Package 'metaweave'

November 3, 2026

Title Spatial Ecological Network Inference

Version 1.0.0

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

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/roxygen2/version 8.1.0

NeedsCompilation no

Author Gabriel Munoz [aut, cre, cph] (Affiliation: Concordia University)

Maintainer Gabriel Munoz <gmunozacevedo@proton.me>

Repository CRAN

Date/Publication 2026-11-03 03:10:03 UTC

Contents

add_network_metrics 2
assemble_communities 3
block_model 3
build_range_stack 4

create_standard_grid . . . 5
crop_filter_stack . . . 6
downscale_networks . . . 6
infer_network . . . 7
infer_networks . . . 8
interaction_table_to_matrix . . . 8
list_shapefiles . . . 9
local_block_network . . . 9
local_probability_network . . . 10
make_bbox . . . 11
map_metrics . . . 11
maxent_model . . . 12
network_metrics_raster . . . 14
new_assemblage . . . 14
new_distribution_collection . . . 15
new_ecological_network . . . 15
new_inference_model . . . 16
new_network_collection . . . 17
new_spatial_result . . . 17
prepare_range_stacks . . . 18
print.metaweave . . . 19
probability_matrix_model . . . 19
rarefied_network_metric . . . 20
rasterize_range_file . . . 21
rasterize_range_rows . . . 22
reconstruct_networks . . . 22
run_network_downscaling . . . 23
run_spatial_inference . . . 23
sbm_model . . . 24
simulate_interaction_counts . . . 24
simulate_network . . . 25
summarize_network . . . 26
summarize_networks . . . 26

Index **28**

`add_network_metrics` *Add standard metrics to network collection*

Description

Add standard metrics to network collection

Usage

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

Arguments

x	Network collection.
threshold	Optional probability threshold.

Value

A network_collection with summary columns appended to index; see summarize_networks().

---

assemble_communities *Assemble local communities from aligned distribution rasters*

Description

Assemble local communities from aligned distribution rasters

Usage

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

Arguments

distributions	Nonempty named list of SpatRaster or distribution_collection objects with unique group and species names, aligned geometry, and nonnegative values. Zero and NA indicate absence.
min_species	Positive integer minimum species per group.

Value

A list of assemblage objects, one per cell meeting min_species in every group. Nonzero, nonmissing raster values indicate presence; the list is empty if no cell qualifies.

---

block_model *Construct a generic stochastic block model backend*

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	Finite guild-by-guild probability matrix with unique row and column guild names covering the supplied lookup guilds.
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.

Value

An `inference_model` with subclass `block_model`. Prediction returns a probability network using the supplied block assignments and probabilities; no model fitting is performed. Unmatched local species are listed in `metadata$excluded_rows` and `metadata$excluded_columns`.

---

build_range_stack *Build a species range stack*

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.

Value

A terra::SpatRaster with named species layers, optionally cropped and filtered to the region.

---

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.

Value

A terra::SpatRaster with the requested extent, resolution, and coordinate reference system.

Examples

```
create_standard_grid(c(0, 2, 0, 2), resolution = 1)
```

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.

Value

A `terra::SpatRaster` containing layers with nonmissing cells in the crop region. Zero values count as nonmissing; this helper expects presence/NA range rasters.

downscale_networks *Compatibility wrapper for spatial network downscaling*

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.

Value

A network_collection with local probability networks, cell index, and raster template.

---

infer_network *Infer a network for one assemblage*

Description

Infer a network for one assemblage

Usage

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

Arguments

assemblage	An assemblage.
model	An inference model.

Value

The ecological_network returned by the model prediction function.

infer_networks *Infer networks for many assemblages*

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.

Value

A network_collection containing one network per assemblage and an index with cell_id, x, and y columns. An empty input returns a zero-network collection with a zero-row index.

interaction_table_to_matrix *Convert an interaction table to a matrix*

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.

Value

A numeric matrix with row and column species names. Unspecified pairs receive fill; repeated pairs use the last supplied value.

Examples

```
records <- data.frame(plant = c("a", "b"), animal = c("x", "x"), p = c(0.8, 0.3))
interaction_table_to_matrix(records, "plant", "animal", "p")
```

---

list_shapefiles *List ESRI shapefiles*

Description

List ESRI shapefiles

Usage

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

Arguments

path	Directory.
recursive	Search recursively.

Value

A character vector of shapefile paths, empty if no matches are found.

---

local_block_network *Build one local generic block-model network directly*

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.

Value

A numeric probability matrix with species row and column names.

---

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.

Value

A numeric palm-by-mammal probability matrix.

`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.

Value

A polygon `terra::SpatVector` representing the bounding box.

`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.

Value

A `spatial_result` with a multilayer `terra::SpatRaster` in `data`, one layer per requested metric, and metric names in `metrics`.

maxent_model *Maximum-entropy ecological network model*

Description

Experimental: this backend remains available in MetaWeave 1.0, but its degree-sequence sampler and scientific interpretation are not part of the stable 1.x API. Review the constraints and sampling limits below before using its output for inference.

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

`row_group`	Assemblage group represented by matrix rows (consumers).
`column_group`	Assemblage group represented by matrix columns (resources). May equal `row_group` for a directed unipartite food web.
`constraint`	One of `"connectance"`, `"links"`, or `"degree_sequence"`.
`connectance`	Desired connectance in `[0, 1]`. Converted to an exact local link count with `round(connectance * allowed_dyads)`.
`links`	Exact number of interactions in every local ensemble member.
`out_degree`	Row sums (numbers of resources per consumer).
`in_degree`	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.
`ensemble_size`	Number of adjacency matrices used for marginal probabilities.

self_links	Whether a species may interact with itself when its name appears in both groups.
seed	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 NULL 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 degree-preserving swap chain has no built-in convergence diagnostic; burn_in and thin count proposals, not accepted swaps. Marginal probabilities from this route should be checked for sensitivity to these settings and to the seed.

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

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.

Value

A terra::SpatRaster with one layer per requested metric.

new_assemblage *Construct a local assemblage*

Description

Construct a local assemblage

Usage

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

Arguments

species	Nonempty named list of species vectors, one per ecological group. Group names and species names within a group must be unique and nonempty; species names are coerced to character.
cell_id	Optional positive integer raster cell identifier or NA.
coordinates	Optional two finite numeric values in x/y order.

Value

An assemblage list with species, cell_id, and coordinates.

Examples

```
new_assemblage(list(plants = c("a", "b"), animals = "x"), cell_id = 1)
```

new_distribution_collection

Construct a distribution collection

Description

Construct a distribution collection

Usage

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

Arguments

data	A nonempty terra::SpatRaster with unique, nonempty species layer names.
guild	Optional guild label for the collection.
metadata	Optional species metadata data frame.

Value

A distribution_collection list with data (the raster), guild, and metadata.

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

matrix	Numeric interaction matrix. Probabilities must be finite in [0, 1]; binary entries must be 0 or 1. Weighted and observed entries must be nonnegative and may be NA.
type	One of probability, binary, weighted, or observed.
metadata	Optional metadata.
row_group, column_group	Ecological groups represented by rows and columns.
directed	Whether matrix orientation represents directed interactions.
self_links	Whether self-links are permitted; NA when not applicable.

Value

An ecological_network list with matrix, type, metadata, row_group, column_group, directed, and self_links; also inherits from the type-specific network class.

---

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.

Value

An inference_model list containing name, predict, and parameters, with the optional subclass prepended.

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.

Value

A network_collection list with networks, index, template, and settings. Each index row corresponds to one network.

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.

Value

A spatial_result list with the spatial object in data and metric names in 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.

Value

A list with `palms` and `mammals`, each a `terra::SpatRaster`. If `output_dir` is supplied, RDS files are also written there.

`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()`.

Usage

```
## S3 method for class 'assemblage'
print(x, ...)
## S3 method for class 'distribution_collection'
print(x, ...)
## S3 method for class 'ecological_network'
print(x, ...)
## S3 method for class 'inference_model'
print(x, ...)
## S3 method for class 'network_collection'
print(x, ...)
```

Arguments

`x`	A metaweave object of the corresponding class.
`...`	Additional arguments, currently unused.

Value

The input object, returned invisibly after printing a compact summary.

`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.

Value

An `inference_model` with subclass `probability_matrix_model`. Prediction subsets the supplied matrix to locally present, matched species. Unmatched local species are listed in the resulting network's `metadata$excluded_rows` and `metadata$excluded_columns`.

Examples

```
p <- matrix(c(0.8, 0.3), 2, dimnames = list(c("a", "b"), "x"))
model <- probability_matrix_model(p, "plants", "animals")
site <- new_assemblage(list(plants = "a", animals = "x"))
infer_network(site, model)$matrix
```

---

rarefied_network_metric

Experimental rarefied network metric

Description

Simulates standardized count matrices and evaluates a user-supplied metric function. This helper is experimental in MetaWeave 1.0; its metric and random-number behavior are outside the stable 1.x API.

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.

Value

A numeric mean of the supplied metric across simulated count matrices, with missing metric values omitted.

---

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.

Value

A one-layer `terra::SpatRaster`, named from the file stem, with 1 for presence and `NA` for background.

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.

Value

A terra::SpatRaster with one named layer per retained species, using 1 for presence and NA for background.

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.

Value

A network_collection of locally inferred networks with a spatial template and cell index.

---

run_network_downscaling *Complete compatibility workflow*

Description

Complete compatibility workflow

Usage

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

Arguments

...	Arguments passed to downscale_networks().

Value

A list with result (a summarized network_collection) and rasters (a terra::SpatRaster of metrics).

---

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

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

Value

A list with result (a summarized network_collection) and spatial (a spatial_result containing metric rasters).

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

`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.

Value

A block-model `inference_model` using group names `palms` and `mammals`.

simulate_interaction_counts *Simulate an interaction-count matrix*

Description

Simulate an interaction-count matrix

Usage

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

Arguments

network	Ecological network or nonnegative numeric matrix.
size	Nonnegative integer total number of interactions.
seed	Optional nonnegative integer seed. A supplied seed produces a repeatable draw without changing the caller's global random state.

Value

An integer matrix with the input dimensions and names. Counts sum to `size` when positive probability mass exists; otherwise all entries are zero.

---

simulate_network *Simulate a network*

Description

Simulate a network

Usage

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

Arguments

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

Value

A weighted `ecological_network` containing simulated interaction counts and the source network in its metadata.

summarize_network *Summarize one ecological network*

Description

Summarize one ecological network

Usage

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

Arguments

network	Ecological network or numeric matrix.
threshold	Optional finite threshold for realized links; for probability networks it must lie in [0, 1].

Value

A one-row data frame with row_richness, column_richness, possible_links (number of eligible matrix entries), expected_links (sum of eligible entries, or number at least threshold), and mean_probability (mean of eligible nonmissing entries). For networks with self_links = FALSE, pairs with the same species name are excluded. Unnamed same-group square networks use the diagonal; unnamed networks across different groups require species names. Other inputs use all matrix entries, including missing entries in the possible-link count. Missing values are omitted from sums and means. Without eligible entries the mean is NA; if all are missing it is NaN. For weighted/observed networks these fields summarize weights, not link probabilities; use threshold to count entries at or above a cutoff.

Examples

```
p <- matrix(c(0.8, 0.3), 2)
summarize_network(p)
summarize_network(p, threshold = 0.5)
```

summarize_networks *Summarize a collection of networks*

Description

Summarize a collection of networks

Usage

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

Arguments

x	Network collection.
threshold	Optional probability threshold.

Value

The input network_collection with summary columns appended to its index; see summarize_network() for column definitions. A zero-network collection receives the same five columns with zero rows.

Index

add_network_metrics, 2
assemble_communities, 3

block_model, 3
build_range_stack, 4

create_standard_grid, 5
crop_filter_stack, 6

downscale_networks, 6

infer_network, 7
infer_network(), *13*
infer_networks, 8
infer_networks(), *13*
interaction_table_to_matrix, 8

list_shapefiles, 9
local_block_network, 9
local_probability_network, 10

make_bbox, 11
map_metrics, 11
maxent_model, 12

network_metrics_raster, 14
new_assemblage, 14
new_distribution_collection, 15
new_ecological_network, 15
new_inference_model, 16
new_network_collection, 17
new_spatial_result, 17

prepare_range_stacks, 18
print.assemblage *(*print.metaweave*)*, 19
print.distribution_collection *(*print.metaweave*)*, 19
print.ecological_network *(*print.metaweave*)*, 19
print.inference_model *(*print.metaweave*)*, 19
print.metaweave, 19
print.network_collection *(*print.metaweave*)*, 19
probability_matrix_model, 19

rarefied_network_metric, 20
rasterize_range_file, 21
rasterize_range_rows, 22
reconstruct_networks, 22
run_network_downscaling, 23
run_spatial_inference, 23

sbm_model, 24
simulate_interaction_counts, 24
simulate_network, 25
summarize_network, 26
summarize_network(), *27*
summarize_networks, 26
summarize_networks(), *3*