

Package ‘calba’

December 18, 2025

Title Efficient Neighborhood Basal Area Metrics for Trees

Version 0.1.2

Description Fast ‘C++’-backed tools for computing conspecific and total neighborhood basal area in mapped forest plots. Includes unweighted and distance-weighted neighborhoods, multiple radii, decay kernels, and basic edge correction. Outputs are model-ready covariates for forest competition, growth, and survival models, following neighborhood modeling workflows commonly used in spatial ecology (e.g., Hülsmann et al. 2024 <[doi:10.1038/s41586-024-07118-4](https://doi.org/10.1038/s41586-024-07118-4)>).

License GPL-3

BugReports <https://github.com/mattocci27/calba/issues>

Encoding UTF-8

Imports Rcpp

Suggests testthat (>= 3.1.0), knitr, rmarkdown, ggplot2

LinkingTo Rcpp

Maintainer Masatoshi Katabuchi <mattocci27@gmail.com>

Author Masatoshi Katabuchi [aut, cre]

RoxygenNote 7.3.3

VignetteBuilder knitr

NeedsCompilation yes

Repository CRAN

Date/Publication 2025-12-18 13:40:07 UTC

Contents

ba_decay	2
ba_decay_long	3
ba_simple	4
count_con	6
count_total	7
neigh_ba	8
neigh_multi_r	9

ba_decay*Calculate Basal Area with Decay Function***Description**

This function calculates the basal area across a given set of trees, applying a decay effect based on distance and species identity for each tree within a given radius.

Usage

```
ba_decay(
  mu_values,
  sp,
  gx,
  gy,
  ba,
  r,
  exponential_normal = FALSE,
  edge_correction = c("none", "safe"),
  bounds = NULL
)
```

Arguments

<code>mu_values</code>	A numeric vector of decay parameters. Each value in ‘mu_values’ represents a decay factor that modifies how the basal area contribution diminishes with distance.
<code>sp</code>	A character vector containing species names for each tree.
<code>gx</code>	A numeric vector of x-coordinates for the trees.
<code>gy</code>	A numeric vector of y-coordinates for the trees.
<code>ba</code>	A numeric vector of basal area values for the trees.
<code>r</code>	A numeric scalar representing the radius to consider for neighboring trees.
<code>exponential_normal</code>	A logical value. If ‘FALSE’ (default), use exponential decay. If ‘TRUE’, use exponential-normal decay.
<code>edge_correction</code>	Character. See ‘ba_simple()’ for the ““safe”“ behavior that skips edge trees.
<code>bounds</code>	Optional numeric vector ‘c(xmin, xmax, ymin, ymax)’ giving the plot extent. When ‘NULL’, the range of ‘gx’/‘gy’ is used; supply bounds if your data do not span the full plot.

Details

The function applies an exponential decay model where the basal area contribution diminishes with distance from the focal tree:

$$\text{decayed basal area} = \text{ba} \cdot \exp\left(-\frac{\text{dist}}{\mu}\right)$$

where ‘mu’ is the decay parameter, ‘ba’ is the basal area, and ‘dist’ is the Euclidean distance between trees.

Value

A list with two matrices:

- ‘**con_ba_matrix**‘ A numeric matrix of basal areas with decay applied for conspecific (same species) trees.
- ‘**total_ba_matrix**‘ A numeric matrix of basal areas with decay applied for all trees (conspecific + heterospecific).

Examples

```
# Generate a sample dataset
set.seed(42)
sample_data <- data.frame(
  latin = sample(letters[1:4], 100, replace = TRUE),
  gx = runif(100, 0, 10),
  gy = runif(100, 0, 10),
  ba = runif(100, 10, 30)
)
mu_values <- c(1, 3, 5, 7)
ba_decay(
  mu_values = mu_values,
  sp = sample_data$latin,
  gx = sample_data$gx,
  gy = sample_data$gy,
  ba = sample_data$ba,
  r = 3,
  exponential_normal = FALSE
)
```

Description

Transform the matrices returned by ‘ba_decay’ into a tidy table that can be joined with other data or mapped with ‘ggplot2’.

Usage

```
ba_decay_long(
  mu_values,
  sp,
  gx,
  gy,
  ba,
  r,
  exponential_normal = FALSE,
  edge_correction = c("none", "safe"),
  bounds = NULL
)
```

Arguments

<code>mu_values</code>	Numeric vector of decay parameters.
<code>sp</code>	Character/factor species vector.
<code>gx</code>	Numeric x-coordinates.
<code>gy</code>	Numeric y-coordinates.
<code>ba</code>	Numeric basal area.
<code>r</code>	Positive radius threshold.
<code>exponential_normal</code>	Logical passed to ‘ba_decay’.
<code>edge_correction</code>	Character; see ‘ba_simple()‘ for the “safe” option.
<code>bounds</code>	Optional numeric vector ‘c(xmin, xmax, ymin, ymax)‘ giving the plot extent. When ‘NULL‘, the range of ‘gx’/‘gy‘ is used; supply bounds if your data do not span the full plot.

Value

A data frame with ‘tree_id’, ‘species’, ‘mu’, ‘con_ba’, and ‘total_ba’.

ba_simple

Calculate Simple Basal Area

Description

This function calculates the total basal area (conspecific + heterospecific) for a given set of tree data, focusing on the number of focal trees and a radius parameter. The basal area represents the cumulative cross-sectional area of tree trunks, either unadjusted or adjusted for a simple distance-weighted decay model.

Usage

```
ba_simple(
  sp,
  gx,
  gy,
  ba,
  r,
  dist_weighted = FALSE,
  edge_correction = c("none", "safe"),
  bounds = NULL
)
```

Arguments

sp	A character vector containing species names for each tree.
gx	A numeric vector of x-coordinates for the trees (e.g., 0 to 10 for realistic spatial data).
gy	A numeric vector of y-coordinates for the trees (e.g., 0 to 10 for realistic spatial data).
ba	A numeric vector of basal area values for the trees (e.g., cross-sectional area at breast height).
r	A numeric scalar for the radius parameter. This parameter represents the maximum distance to consider when summing basal areas of neighboring trees (e.g., 1–5 units for spatially distributed data).
dist_weighted	Logical. If ‘TRUE’, use the distance-weighted approach (‘ba / dist’); if ‘FALSE’, use the unadjusted ‘ba’.
edge_correction	Character. Use ‘"none"’ (default) to return all focal trees or ‘"safe"’ to skip neighborhood calculations for trees within ‘r’ of the plot edges and mark their output as ‘NA’.
bounds	Optional numeric vector ‘c(xmin, xmax, ymin, ymax)’ giving the plot extent. When ‘NULL’, the range of ‘gx’/‘gy’ is used; supply bounds if your data do not span the full plot.

Details

The function either:

- Calculates the unadjusted basal area if ‘dist_weighted = FALSE’.
- Applies a simple distance-weighted approach (‘ba / dist’) if ‘dist_weighted = TRUE’.

Value

A list with two numeric vectors:

- ‘con_ba’ A numeric vector representing the cumulative basal area of conspecific trees within the radius.
- ‘total_ba’ A numeric vector representing the cumulative basal area of all trees (conspecific + heterospecific) within the radius.

Examples

```
# Generate a sample dataset
set.seed(42) # For reproducibility
sample_data <- data.frame(
  latin = sample(letters[1:4], 100, replace = TRUE),
  gx = runif(100, 0, 10), # Spatial coordinates between 0 and 10
  gy = runif(100, 0, 10),
  ba = runif(100, 10, 30) # Basal area between 10 and 30
)

# Calculate with distance weighting
ba_simple(
  sp = sample_data$latin,
  gx = sample_data$gx,
  gy = sample_data$gy,
  ba = sample_data$ba,
  r = 3, # Radius within the spatial scale
  dist_weighted = TRUE
)

# Calculate without distance weighting
ba_simple(
  sp = sample_data$latin,
  gx = sample_data$gx,
  gy = sample_data$gy,
  ba = sample_data$ba,
  r = 3,
  dist_weighted = FALSE
)
```

count_con

Count Conspecific Trees

Description

This function counts the number of conspecific trees within a given radius for each focal tree.

Usage

```
count_con(sp, gx, gy, r, edge_correction = c("none", "safe"), bounds = NULL)
```

Arguments

sp	A character vector of species names.
gx	A numeric vector of x-coordinates for the trees.
gy	A numeric vector of y-coordinates for the trees.
r	A numeric scalar for the radius parameter.

edge_correction	Character; see ‘ba_simple()‘ for the “safe” option that skips focal trees near the boundary.
bounds	Optional numeric vector ‘c(xmin, xmax, ymin, ymax)‘ giving the plot extent. When ‘NULL‘, the range of ‘gx‘/‘gy‘ is used; supply bounds if your data do not span the full plot.

Value

A numeric vector containing the count of conspecific trees within the radius for each focal tree.

Examples

```
sample_data <- data.frame(
  latin = sample(letters[1:4], 100, replace = TRUE),
  gx = runif(100, 0, 10),
  gy = runif(100, 0, 10)
)
count_con(
  sp = sample_data$latin,
  gx = sample_data$gx,
  gy = sample_data$gy,
  r = 3
)
```

count_total*Count Total Trees***Description**

This function counts the total number of trees within a given radius for each focal tree.

Usage

```
count_total(gx, gy, r, edge_correction = c("none", "safe"), bounds = NULL)
```

Arguments

gx	A numeric vector of x-coordinates for the trees.
gy	A numeric vector of y-coordinates for the trees.
r	A numeric scalar for the radius parameter.
edge_correction	Character; see ‘ba_simple()‘ for the “safe” option that skips focal trees close to the edges.
bounds	Optional numeric vector ‘c(xmin, xmax, ymin, ymax)‘ giving the plot extent. When ‘NULL‘, the range of ‘gx‘/‘gy‘ is used; supply bounds if your data do not span the full plot.

Value

A numeric vector containing the count of all trees within the radius for each focal tree.

Examples

```
sample_data <- data.frame(
  gx = runif(100, 0, 10),
  gy = runif(100, 0, 10)
)
count_total(
  gx = sample_data$gx,
  gy = sample_data$gy,
  r = 3
)
```

neigh_ba

Neighborhood summaries for basal area and counts

Description

Provide a tidy summary of pairwise neighborhood basal area and counts, optionally including decay results for a vector of decay parameters.

Usage

```
neigh_ba(
  sp,
  gx,
  gy,
  ba,
  r,
  mu_values = NULL,
  dist_weighted = FALSE,
  exponential_normal = FALSE,
  edge_correction = c("none", "safe"),
  bounds = NULL
)
```

Arguments

sp	A character or factor vector of species names.
gx	Numeric x-coordinates of each tree.
gy	Numeric y-coordinates of each tree.
ba	Numeric basal area of each tree.
r	Positive numeric radius within which neighbors are considered.

<code>mu_values</code>	Optional numeric vector of decay parameters. When ‘NULL’, the decay table is omitted.
<code>dist_weighted</code>	Logical flag passed to ‘ba_simple’ to use a simple ‘ba / dist’ weighting when ‘TRUE’.
<code>exponential_normal</code>	Logical passed to ‘ba_decay’ to select the exponential-normal kernel.
<code>edge_correction</code>	Character; see ‘ba_simple()’ for the “safe” option that skips edge focal trees.
<code>bounds</code>	Optional numeric vector ‘c(xmin, xmax, ymin, ymax)’ giving the plot extent. When ‘NULL’, the range of ‘gx’/‘gy’ is used; supply bounds if your data do not span the full plot.

Value

A list with * ‘summary’: data frame with ‘tree_id’, ‘species’, ‘con_ba’, ‘total_ba’, ‘con_count’, ‘total_count’. * ‘decay’: (‘NULL’ or) long data frame with ‘tree_id’, ‘species’, ‘mu’, ‘con_ba’, ‘total_ba’.

The ‘summary’ component also includes derived columns: ‘prop_con_ba’, ‘het_ba’, ‘het_count’, and ‘competition_index’.

Examples

```
sample_data <- data.frame(
  latin = sample(letters[1:4], 10, replace = TRUE),
  gx = runif(10, 0, 10),
  gy = runif(10, 0, 10),
  ba = runif(10, 10, 30)
)
neigh_ba(
  sp = sample_data$latin,
  gx = sample_data$gx,
  gy = sample_data$gy,
  ba = sample_data$ba,
  r = 3,
  mu_values = c(1, 3)
)
```

Description

Compute basal area sums and counts for each tree across multiple radii in one pass.

Usage

```
neigh_multi_r(
  sp,
  gx,
  gy,
  ba,
  r_values,
  dist_weighted = FALSE,
  edge_correction = c("none", "safe"),
  bounds = NULL
)
```

Arguments

sp	A character or factor vector of species names.
gx	Numeric x-coordinates of the trees.
gy	Numeric y-coordinates of the trees.
ba	Numeric basal area for each tree.
r_values	Numeric vector of radii to evaluate.
dist_weighted	Logical flag to use ‘ba / dist‘ weighting within each radius.
edge_correction	Character; see ‘ba_simple()‘ for the “safe”‘ option.
bounds	Optional numeric vector ‘c(xmin, xmax, ymin, ymax)‘ giving the plot extent. When ‘NULL‘, the range of ‘gx‘/‘gy‘ is used; supply bounds if your data do not span the full plot.

Value

A tidy tibble with ‘tree_id‘, ‘species‘, ‘radius‘, ‘con_ba‘, ‘total_ba‘, ‘con_count‘, ‘total_count‘, ‘prop_con_ba‘, ‘het_ba‘, ‘het_count‘, and ‘competition_index‘.

Examples

```
sample_data <- data.frame(
  latin = sample(letters[1:4], 10, replace = TRUE),
  gx = runif(10, 0, 10),
  gy = runif(10, 0, 10),
  ba = runif(10, 10, 30)
)
neigh_multi_r(
  sp = sample_data$latin,
  gx = sample_data$gx,
  gy = sample_data$gy,
  ba = sample_data$ba,
  r_values = c(3, 5)
)
```

Index

ba_decay, [2](#)
ba_decay_long, [3](#)
ba_simple, [4](#)

count_con, [6](#)
count_total, [7](#)

neigh_ba, [8](#)
neigh_multi_r, [9](#)