Alpha shape

March 27, 2019

`alphashape` *Computation of n dimension α-shape*

Description

Implementation in n dimension of the alpha shape using the Q-hull library

Package:	alphashape
Date:	2019-03-14
License:	GPL-2
LazyLoad:	yes

Author(s)

Pascal Omondiagbe
Tom Etherington
Maintainers: pascal Omondiagbe <omondiagbep@landcareresearch.co.nz>

References

http://www.qhull.org/html/qh-code.htm

`convex` *Convex hull in N-dimensions.*

Description

This function calculates the Convex hull in N-dimensions using the qhull library

Usage

```
convex(inputFile = NULL, point = NULL, options = NULL)
```

Arguments

`inputFile`	`inputFile` is an input file containing the point, where the first line is the number of dimension.
`point`	`point` is an `n`-by-`dim` matrix. The rows of `point` represent `n` points in `dim`-dimensional space.
`options`	String containing extra options for the underlying Qhull command.(See the Qhull documentation (`../doc/html/qdelaun.html`) for the available options.) The `Qbb` option is always passed to Qhull. The default options are `Qt`. The degenerate (zero area) regions are returned For silent operation, specify the option `Pp`.

See Also

Used internally by `convex`

`delaunay` *Delaunay triangulation in N-dimensions.*

Description

The Delaunay triangulation is a tessellation of the convex hull of the points such that no N-sphere defined by the N-triangles contains any other points from the set. This function calculates the Delaunay triangulation in N-dimensions using the qhull library

Usage

```
delaunay(inputFile = NULL, point = NULL, options = NULL,
  full = FALSE)
```

Arguments

`inputFile`	`inputFile` is an input file containing the point, where the first line is the number of dimension. T
`point`	`point` is an `n`-by-`dim` matrix. The rows of `point` represent `n` points in `dim`-dimensional space.
`options`	String containing extra options for the underlying Qhull command.(See the Qhull documentation (`../doc/html/qdelaun.html`) for the available options.) The `Qbb` option is always passed to Qhull. The default options are `Qcc Qc Qt Qz` for `dim` <4 and `Qcc Qc Qt Qx` for `dim`>=4. If neither of the `QJ` or `Qt` options are supplied, the `Qt` option is passed to Qhull. The `Qt` option ensures all Delaunay regions are simplical (e.g., triangles in 2-d). See `../doc/html/qdelaun.html` for more details. The degenerate (zero area) regions are returned For silent operation, specify the option `Pp`.
`full`	Return all information associated with the triangulation as a list and these are triangulation (`tri`), a vector of facet areas (`areas`) and a list of neighbours of each facet (`neighbours`) OR return the convexhull and the input point

`findSimplex` *Find Simplex*

Description

Returns the simplicies of the delaunay trigulation which contains a given point.

Usage

```
findSimplex(tri, inputPoint, testPoint)
```

Arguments

`tri`	delaunay trigulation simplex using `delaunay`
`inputPoint`	`n-by-dim` matrix of original inputPoint. \
`testPoint`	`n-by-dim` matrix of points to check.

Details

Given a grid point and a test point point, the find Simplex will identify the simplicies contianing the test point. It works by first checking for all point inside a convex hull, and then check if the center of the grid cell is inside the trigulation

Value

A `n*m` vector containing the result. -1 if a given point is outside the trigulation, or the trigulation index if the center of the cell is inside the alpha shape

`gridGeneration` *gridGeneration*

Description

Generate a `n*m` grid point by passing the vector of max and min point or file containing the initial data point or a matrix of input point

Usage

```
gridGeneration(inputFile = NULL, point = NULL, nCoords,
  minPoint = NULL, maxPoint = NULL)
```

Arguments

`nCoords`	`n` number of coordinates across the grid
`minPoint`	Vector of length `n` listing the minimum point for each dimension.
`maxPoint`	Vector of length `n` listing the maximum point for each dimension.
`inputfile`	file name containing the input point and number of dimension
`Point:`	Matrix. The input point.

Details

This function creates a grid by using the max/min values specify by the user. Note that when the user sepcify a matrix of point the maximum and minimum point will be derived from the input point and use when generating the grid (+ 0.3 is added to the max point, and -0.-3 is taken away from the min point). Point could be passed as a file or matrix of point

Value

A `n*m` vector of matrix @export gridGeneration

`inconvexhull` *in convex hull*

Description

To test if a given point is inside the convex hull. `TRUE` if the point lies within the hull and `FALSE` if it lies outwith the hull

Usage

```
inconvexhull(hull, points)
```

Arguments

`hull`	object Convex hull simplices produced using convex function
`points:`	matrix `n`-by-`dim` matrix of points to check.

Value

A `n*m` vector containing the result. True if a given point was inside the convexhull , otherwise false

---

`nalphaShape` *alphaShape*

Description

Compute an alpha Shape Grid using the Q-hull library.

Usage

```
nalphaShape(voronoiObject, alphaRange, n = NULL)
```

Arguments

`voronoiObject`	Object containing the trigulation, inputpoint, voronoi vertices and circumradii returned from the voronoi function. voronoiObject is produce using the function `voronoi`
`alphaRange,`	range of alpha value
`n`	`n` number of point to generate for this grid

Details

The calculation is done by assigning the trigulation index when the grid cell center lies within the trigulation or -1 if it lies outside

Value

grid stack as vector, gridSimplex, and the inputted grid point.

`voronoi` *Get Voronoi triangle and Delaunay triangulation*

Description

Get Voronoi vertices(circumcenters of Delaunay triangle) and Delaunay triangulation in N-dimensions using the qhull library. The Voronoi diagram is the nearest-neighbor map for a set of points. Each region contains those points that are nearer one input site than any other input site. They can also be describe as the circumcenters of Delaunay triangle. The Delaunay triangulation is a tessellation of the convex hull ofthe points such that no N-sphere defined by the N-triangles contains any other points from the set.

Usage

```
voronoi(inputFile = NULL, point = NULL, options = NULL,
  full = TRUE)
```

Arguments

`inputFile`	`inputFile` is an input file containing the point, where the first line is the number of dimension. T
`point`	`point` is an `n`-by-`dim` matrix. The rows of `point` represent `n` points in `dim`-dimensional space.
`options`	String containing extra options for the underlying Qhull command.(See the Qhull documentation (`../doc/html/qvoronoi.html`) for the available options.) The `Qbb` option is always passed to Qhull. The default options are `Qcc Qc Qt Qz` for `dim` <4 and `Qcc Qc Qt Qx` for `dim`>=4. If neither of the `QJ` or `Qt` options are supplied, the `Qt` option is passed to Qhull. The `Qt` option ensures all Delaunay regions are simplical (e.g., triangles in 2-d). See `../doc/html/qdelaun.html` for more details.
`full`	Return all information associated with triangulation as a list. This will return the triangulation (`tri`), list of voronoi Vertices (`voronoiVertices`) , vpoints (`points`) ,matrix list of the circum radii (`circumRadii`), matrix list of (`SimplicesPoints`) and a list of neighbours of each facet (`neighbours`). @return The voronoi vertics, circumRadii,trigulation Points and Delaunay triangulation @example x = c(30,70,20,50,40,70) y = c(35,80,70,50,60,20) p = matrix(c(x,y), ncol=2) voronoi(point =p)

Index

∗Topic **package**
    alphashape, 1

alphashape, 1
alphashape-package *(*alphashape*)*, 1

convex, 1, *2*

delaunay, 2, *3*

findSimplex, 3

gridGeneration, 3

inconvexhull, 4

nalphaShape, 4

voronoi, *4*, 5