Package: bean (via r-universe)

October 21, 2026

Type Package

Title Data Thinning of Species Occurrences in Environmental Space

Version 0.2.2

Maintainer Paanwaris Paansri <paanwaris@vt.edu>

Description A suite of tools to mitigate sampling bias in species occurrence records by thinning data in the environmental space (E-space). This process can improve the accuracy and precision of species distribution models (SDM, also known as ecological niche models, ENM). The package offers a data-driven protocol to determine thinning parameters using kernel-density bandwidth selection. Two thinning methods are provided (stochastic and deterministic) to reduce over-sampled environmental conditions and down-weight outlier observations. The name 'bean' reflects the core principle of the method: each 'pod' (a grid cell in E-space) is allowed to contain only a limited number of 'beans' (occurrence points). See Silverman (1986, ISBN:978-0-412-24620-3) and Rousseeuw and Leroy (2003, ISBN:978-0-471-48855-2) for the underlying statistical methods.

License MIT + file LICENSE

URL https://github.com/paanwaris/bean, https://paanwaris.github.io/bean/

BugReports https://github.com/paanwaris/bean/issues

Encoding UTF-8

LazyData true

Depends R (>= 4.0)

Imports MASS, stats, terra

Suggests covr, knitr, rmarkdown, testthat (>= 3.0.0), ggplot2, rgl, nicheR

VignetteBuilder knitr

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

RoxygenNote 8.0.0

Repository https://paanwaris.r-universe.dev

Date/Publication 2026-06-19 16:43:27 UTC

RemoteUrl https://github.com/paanwaris/bean

RemoteRef HEAD

RemoteSha 3f76b69fa668aee12c1b333622e70ff75b4b4220

Contents

find_env_resolution 2
fit_ellipsoid 4
occ_data_raw 6
origin_dat_prepared 7
origin_ellipse 7
plot_bean 8
prepare_bean 8
thin_env_center 10
thin_env_nd 11
thinned_deterministic 12
thinned_stochastic 13

Index **14**

`find_env_resolution` *Find an objective environmental grid resolution*

Description

Calculates an objective, data-driven grid resolution for environmental thinning. For each environmental variable, the function selects a bandwidth for a kernel-density estimate (KDE) of the marginal distribution. The chosen bandwidth defines the spatial scale below which two observations carry essentially redundant information, and is therefore a natural choice for the edge length of an environmental grid cell.

Three established bandwidth selectors are supported (see Details):

- `"sheather-jones"` (default) — the Sheather–Jones direct plug-in estimator (Sheather & Jones, 1991), the modern recommended default for non-Gaussian data;
- `"silverman"` — Silverman's rule of thumb (Silverman, 1986);
- `"scott"` — Scott's rule (Scott, 1992).

Usage

```
find_env_resolution(
  data,
  env_vars,
  method = c("sheather-jones", "silverman", "scott")
)
```

Arguments

`data`	A `data.frame` containing the environmental variables.
`env_vars`	A character vector specifying the environmental variables to analyse.
`method`	The bandwidth selector. One of `"sheather-jones"` (default), `"silverman"`, or `"scott"`.

Details

Why a bandwidth? A good environmental grid cell should be small enough to distinguish ecologically meaningful differences, but large enough to absorb sampling noise. A kernel density bandwidth chosen from the data answers exactly that question: it is the scale at which the empirical density of observations becomes smooth. Using it as the grid resolution yields one occurrence per cell on average when the sampling intensity is near the mode of the data.

Selectors.

- *Sheather–Jones* (`stats::bw.SJ` with `method = "dpi"`) is a plug-in selector that is robust for non-Gaussian densities and is the standard recommendation in the modern literature (Sheather & Jones, 1991; Jones, Marron & Sheather, 1996). Recommended default.
- *Silverman* (`stats::bw.nrd0`) is the rule-of-thumb $h = 0.9 \min(\hat{\sigma}, IQR/1.34)\, n^{-1/5}$ (Silverman, 1986). Fast and stable, but assumes near-Gaussian shape.
- *Scott* (`stats::bw.nrd`) is the Gaussian-optimal rule $h = 1.06\, \hat{\sigma}\, n^{-1/5}$ (Scott, 1992). Simpler than Silverman but less robust to outliers.

If `"sheather-jones"` fails (this can happen with strongly tied data), the function falls back to Silverman's rule for that variable and emits a `message()`.

Value

An object of class `bean_env_resolution` (a list) with:

`suggested_resolution` A named numeric vector of the suggested grid resolution for each variable, in the units of that variable.

`bandwidths` The bandwidths used to derive each resolution (identical to `suggested_resolution`).

`density_data` A long-format `data.frame` of the kernel density estimates, used by `plot.bean_env_resolution`.

`method` The bandwidth selector that was used.

References

Sheather, S. J. & Jones, M. C. (1991). A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society: Series B*, 53(3), 683–690.

Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, London.

Scott, D. W. (1992). *Multivariate Density Estimation: Theory, Practice, and Visualization*. Wiley, New York.

Jones, M. C., Marron, J. S. & Sheather, S. J. (1996). A brief survey of bandwidth selection for density estimation. *Journal of the American Statistical Association*, 91(433), 401–407.

See Also

`thin_env_nd`, `thin_env_center`, `bw.SJ`, `bw.nrd0`.

Examples

```
set.seed(1)
df <- data.frame(
  bio1  = c(rnorm(200, 15, 2), rnorm(50, 25, 1)),
  bio12 = c(rnorm(200, 1200, 200), rnorm(50, 2500, 100))
)
res <- find_env_resolution(df, env_vars = c("bio1", "bio12"))
res$suggested_resolution
plot(res)
```

`fit_ellipsoid` *Fit an environmental niche ellipsoid*

Description

Fits an ellipsoid that encompasses a chosen proportion of the data points in an environmental space of two or more dimensions. The centroid and covariance matrix can be estimated either by the classical sample moments (`"covmat"`) or by the robust Minimum Volume Ellipsoid (`"mve"`; Rousseeuw, 1985). Points are classified as inside or outside the ellipsoid using a χ^2 cutoff on their squared Mahalanobis distance.

Usage

```
fit_ellipsoid(data, env_vars, method = "covmat", level = 0.95)
```

Arguments

`data`	A `data.frame` containing the environmental variables.
`env_vars`	A character vector of at least two column names in `data` representing the environmental variables.

`method`	One of `"covmat"` (default, classical) or `"mve"` (robust Minimum Volume Ellipsoid via `cov.mve`).
`level`	A single number in `(0, 1)`: the confidence level of the ellipsoid. Default `0.95`.

Details

Methods. `"covmat"` uses the sample mean and sample covariance matrix. It is optimal under multivariate normality but sensitive to outliers. `"mve"` (Rousseeuw, 1985) finds the smallest-volume ellipsoid that contains a fraction of the data and is robust to a moderate proportion of contaminating points.

Confidence level. Assuming approximate multivariate normality, the boundary of the ellipsoid is the set of points whose squared Mahalanobis distance equals `qchisq(level, df = n_dim)`.

Value

An object of class `c("bean_ellipsoid", "nicheR_ellipsoid")` (a list) with:

`centroid` Named vector of variable means / centre.

`covariance_matrix`, `cov_matrix` The covariance matrix used. Both names point to the same object; the former is kept for backward compatibility, the latter is the name expected by the **nicheR** package.

`Sigma_inv` The inverse of `cov_matrix`, pre-computed so that `nicheR::predict()` does not have to invert it on every call.

`dimensions` Integer, the number of environmental variables.

`var_names` Character vector of the variable names used to fit the ellipsoid.

`cl` Confidence level (same value as `parameters$level`); name expected by **nicheR**.

`chi2_cutoff` The chi-square threshold, `stats::qchisq(level, df = dimensions)`.

`niche_ellipse` A `data.frame` of polygon vertices for the 2-D ellipse. `NULL` when more than two variables are supplied (the 3-D mesh is generated lazily on plot).

`all_points_used` Complete-case input data.

`points_in_ellipse` Subset inside the ellipsoid.

`points_outside_ellipse` Subset outside the ellipsoid.

`inside_indices` Row indices (in `all_points_used`) classified as inside.

`parameters` List with `level` and `method`.

The object carries two S3 classes: `"bean_ellipsoid"` (used by `print()` and `plot()` in this package) and `"nicheR_ellipsoid"` (used by `nicheR::predict()`, available from CRAN). Both methods work on the same object; the appropriate one is dispatched depending on which package is attached.

References

If you intend to project a `bean_ellipsoid` into geographic space, please install the **nicheR** package and use its `predict()` method; the dual S3 class on the returned object allows `nicheR::predict()` to dispatch on it directly. If you use the prediction step in published work, please cite **nicheR**:

Castaneda-Guzman, M., Hughes, C., Paansri, P. & Cobos, M. E. (2026). *nicheR: Ellipsoid-Based Virtual Niches and Visualization.* R package version 0.1.0. https://github.com/castanedaM/nicheR.

Rousseeuw, P. J. (1985). Multivariate estimation with high breakdown point. In *Mathematical Statistics and Applications, Vol. B*, 283–297.

Van Aelst, S. & Rousseeuw, P. (2009). Minimum volume ellipsoid. *Wiley Interdisciplinary Reviews: Computational Statistics*, 1(1), 71–82.

Cobos, M. E., Osorio-Olvera, L., Soberón, J., Peterson, A. T., Barve, V. & Barve, N. (2024). ellipsenm: ecological niches' characterizations using ellipsoids. https://github.com/marlonecobos/ellipsenm.

Examples

```
set.seed(81)
env_data <- data.frame(
  BIO1  = c(rnorm(50, 10, 1), 30),
  BIO12 = c(rnorm(50, 20, 2), 50)
)
fit <- fit_ellipsoid(env_data, env_vars = c("BIO1", "BIO12"),
                     method = "covmat", level = 0.95)
print(fit)
plot(fit)
```

---

occ_data_raw *Raw Rusa unicolor occurrence data*

Description

Raw, unthinned occurrence records for Rusa unicolor (Sambar deer) in Thailand. Used to demonstrate the spatial clustering and environmental bias typical of SDM datasets.

Usage

```
occ_data_raw
```

Format

A `data.frame` with one row per occurrence and the columns:

species Species name.

x Longitude (decimal degrees).

y Latitude (decimal degrees).

Source

Example dataset shipped with the **bean** package.

origin_dat_prepared *Cleaned and scaled occurrence data*

Description

Output of prepare_bean applied to occ_data_raw using the bundled environmental rasters. Missing coordinates and records outside the raster extent have been removed, and environmental values have been extracted and standardised.

Usage

```
origin_dat_prepared
```

Format

A data.frame with the columns:

species Species name.

x, y Coordinates.

bio_1 Scaled annual mean temperature.

bio_4 Scaled temperature seasonality.

bio_12 Scaled annual precipitation.

bio_15 Scaled precipitation seasonality.

origin_ellipse *Fitted niche ellipsoid for Rusa unicolor*

Description

A bean_ellipsoid fitted to origin_dat_prepared representing the baseline environmental niche of the species.

Usage

```
origin_ellipse
```

Format

A bean_ellipsoid object (see fit_ellipsoid).

plot_bean *Visualize n-dimensional environmental thinning results*

Description

This function creates a scatterplot matrix (pairs plot) to visualize the results of n-dimensional environmental thinning using base R graphics. It can accept thinned objects from either density-based thinning (‘thin_env_nd‘) or deterministic centroid thinning (‘thin_env_center‘).

Usage

```
plot_bean(original_data, thinned_object, env_vars)
```

Arguments

original_data	A data.frame of the prepared, unthinned occurrence points.
thinned_object	The output object from ‘thin_env_nd()‘ or ‘thin_env_center()‘.
env_vars	A character vector of the environmental variables to plot.

Value

Invisibly returns ‘NULL‘. Draws a plot to the active graphics device.

Examples

```
data(origin_dat_prepared, package = "bean")
env_vars <- c("bio_1", "bio_12")
thinned <- thin_env_nd(
  data            = origin_dat_prepared,
  env_vars        = env_vars,
  grid_resolution = c(0.5, 0.5),
  seed            = 1
)
plot_bean(origin_dat_prepared, thinned, env_vars = env_vars)
```

prepare_bean *Prepare data for environmental thinning*

Description

This function serves as a pre-processing step to clean and prepare species occurrence data. It performs three key actions: 1. Removes records with missing longitude or latitude values. 2. Extracts environmental data from raster layers that are already scaled for each occurrence point. 3. Removes records that fall outside the raster extent or have missing environmental data. The final output is a clean data frame where the environmental variables have a mean of 0 and a standard deviation of 1.

Usage

```
prepare_bean(
  data,
  env_rasters,
  longitude,
  latitude,
  transform = c("scale", "pca", "none")
)
```

Arguments

data	A data.frame of species occurrences records, including columns for longitude and latitude.
env_rasters	A SpatRaster (from terra package) or RasterStack (from raster package) object of environmental variables.
longitude	(character) The name of the longitude column in data.
latitude	(character) The name of the latitude column in data.
transform	(character) The transformation to apply to the environmental rasters before extracting data. Options are "scale" (default), "pca", or "none". See Details.

Details

Environmental Variable Transformation

The transform argument allows for different pre-processing of the environmental raster layers to address issues like differing units and multicollinearity.

- "scale" (Default):** This is the standard approach to handle variables with different units (e.g., °C vs. mm). It transforms each raster layer to have a mean of 0 and a standard deviation of 1 (Baddeley et al., 2016). This process makes the variables equal variance. As a result, each variable contributes equally to the analysis, ensuring that the resulting resolutions are based on the relative distribution of data points within each environmental dimension, not their arbitrary original units (Beaugrand, 2024; Kléparski et al., 2021).

- "pca": This option performs a Principal Component Analysis (PCA) on the environmental rasters. This is a powerful technique for dealing with multicollinearity (highly correlated variables). It transforms the original rasters into a new set of uncorrelated layers (Principal Components) (Qiao et al., 2016). The function then extracts the PC scores for each occurrence point.

- "none": This option extracts the raw environmental values from the rasters without any transformation. This is suitable if your rasters are already scaled or if you have a specific reason to use the raw values.

Value

A data.frame containing the cleaned and scaled occurrence data, with the following columns:

`Original Columns`
: All columns from the input data are preserved for the valid records.

`Environmental Variables`
: New columns, named after the layers in env_rasters, containing the extracted and scaled environmental data.

References

Baddeley, A., Rubak, E. and Turner, R. (2016). Spatial point patterns: methodology and applications with R. CRC press.

Beaugrand, G. (2024). An ecological niche model that considers local relationships among variables: The Environmental String Model. Ecosphere, 15(10), e70015.

Kléparski, L., Beaugrand, G. and Edwards, M. (2021). Plankton biogeography in the North Atlantic Ocean and its adjacent seas: Species assemblages and environmental signatures. Ecology and Evolution, 11(10), 5135-5149.

Qiao, H., Peterson, A. T., Campbell, L. P., Soberón, J., Ji, L. and Escobar, L. E. (2016). NicheA: creating virtual species and ecological niches in multivariate environmental scenarios. Ecography, 39(8), 805-813.

Examples

```
env_file <- system.file("extdata", "thai_env.tif", package = "bean")
occ_file <- system.file("extdata", "Rusa_unicolor.csv", package = "bean")
if (nzchar(env_file) && nzchar(occ_file) &&
    requireNamespace("terra", quietly = TRUE)) {
  env <- terra::rast(env_file)
  occ <- read.csv(occ_file)
  prepared <- prepare_bean(
    data        = occ,
    env_rasters = env,
    longitude   = "x",
    latitude    = "y",
    transform   = "scale"
  )
  head(prepared)
}
```

---

thin_env_center *Deterministic centroid*

Description

This function thins species occurrence records by finding all occupied cells in a 2D environmental grid and returning a single new point at the exact center of each of those cells. This is a deterministic method.

Usage

```
thin_env_center(data, env_vars, grid_resolution)
```

Arguments

data	A data.frame containing species occurrence coordinates and the environmental variables.
env_vars	A character vector specifying the column names in data that represent the environmental variables to be used in the analysis.
grid_resolution	A numeric vector of length one or two specifying the resolution(s) for the grid axes. If length one, it is used for both axes.

Value

An object of class `bean_thinned_center`. which is a list containing:

thinned_points	A data.frame with two columns representing the new points at the center of each occupied environmental grid cell.
n_original	An integer representing the number of complete occurrence records in the input data.
n_thinned	An integer representing the number of unique grid cells that were occupied, which is also the number of points returned.
parameters	A list of the key parameters used, such as whether scaling was applied.

See Also

thin_env_nd, find_env_resolution

Examples

```
env_data <- data.frame(
  BIO1  = c(0.1, 0.2, 1.1, 1.2, 1.3),
  BIO12 = c(0.1, 0.2, 2.1, 2.2, 2.3)
)
thin_env_center(env_data, env_vars = c("BIO1", "BIO12"),
                grid_resolution = c(0.1, 0.2))
```

---

thin_env_nd *Thin occurrence data in n-dimensional environmental space*

Description

This function thins species occurrence records in an n-dimensional environmental space by randomly sampling exactly one point from each occupied n-dimensional grid cell (hypercube).

Usage

```
thin_env_nd(data, env_vars, grid_resolution, seed = NULL)
```

Arguments

data	A data.frame containing species occurrences and pre-scaled environmental variables, typically the output of 'prepare_bean()'.
env_vars	A character vector of two or more column names representing the environmental variables (dimensions) to use for thinning.
grid_resolution	A numeric vector of resolutions for each environmental axis. Its length must match the length of 'env_vars'.
seed	(numeric) An optional random seed for reproducibility. If provided, the random number generator state is safely isolated to this function call and will not affect the global environment. Default = NULL.

Value

An object of class 'bean_thinned', which is a list containing:

thinned_data	A data.frame containing the occurrence records that were retained after the thinning process.
n_original	An integer representing the number of complete occurrence records in the input data before thinning.
n_thinned	An integer representing the number of occurrence records remaining after thinning.
parameters	A list of the key parameters used during the thinning process.

Examples

```
data(origin_dat_prepared, package = "bean")
thinned <- thin_env_nd(
  data            = origin_dat_prepared,
  env_vars        = c("bio_1", "bio_12"),
  grid_resolution = c(0.5, 0.5),
  seed            = 123
)
print(thinned)
```

---

thinned_deterministic *Deterministically thinned environmental data*

Description

Result of thin_env_center applied to origin_dat_prepared. Contains one calculated centroid per occupied environmental grid cell.

Usage

```
thinned_deterministic
```

Format

A bean_thinned_center object (see thin_env_center).

---

thinned_stochastic *Stochastically thinned environmental data*

Description

Result of thin_env_nd applied to origin_dat_prepared. Contains one randomly chosen occurrence per occupied environmental grid cell.

Usage

```
thinned_stochastic
```

Format

A bean_thinned object (see thin_env_nd).

Index

∗ **datasets**
occ_data_raw, 6
origin_dat_prepared, 7
origin_ellipse, 7
thinned_deterministic, 12
thinned_stochastic, 13

bw.nrd0, *4*
bw.SJ, *4*

cov.mve, *5*

find_env_resolution, 2, *11*
fit_ellipsoid, 4, *7*

occ_data_raw, 6
origin_dat_prepared, 7, *7*, *12, 13*
origin_ellipse, 7

plot.bean_env_resolution, *3*
plot_bean, 8
prepare_bean, *7*, 8

thin_env_center, *4*, 10, *12, 13*
thin_env_nd, *4*, 11, *11*, *13*
thinned_deterministic, 12
thinned_stochastic, 13