Package: rvoronoi (via r-universe)

October 11, 2026

Type Package

Title Fast Voronoi Tessellation and Delaunay Triangulation using Fortune's Algorithm

Version 0.1.0.9005

Maintainer Mike Cheng <mikefc@coolbutuseless.com>

Description Fast voronoi tessellation and delaunay triangulation with Fortune's algorithm.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.2.0)

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Copyright The included voronoi C code was written by Steven Fortune and is Copyright (c) 1994 by AT&T Bell Laboratories.

Config/roxygen2/version 8.0.0

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

Date/Publication 2026-06-12 10:45:19 UTC

RemoteUrl https://github.com/coolbutuseless/rvoronoi

RemoteRef HEAD

RemoteSha 6965bc26ba953328c36ecfeefa69e8f38f1a85d2

Contents

delaunay 2
plot.del 3
plot.vor 4
voronoi 5

Index **7**

`delaunay` *Delaunay Triangulation*

Description

Delaunay Triangulation

Usage

```
delaunay(x, y, calc_polygons = TRUE, calc_areas = FALSE, calc_segments = FALSE)
```

Arguments

`x, y`	coordinates of seed sites. Duplicate points are not allowed.
`calc_polygons`	calculate polygon coordinates default: TRUE
`calc_areas`	calculate the areas of each triangle. Default: FALSE
`calc_segments`	calculate segments. Default: FALSE

Value

named list of data

sites coordinates of sites

ntris Number of triangles in this delaunay triangulation

tris data.frame. Each row specifies the indices of the three sites which define a single delaunay triangle. If `calc_areas = TRUE` then this data.frame also includes an `area` column

polygons data.frame of polygon coordinates with x,y and polygon index. Only calculated when `calc_polygons = TRUE`

segments The end-points and length of each unique undirected edge in the triangulation. Only calculated when `calc_segments = TRUE`

- **v1,v2** vertex indices referencing the original `sites`
- **x1,y1,x2,y2** coordinates of ends of segment
- **len** length of this segment

Examples

```
set.seed(1)
x <- runif(10)
y <- runif(10)
del <- delaunay(x, y)
plot(del)
del
```

plot.del *Simple plot of Delaunay triangulation*

Description

Simple plot of Delaunay triangulation

Usage

```
## S3 method for class 'del'
plot(
  x,
  sites = TRUE,
  tris = TRUE,
  segments = FALSE,
  site_pch = 19,
  site_col = "black",
  segment_col = "black",
  tri_col = rainbow(x$ntris),
  ...
)
```

Arguments

x	output from delaunay()
sites, tris, segments	logical. draw geometric feature
site_pch, site_col	graphics parameters for each input site
segment_col	colour for segments (length = 1)
tri_col	colour for polygons (length = 1 or N)
...	other arguments passed to plot()

Value

None

Examples

```
set.seed(1)
x <- runif(10)
y <- runif(10)
del <- delaunay(x, y)
plot(del)
```

plot.vor *Plot a Voronoi tessellation*

Description

Barebones plotting of Voronoi

Usage

```
## S3 method for class 'vor'
plot(
  x,
  sites = TRUE,
  labels = !sites,
  verts = TRUE,
  polys = TRUE,
  fsegs = !polys,
  isegs = !polys,
  bounds = TRUE,
  site_pch = ".",
  site_cex = 1,
  site_col = "black",
  vert_pch = 19,
  vert_cex = 0.3,
  vert_col = "black",
  label_cex = 0.5,
  fseg_col = "black",
  iseg_col = "red",
  poly_col = rainbow(vor$polygons$npolygons),
  buffer = 0,
  ...
)
```

Arguments

x	object returned by voronoi()
sites, labels, verts, polys, fsegs, isegs, bounds	logical values. Should this geometric feature be plotted? Default: TRUE
site_pch, site_cex, site_col	parameters for sites
vert_pch, vert_cex, vert_col	parameters for Voronoi vertices
label_cex	size of text labels for sites
fseg_col, iseg_col	colours for finite and infinite segments
poly_col	vector of colours for polygons

buffer	buffer around extents. default: 0
...	other arguments passed to plot()

Value

None.

Examples

```
set.seed(1)
x <- runif(10)
y <- runif(10)
vor <- voronoi(x, y)
plot(vor)
```

---

voronoi *Voronoi Tessellation*

Description

Voronoi Tessellation

Usage

```
voronoi(
  x,
  y,
  calc_polygons = TRUE,
  match_sites = TRUE,
  bound_segments = TRUE,
  merge_tolerance = 1e-10
)
```

Arguments

x, y	coordinates of seed sites. Duplicate points are not allowed.
calc_polygons	Logical. Should voronoi polygons be calculated? Default: TRUE
match_sites	Logical. Should the polygons be re-ordered to match the seed points? Default: TRUE. This option only makes sense when calc_polygons = TRUE.
bound_segments	logical. Default: TRUE. If calc_polygons = TRUE then this option is always TRUE.
merge_tolerance	Limit of how close the ends of a segment must be before the segment is collapsed to a single vertex Default: 1e-10

Value

An object of class "vor" which is a named list of data.frames.

sites data.frame of original sites

vertices data.frame of voronoi vertices. This is the raw output from Fortune's algorithm

segments data.frame of segments defined by 'line', 'v1' and 'v2'. 'line' is the row index into the 'lines' data.frame. 'v1' and 'v2' are the indices into 'vertices' which define the endpoints along the specified 'line'. If 'v1' or 'v2' are NA this indicates that the segment continues to infinity

polygons polygon information for each voronoi cell. Only calculated when `calc_polygons = TRUE`

- **npolygons** Number of polygons
- **coords** data.frame of x,y coordiantes and polygon index for all polygons
- **bbox** data.frame of polygon bounding box information. Each row represents a polygon
- **centrois** data.frame of polygon centroids. Each row represents a polygon

lines data.frame of line equations in the voronoi diagram of the form 'ax + by = c'

extents list of (xmin, ymin), (xmax, ymax) bounding box which encompasses all input sites and voronoi vertices

mvertices Merged and bounded vertices used to calculate polygons

msegments Merged and bounded segments used to calculate polygons

Examples

```
set.seed(1)
x <- runif(10)
y <- runif(10)
vor <- voronoi(x, y)
plot(vor)
vor
```

Index

delaunay, 2

plot.del, 3
plot.vor, 4

voronoi, 5