeedsCompilation no

Author Tanya Strydom [aut, cre] (ORCID:
<`https://orcid.org/0000-0001-6067-1349`>),
Jack O. Shaw [ctb] (ORCID: <`https://orcid.org/0000-0002-2725-2334`>),
Alexander M. Dunhill [ctb] (ORCID:
<`https://orcid.org/0000-0002-8680-9163`>),
Andrew Beckerman [ctb] (ORCID: <`https://orcid.org/0000-0002-7859-8394`>),
Baran Karapunar [ctb] (ORCID: <`https://orcid.org/0000-0001-9711-1492`>),
Jennifer A. Dunne [ctb] (ORCID:
<`https://orcid.org/0000-0002-3789-9561`>),
Pincelli M. Hull [ctb] (ORCID: <`https://orcid.org/0000-0001-8607-4817`>)

Maintainer Tanya Strydom <`t.strydom@sheffield.ac.uk`>

Repository CRAN

Date/Publication 2026-04-28 19:30:24 UTC

Contents

`feeding_rules` *Example feeding rules for* `traits`

Description

A mock trait dataset using four trait classes specified in `traits` to infer feeding interactions

A mock trait dataset using four trait classes specified in `traits` to infer feeding interactions

Usage

```
feeding_rules
```

```
feeding_rules
```

Format

`feeding_rules`**:**
A data frame with 7,240 rows and 60 columns:

trait_type_resource broader resource trait class - i.e. column name in traits
trait_resource specific resource trait class - i.e. row entry in trait column
trait_type_consumer broader consumer trait class - i.e. column name in traits
trait_consumer specific consumer trait class - i.e. row entry in trait column ...

`feeding_rules`**:**
A data frame with 7,240 rows and 60 columns:

trait_type_resource broader resource trait class - i.e. column name in traits
trait_resource specific resource trait class - i.e. row entry in trait column
trait_type_consumer broader consumer trait class - i.e. column name in traits
trait_consumer specific consumer trait class - i.e. row entry in trait column ...

Source

NA

NA

`infer_edgelist` *Infer trophic interactions using PFWIM trait rules*

Description

Infers a consumer–resource edgelist using categorical trait matching rules based on the PFWIM (Paleo Food Web Inference Model) described in Shaw (2024). Interactions are inferred by comparing resource and consumer trait combinations against a set of allowed trait rules.

Usage

```
infer_edgelist(
  data,
  cat_combo_list,
  col_taxon = "taxon",
  col_num_size = NULL,
  cat_trait_types = NULL,
  num_size_rule = NULL,
  certainty_req = "all",
  allow_self = TRUE,
  return_full_matrix = FALSE,
  print_dropped_taxa = FALSE,
  hide_printout = FALSE,
  ...
)
```

Arguments

`data`	A `data.frame` containing taxa and associated trait values. Each row represents a taxon and each column represents a trait.
`cat_combo_list`	A `data.frame` defining allowed consumer–resource trait combinations. Must contain columns: **trait_type_resource** Resource trait category **trait_resource** Resource trait value **trait_type_consumer** Consumer trait category **trait_consumer** Consumer trait value
`col_taxon`	Character string indicating the column containing taxon names in `data`. Default `"taxon"`.
`col_num_size`	Optional column name containing numerical size values for taxa. Used when applying a numeric predator–prey size rule.
`cat_trait_types`	Optional character vector specifying a subset of categorical trait columns to use. If `NULL`, all traits appearing in `cat_combo_list` are used.

`num_size_rule`	Optional function defining the predator–prey size feasibility rule. The function must accept two numeric vectors: `(resource_size, consumer_size)` and return `1` for feasible interactions and `0` otherwise. Example: `function(res_size, con_size) { ifelse(res_size <= con_size, 1, 0) }`
`certainty_req`	Defines how many trait rules must be satisfied for an interaction to be considered feasible. **"all"** All trait types must match **numeric** Minimum number of matching trait types required
`allow_self`	Logical. If TRUE, allows interactions where the resource and consumer are the same taxon (self-loops). Default is FALSE.
`return_full_matrix`	Logical. If `TRUE`, returns the full matrix of taxon pairs and the number of matching trait rules. If `FALSE`, returns only inferred interactions as an edgelist.
`print_dropped_taxa`	Logical. If `TRUE`, prints taxa that were removed from the inferred food web because they have no feasible interactions.
`hide_printout`	Logical. If `TRUE`, suppresses progress messages.
`...`	Additional arguments reserved for future extensions.

Details

Trait matching is performed across all trait types present in `cat_combo_list`. For each potential taxon pair, the number of satisfied trait rules is calculated. An interaction is inferred when the number of satisfied rules meets the required threshold defined by `certainty_req`.

Optionally, a numerical predator–prey size rule can also be applied using `num_size_rule`.

The function evaluates all possible consumer–resource taxon pairs and determines interaction feasibility using categorical trait rules and, optionally, a numerical size constraint. The final edgelist includes only interactions meeting the certainty requirement.

Value

If `return_full_matrix = FALSE`:

A two-column matrix containing:

taxon_resource Resource taxon

taxon_consumer Consumer taxon

If `return_full_matrix = TRUE`:

A matrix containing all taxon pairs and the number of trait rules satisfied.

References

Shaw, J. (2024). PFWIM: Paleo Food web Inference Model. *Preprint*.

Examples

```
infer_edgelist(
  data = traits,
  cat_combo_list = feeding_rules,
  col_taxon = "species",
  certainty_req = "all"
)
```

`powerlaw_prey` *Generate hypothetical realised webs using a power-law link distribution*

Description

PFWIM generates a series of replicate hypothetical realised food webs by reducing the feasible links for each consumer to match a target link distribution. The default distribution is a mixed exponential–power law in-degree distribution as described in Shaw (2024) and Roopnarine (2006).

Usage

```
powerlaw_prey(
  el,
  n_samp = 50,
  y = 2.5,
  func = function(r, M, y) exp(-r/(exp((y - 1) * (log(M)/y))))
)
```

Arguments

`el`	Data frame or matrix containing a feasible consumer–resource edgelist. Column 1 = resource, Column 2 = consumer.
`n_samp`	Integer. Number of replicate realised webs to generate. Default = 50.
`y`	Numeric. Parameter controlling the shape of the power-law distribution. Default = 2.5.
`func`	Function. Probability function of the in-degree `r`, total prey richness `M`, and parameter `y`. Must return a numeric value > 0. Default: `function(r, M, y) exp(-r / (exp((y - 1) * (log(M) / y))))`.

Details

For each consumer in `el`, the number of prey links in a realised web is sampled according to the distribution defined by `func`. The sampled prey are drawn randomly without exceeding the maximum feasible prey for that consumer.

Value

A list of length `n_samp`. Each element is a data frame representing a realised food web edgelist with two columns:

resource Resource species

consumer Consumer species

References

Shaw, J. (2024). PFWIM: Paleo Food-web Inference Model. *Preprint*. Roopnarine, P. (2006). *Palaeoecology and food-web structure in fossil communities*.

Examples

```
# Infer a minimal edgelist
edgelist <- infer_edgelist(
  data = data.frame(
    species = c("plankton","plant_1","plant_2","cod","rat","deer"),
    feeding = c("primary","primary","primary","secondary","secondary","secondary")
  ),
  cat_combo_list = data.frame(
    trait_type_resource = c("feeding","feeding","feeding"),
    trait_resource = c("primary","primary","primary"),
    trait_type_consumer = c("feeding","feeding","feeding"),
    trait_consumer = c("secondary","secondary","secondary")
  ),
  col_taxon = "species",
  certainty_req = "all"
)

# Generate realized webs
webs <- powerlaw_prey(edgelist, n_samp = 3, y = 2.5)
```

`traits` *Example species traits data to infer feeding rules*

Description

A mock trait dataset using four trait classes to determine interactions as specified in `feeding_rules`

Usage

```
traits
```

Format

`traits`:

A data frame with 7,240 rows and 60 columns:

species	species name
motility	motility class of species
habitat	habitat species found in
feeding	trophic level of species
size	categorical size classes ...

Source

NA

Index