

Package: metaweave (via r-universe)

October 7, 2026

Title Spatial Ecological Network Inference

Version 0.4.0

Author Gabriel Munoz [aut, cre]

Maintainer Gabriel Munoz <gabriel.munoz@example.com>

Description A model-agnostic framework for reconstructing and analysing spatially explicit ecological networks from species distributions and ecological inference models. Supports arbitrary ecological groups and includes stochastic block and maximum-entropy inference backends, simulation helpers, network summaries, and spatial mapping utilities.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Depends R (>= 4.1.0)

Imports methods, stats, terra

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

VignetteBuilder knitr

Config/testthat/edition 3

URL <https://github.com/fgabriel1891/metaweave>

BugReports <https://github.com/fgabriel1891/metaweave/issues>

Config/pak/sysreqs libgdal-dev gdal-bin libgeos-dev libproj-dev libsqlite3-dev

Repository <https://fgabriel1891.r-universe.dev>

Date/Publication 2026-09-14 15:53:46 UTC

RemoteUrl <https://github.com/fgabriel1891/metaweave>

RemoteRef HEAD

RemoteSha 694b83eba90e47ce7e627330882170aeabf74ef3

Contents

add_network_metrics	3
assemble_communities	3
block_model	4
build_range_stack	4
create_standard_grid	5
crop_filter_stack	5
downscale_networks	6
infer_network	6
infer_networks	7
interaction_table_to_matrix	7
list_shapefiles	8
local_block_network	8
local_probability_network	9
make_bbox	10
map_metrics	10
maxent_model	11
network_metrics_raster	12
new_assemblage	13
new_distribution_collection	13
new_ecological_network	14
new_inference_model	14
new_network_collection	15
new_spatial_result	15
prepare_range_stacks	16
print.metaweave	16
probability_matrix_model	17
rarefied_network_metric	17
rasterize_range_file	18
rasterize_range_rows	18
reconstruct_networks	19
run_network_downscaling	19
run_spatial_inference	20
sbm_model	20
simulate_interaction_counts	21
simulate_network	21
summarize_network	22
summarize_networks	22

Index

add_network_metrics	<i>Add standard metrics to network collection</i>
---------------------	---

Description

Add standard metrics to network collection

Usage

```
add_network_metrics(x, threshold = NULL)
```

Arguments

x	Network collection.
threshold	Optional probability threshold.

assemble_communities	<i>Assemble local communities from aligned distribution rasters</i>
----------------------	---

Description

Assemble local communities from aligned distribution rasters

Usage

```
assemble_communities(distributions, min_species = 1L)
```

Arguments

distributions	Named list of SpatRaster or distribution_collection objects.
min_species	Minimum species per group.

block_model	<i>Construct a generic stochastic block model backend</i>
-------------	---

Description

Construct a generic stochastic block model backend

Usage

```
block_model(
  row_lookup,
  column_lookup,
  theta,
  row_group,
  column_group,
  species_col = "species",
  guild_col = "guild"
)
```

Arguments

row_lookup	Data frame mapping species in the matrix rows to blocks.
column_lookup	Data frame mapping species in the matrix columns to blocks.
theta	Guild-by-guild probability matrix.
row_group	Name of the assemblage group placed in matrix rows.
column_group	Name of the assemblage group placed in matrix columns.
species_col	Species column in lookup tables.
guild_col	Guild column in lookup tables.

build_range_stack	<i>Build a species range stack</i>
-------------------	------------------------------------

Description

Build a species range stack

Usage

```
build_range_stack(
  source,
  template,
  species_col = NULL,
  species_keep = NULL,
  region = NULL,
  touches = TRUE
)
```

Arguments

source	Directory, vector file, sf, or SpatVector.
template	Raster template.
species_col	Species column for multi-species data.
species_keep	Optional species subset.
region	Optional crop region.
touches	Count all touched cells.

create_standard_grid *Create a standard raster grid*

Description

Create a standard raster grid

Usage

```
create_standard_grid(
  extent = c(-180, 180, -90, 90),
  resolution = 1,
  crs = "EPSG:4326"
)
```

Arguments

extent	Numeric vector c(xmin, xmax, ymin, ymax).
resolution	Cell resolution.
crs	Coordinate reference system.

crop_filter_stack *Crop a stack and remove species absent from the region*

Description

Crop a stack and remove species absent from the region

Usage

```
crop_filter_stack(stack, region, return_cropped = TRUE)
```

Arguments

stack	Species raster stack.
region	Crop region.
return_cropped	Return cropped rather than original layers.

downscale_networks	<i>Compatibility wrapper for spatial network downscaling</i>
--------------------	--

Description

Compatibility wrapper for spatial network downscaling

Usage

```
downscale_networks(
  palm_stack,
  mammal_stack,
  palm_lookup,
  mammal_lookup,
  theta,
  species_col = "species",
  guild_col = "guild",
  min_palms = 1L,
  min_mammals = 1L
)
```

Arguments

palm_stack, mammal_stack	Aligned species distribution rasters.
palm_lookup, mammal_lookup	Species-to-guild lookup tables.
theta	Guild-by-guild probability matrix.
species_col	Species column in lookup tables.
guild_col	Guild column in lookup tables.
min_palms, min_mammals	Minimum local richness in each group.

infer_network	<i>Infer a network for one assemblage</i>
---------------	---

Description

Infer a network for one assemblage

Usage

```
infer_network(assemblage, model)
```

Arguments

assemblage	An assemblage.
model	An inference model.

infer_networks	<i>Infer networks for many assemblages</i>
----------------	--

Description

Infer networks for many assemblages

Usage

```
infer_networks(assemblages, model, template = NULL)
```

Arguments

assemblages	List of assemblages.
model	Inference model.
template	Optional spatial template.

interaction_table_to_matrix	<i>Convert an interaction table to a matrix</i>
-----------------------------	---

Description

Convert an interaction table to a matrix

Usage

```
interaction_table_to_matrix(data, row_col, column_col, value_col, fill = 0)
```

Arguments

data	Data frame.
row_col, column_col, value_col	Column names.
fill	Missing-pair value.

list_shapefiles	<i>List ESRI shapefiles</i>
-----------------	-----------------------------

Description

List ESRI shapefiles

Usage

```
list_shapefiles(path, recursive = FALSE)
```

Arguments

path	Directory.
recursive	Search recursively.

local_block_network	<i>Build one local generic block-model network directly</i>
---------------------	---

Description

Build one local generic block-model network directly

Usage

```
local_block_network(  
  rows,  
  columns,  
  row_lookup,  
  column_lookup,  
  theta,  
  row_group = "rows",  
  column_group = "columns",  
  species_col = "species",  
  guild_col = "guild",  
  cell_id = NA_integer_  
)
```

Arguments

rows, columns	Species names in each network group.
row_lookup	Data frame mapping species in the matrix rows to blocks.
column_lookup	Data frame mapping species in the matrix columns to blocks.
theta	Guild-by-guild probability matrix.

row_group	Name of the assemblage group placed in matrix rows.
column_group	Name of the assemblage group placed in matrix columns.
species_col	Species column in lookup tables.
guild_col	Guild column in lookup tables.
cell_id	Optional raster cell identifier.

local_probability_network

Build one local palm-mammal probability network directly

Description

Compatibility wrapper around local_block_network().

Usage

```
local_probability_network(
  palms,
  mammals,
  palm_lookup,
  mammal_lookup,
  theta,
  species_col = "species",
  guild_col = "guild",
  cell_id = NA_integer_
)
```

Arguments

palms, mammals	Species names in each network group.
palm_lookup, mammal_lookup	Species-to-guild lookup tables.
theta	Guild-by-guild probability matrix.
species_col	Species column in lookup tables.
guild_col	Guild column in lookup tables.
cell_id	Optional raster cell identifier.

make_bbox	Create a bounding-box polygon
-----------	-------------------------------

Description

Create a bounding-box polygon

Usage

```
make_bbox(extent, crs = "EPSG:4326")
```

Arguments

- extent Numeric vector c(xmin, xmax, ymin, ymax).
- crs Coordinate reference system.

map_metrics	Map cell metrics back to rasters
-------------	----------------------------------

Description

Map cell metrics back to rasters

Usage

```
map_metrics(x, metrics = NULL)
```

Arguments

- x Network collection with metrics in its index.
- metrics Metric columns; inferred when omitted.

maxent_model	<i>Maximum-entropy ecological network model</i>
--------------	---

Description

Constructs a directed network inference backend that generates binary adjacency matrices under hard structural constraints and returns their marginal species-pair probabilities. Rows are consumers, columns are resources, and `matrix[i, j] = 1` means consumer `i` uses resource `j`.

Usage

```
maxent_model(
  row_group,
  column_group = row_group,
  constraint = c("connectance", "links", "degree_sequence"),
  connectance = NULL,
  links = NULL,
  out_degree = NULL,
  in_degree = NULL,
  ensemble_size = 100L,
  self_links = FALSE,
  seed = 1L,
  burn_in = 1000L,
  thin = 100L,
  keep_ensemble = FALSE
)
```

Arguments

<code>row_group</code>	Assemblage group represented by matrix rows (consumers).
<code>column_group</code>	Assemblage group represented by matrix columns (resources). May equal <code>row_group</code> for a directed unipartite food web.
<code>constraint</code>	One of "connectance", "links", or "degree_sequence".
<code>connectance</code>	Desired connectance in $[0, 1]$. Converted to an exact local link count with <code>round(connectance * allowed_dyads)</code> .
<code>links</code>	Exact number of interactions in every local ensemble member.
<code>out_degree</code>	Row sums (numbers of resources per consumer).
<code>in_degree</code>	Column sums (numbers of consumers per resource). Degree constraints may be named numeric vectors or functions accepting the locally present species names and returning a numeric vector.
<code>ensemble_size</code>	Number of adjacency matrices used for marginal probabilities.
<code>self_links</code>	Whether a species may interact with itself when its name appears in both groups.
<code>seed</code>	Integer seed. A cell identifier is added to this seed so spatial cells receive distinct reproducible ensembles. The previous global random state is restored after inference. Use <code>NULL</code> to use normal R RNG behavior.

burn_in	Number of degree-preserving swap proposals before recording a degree-constrained ensemble.
thin	Number of swap proposals between recorded matrices.
keep_ensemble	Store binary ensemble members in network metadata.

Details

For `constraint = "connectance"` or `"links"`, networks are sampled uniformly from all allowed adjacency matrices with exactly the requested number of links. This is the maximum-entropy distribution under a hard link count. For `constraint = "degree_sequence"`, a feasible matrix is found by maximum flow and randomized with degree-preserving checkerboard swaps. This is an approximate uniform MCMC ensemble conditional on the supplied margins.

The implementation follows the constrained, minimally biased ensemble rationale of Banville, Gravel & Poisot (2023), but does not yet implement their SVD-entropy objective or simulated-annealing optimizer.

Value

An `inference_model` compatible with `infer_network()` and `infer_networks()`.

References

Banville F, Gravel D, Poisot T (2023). What constrains food webs? A maximum entropy framework for predicting their structure with minimal biases. PLOS Computational Biology 19: e1011458. [doi:10.1371/journal.pcbi.1011458](https://doi.org/10.1371/journal.pcbi.1011458)

Examples

```
assemblage <- new_assemblage(list(food_web = c("a", "b", "c")))
model <- maxent_model(
  row_group = "food_web",
  constraint = "connectance",
  connectance = 0.25,
  ensemble_size = 50,
  self_links = FALSE,
  seed = 42
)
network <- infer_network(assemblage, model)
network$matrix
```

network_metrics_raster

Compatibility alias returning metric rasters

Description

Compatibility alias returning metric rasters

Usage

```
network_metrics_raster(x, metrics = NULL)
```

Arguments

x	Network collection with metrics in its index.
metrics	Metric columns; inferred when omitted.

new_assemblage	<i>Construct a local assemblage</i>
----------------	-------------------------------------

Description

Construct a local assemblage

Usage

```
new_assemblage(species, cell_id = NA_integer_, coordinates = NULL)
```

Arguments

species	Named list of character vectors, one per ecological group.
cell_id	Optional raster cell identifier.
coordinates	Optional named numeric vector containing x and y.

new_distribution_collection	<i>Construct a distribution collection</i>
-----------------------------	--

Description

Construct a distribution collection

Usage

```
new_distribution_collection(data, guild = NULL, metadata = NULL)
```

Arguments

data	A named terra::SpatRaster whose layers represent species.
guild	Optional guild label for the collection.
metadata	Optional species metadata data frame.

```
new_ecological_network
```

Construct an ecological network

Description

Construct an ecological network

Usage

```
new_ecological_network(
  matrix,
  type = c("probability", "binary", "weighted", "observed"),
  metadata = list(),
  row_group = "rows",
  column_group = "columns",
  directed = FALSE,
  self_links = NA
)
```

Arguments

<code>matrix</code>	Numeric interaction matrix.
<code>type</code>	One of probability, binary, weighted, or observed.
<code>metadata</code>	Optional metadata.
<code>row_group, column_group</code>	Ecological groups represented by rows and columns.
<code>directed</code>	Whether matrix orientation represents directed interactions.
<code>self_links</code>	Whether self-links are permitted; NA when not applicable.

```
new_inference_model
```

Construct a pluggable inference model

Description

Construct a pluggable inference model

Usage

```
new_inference_model(
  predict,
  name = "custom",
  parameters = list(),
  subclass = NULL
)
```

Arguments

predict	A function accepting (model, assemblage) and returning an ecological network.
name	Model name.
parameters	Model-specific parameters.
subclass	Optional additional class name.

`new_network_collection`*Construct a collection of spatial ecological networks*

Description

Construct a collection of spatial ecological networks

Usage

```
new_network_collection(networks, index, template = NULL, settings = list())
```

Arguments

networks	List of ecological networks.
index	Cell-level data frame.
template	Spatial raster template.
settings	Optional workflow settings.

`new_spatial_result` *Construct a spatial result*

Description

Construct a spatial result

Usage

```
new_spatial_result(data, metrics = names(data))
```

Arguments

data	A spatial object.
metrics	Names of mapped metrics.

```
prepare_range_stacks
```

Prepare palm and mammal range stacks

Description

Prepare palm and mammal range stacks

Usage

```
prepare_range_stacks(
  palm_source,
  mammal_source,
  mammal_species_col,
  mammal_species_keep = NULL,
  template = create_standard_grid(),
  region = NULL,
  output_dir = NULL,
  touches = TRUE
)
```

Arguments

`palm_source, mammal_source`
Range sources accepted by `build_range_stack()`.

`mammal_species_col`
Species-name column in the mammal source.

`mammal_species_keep`
Optional mammal species subset.

`template`
Raster template.

`region`
Optional crop region.

`output_dir`
Optional directory in which to save the range stacks.

`touches`
Count all touched cells.

```
print.metaweave
```

Print a MetaWeave object

Description

Prints a compact summary of an assemblage, distribution collection, ecological network, inference model, or network collection. The object is returned invisibly. These methods are normally called through `print()`.

probability_matrix_model

Construct a species-level probability-matrix model

Description

Use this backend when an external model returns probabilities for named species pairs directly, including ensemble-averaged predictions for which no single coherent block partition exists.

Usage

```
probability_matrix_model(
  probability_matrix,
  row_group,
  column_group,
  name = "species probability matrix"
)
```

Arguments

probability_matrix	Numeric species-by-species probability matrix. Row and column names are required and must be unique.
row_group	Assemblage group corresponding to matrix rows.
column_group	Assemblage group corresponding to matrix columns.
name	Descriptive model name stored in outputs.

rarefied_network_metric

Placeholder adapter for rarefaction packages

Description

Simulates a standardized count matrix and evaluates a user-supplied metric function.

Usage

```
rarefied_network_metric(
  prob_mat,
  target_size = 100L,
  metric,
  n_per_level = 10L,
  seed = NULL
)
```

Arguments

prob_mat	Probability matrix.
target_size	Standardized number of interactions.
metric	A function accepting a count matrix.
n_per_level	Number of replicates.
seed	Optional seed.

rasterize_range_file *Rasterize one range file*

Description

Rasterize one range file

Usage

```
rasterize_range_file(shp_file, template, touches = TRUE)
```

Arguments

shp_file	Shapefile path.
template	Raster template.
touches	Count all touched cells.

rasterize_range_rows *Rasterize rows grouped by species*

Description

Rasterize rows grouped by species

Usage

```
rasterize_range_rows(
  x,
  template,
  species_col,
  species_keep = NULL,
  touches = TRUE
)
```

Arguments

x	An sf or SpatVector object.
template	Raster template.
species_col	Species-name column.
species_keep	Optional species subset.
touches	Count all touched cells.

reconstruct_networks *Reconstruct spatial networks from arbitrary distribution groups*

Description

Reconstruct spatial networks from arbitrary distribution groups

Usage

```
reconstruct_networks(distributions, model, min_species = 1L)
```

Arguments

distributions	Named list of aligned distribution rasters or collections.
model	Inference model whose group names match distributions.
min_species	Minimum number of locally present species required in every group.

run_network_downscaling
 Complete compatibility workflow

Description

Complete compatibility workflow

Usage

```
run_network_downscaling(...)
```

Arguments

... Arguments passed to `downscale_networks()`.

`run_spatial_inference` *Run generic spatial ecological-network inference*

Description

Run generic spatial ecological-network inference

Usage

```
run_spatial_inference(distributions, model, min_species = 1L, threshold = NULL)
```

Arguments

<code>distributions</code>	Named distribution groups.
<code>model</code>	Inference model.
<code>min_species</code>	Minimum local richness in each group.
<code>threshold</code>	Optional probability threshold for network summaries.

`sbm_model` *Construct the original palm-mammal SBM backend*

Description

Compatibility wrapper around `block_model()`.

Usage

```
sbm_model(
  palm_lookup,
  mammal_lookup,
  theta,
  species_col = "species",
  guild_col = "guild"
)
```

Arguments

<code>palm_lookup, mammal_lookup</code>	Species-to-guild lookup tables.
<code>theta</code>	Guild-by-guild probability matrix.
<code>species_col</code>	Species column in lookup tables.
<code>guild_col</code>	Guild column in lookup tables.

simulate_interaction_counts	<i>Simulate an interaction-count matrix</i>
-----------------------------	---

Description

Simulate an interaction-count matrix

Usage

```
simulate_interaction_counts(network, size = 100L, seed = NULL)
```

Arguments

network	Ecological network or probability matrix.
size	Total number of interactions.
seed	Optional random seed.

simulate_network	<i>Simulate a network</i>
------------------	---------------------------

Description

Simulate a network

Usage

```
simulate_network(network, size = 100L, seed = NULL)
```

Arguments

network	Probability network.
size	Total interaction count.
seed	Optional seed.

summarize_network	<i>Summarize one ecological network</i>
-------------------	---

Description

Summarize one ecological network

Usage

```
summarize_network(network, threshold = NULL)
```

Arguments

network	Ecological network or numeric matrix.
threshold	Optional threshold for realized links.

summarize_networks	<i>Summarize a collection of networks</i>
--------------------	---

Description

Summarize a collection of networks

Usage

```
summarize_networks(x, threshold = NULL)
```

Arguments

x	Network collection.
threshold	Optional probability threshold.

Index

add_network_metrics, [3](#)
assemble_communities, [3](#)

block_model, [4](#)
build_range_stack, [4](#)

create_standard_grid, [5](#)
crop_filter_stack, [5](#)

downscale_networks, [6](#)

infer_network, [6](#)
infer_network(), [12](#)
infer_networks, [7](#)
infer_networks(), [12](#)
interaction_table_to_matrix, [7](#)

list_shapefiles, [8](#)
local_block_network, [8](#)
local_probability_network, [9](#)

make_bbox, [10](#)
map_metrics, [10](#)
maxent_model, [11](#)

network_metrics_raster, [12](#)
new_assemblage, [13](#)
new_distribution_collection, [13](#)
new_ecological_network, [14](#)
new_inference_model, [14](#)
new_network_collection, [15](#)
new_spatial_result, [15](#)

prepare_range_stacks, [16](#)
print.assemblage (print.metaweave), [16](#)
print.distribution_collection
 (print.metaweave), [16](#)
print.ecological_network
 (print.metaweave), [16](#)
print.inference_model
 (print.metaweave), [16](#)

print.metaweave, [16](#)
print.network_collection
 (print.metaweave), [16](#)
probability_matrix_model, [17](#)

rarefied_network_metric, [17](#)
rasterize_range_file, [18](#)
rasterize_range_rows, [18](#)
reconstruct_networks, [19](#)
run_network_downscaling, [19](#)
run_spatial_inference, [20](#)

sbm_model, [20](#)
simulate_interaction_counts, [21](#)
simulate_network, [21](#)
summarize_network, [22](#)
summarize_networks, [22](#)