Package: spatbiome (via r-universe)

November 28, 2026

Title A Nonparametric Test for Co-Clustering in Spatial Microbiome Data

Version 0.1.0

Description Implements Area-adjusted Random Shift, a non-parametric test to assess co-clustering of two point patterns. The two point processes are assumed to be independently superimposed under the null hypothesis. The test is based on Ripley's bivariate K-function and uses random shifts of one of the point patterns to generate a null distribution. An area-adjusted scale factor stabilises the variance across random shifts.

Imports GET, future, future.apply, parallel, spatstat.explore, spatstat.geom, spatstat.random

License AGPL (>= 3)

Config/roxygen2/version 8.1.0

Encoding UTF-8

URL https://github.com/asmita112358/spatbiome

BugReports https://github.com/asmita112358/spatbiome/issues

NeedsCompilation no

Author Asmita Roy [aut, cre]

Maintainer Asmita Roy <aroy38@jh.edu>

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

Date/Publication 2026-11-27 22:09:13 UTC

RemoteUrl https://github.com/cran/spatbiome

RemoteRef HEAD

RemoteSha cedeb335bb35ee232c99ad886d1c2460b639fa19

Contents

ARShift 2

Index **5**

ARShift *Tests whether two types of points co-cluster*

Description

Function to implement Area-adjusted Random Shift (ARShift), a non-paramteric test to assess co-clustering of two point patterns. The test is based on the bivariate K-function and uses random shifts of one of the point patterns to generate a null distribution. The test is implemented in a parallelized manner for efficiency. As with bivariate Ripley's K-function, ARShift is directional. ARShift tests whether taxa/cell-type j co-clusters with taxa/cell-type i. This may be different from the co-clustering pattern of taxa/cell-type i with taxa/cell-type j. Therefore, it is recommended to run ARShift based on the scientific hypothesis of interest.

Usage

```
ARShift(
  data,
  i,
  j,
  r = NULL,
  rmax = NULL,
  correction = "isotropic",
  n_perm = 999,
  k = 1.1,
  type = "hom",
  sigma_val = NULL,
  lite = FALSE,
  thin_prob = 0.5,
  jump_radius = NULL,
  workers = 1,
  verbose = FALSE
)
```

Arguments

data	The observed point process pattern on which the test will be applied. It must be a multitype point pattern where marks are factor.
i	The type of points(mark) in data which stays fixed. Should be input as a character string.
j	The type of points(mark) in data which is shifted. Should be input as a character string. ARShift tests whether taxa/cell-type j co-clusters with taxa/cell-type i.
r	Numeric vector at which the K-function is evaluated. There is a sensible default.
rmax	Single numeric value indicating the maximum value of r to be considered. There is a sensible default.

correction	A character vector specifying the type of border correction to be used in the K function. Available options are "isotropic", "translate", "translation" and "border". For very large datasets "border" is recommended.
n_perm	Number of random shifts
k	factor by which number of permutations is increased. Defaults to 1.2. For sparse processes k can be increased.
type	The type of K-function to compute, homogneous or inhomogeneous. Can be "hom" or "inhom". Defaults to homogeneous K.
sigma_val	Bandwidth for evaluating inhomogeneous intensity in the inhomogeneous K. has a sensible default.
lite	logical, indicating whether the point process should be randomly thinned for efficiency. Recommended only for very large datasets. Deafults to FALSE.
thin_prob	numeric scalar between 0 and 1, indicating the thinning probability. Defaults to 0.5. Ignored if lite = FALSE.
jump_radius	Radius of the uniform disc from which the random shift vector is sampled. There is a sensible default.
workers	Number of cores to be used in case of parallel computation. Defaults to 1, serial computation.
verbose	Logical, indicating whether to print progress messages. Defaults to FALSE.

Value

A list containing:

p P-value for the area adjusted random shift test.

K Observed bivariate K-function.

r Radii at which the K-function was evaluated.

n_attempts Total number of shift attempts made. This is equal to ceiling(n_perm * k).

n_valid Number of valid random shift attempts. If this is less than n_perm, increase k in the function input.

n_perm Targeted number of random shifts.

Author(s)

Asmita Roy

Examples

```
win <- spatstat.geom::square(1)
a <- spatstat.random::rThomas(8, 0.05, 20, win)
b <- spatstat.random::rThomas(7, 0.04, 20, win)
spatstat.geom::marks(a) <- factor(
  rep(1, spatstat.geom::npoints(a)),
  levels = c(1, 2)
)
spatstat.geom::marks(b) <- factor(
```

```
  rep(2, spatstat.geom::npoints(b)),
  levels = c(1, 2)
)
X <- spatstat.geom::superimpose(a, b, W = win)
spatstat.geom::marks(X) <- factor(
  spatstat.geom::marks(X),
  levels = c(1, 2)
)

res <- ARShift(
  data = X,
  i = "1",
  j = "2",
  correction = "isotropic",
  n_perm = 499,
  workers = 5,
  type = "hom"
)
print(res)
```

Index

ARShift, 2