Package ‘grainscape’

January 14, 2013

Type Package

Title Grains of connectivity and minimum planar graph modelling of landscape connectivity (Windows only)

Version 0.2

Date 2013-01-14

Author Paul Galpern, Andrew Fall, Micheline Manseau

Maintainer Paul Galpern <pgalpern@gmail.com>

Description Given a landscape resistance surface, creates grains of connectivity and minimum planar graph models that can be used to calculate effective distances for landscape connectivity at multiple scales. Distributed with SELES (Spatially Explicit Landscape Event Simulator) software. The package will currently run only on a Windows-based platform.

License GPL (>= 2)

Depends sp, raster, igraph

Suggests rgeos

OS_type windows

R topics documented:

`grainscape-package` *Grains of connectivity and minimum planar graph modelling of landscape connectivity using resistance surfaces*

Description

Given a landscape resistance surface, creates grains of connectivity and minimum planar graph models that can be used to calculate effective distances for landscape connectivity at multiple scales. Distributed with `SELES` (Spatially Explicit Landscape Event Simulator; Fall and Fall, 2001) software. The package will currently run only on a Windows-based platform.

Details

Landscape connectivity modelling to understand the movement and dispersal of organisms has been done using raster resistance surfaces and landscape graph methods. Grains of connectivity (GOC) models combine elements of both approaches to produce a continuous and scalable tool that can be applied in a variety of study systems. The purpose of this package is to implement grains of connectivity analyses. Routines accept raster-based resistance surfaces as input and return raster, vector and graph-based data structures to represent connectivity at multiple scales. Effective distances describing connectivity between geographic locations can be determined at multiple scales. Analyses of this sort can contribute to corridor identification, landscape genetics, as well as other connectivity assessments. Minimum planar graph (MPG; Fall et al., 2007) models of resource patches on landscapes can also be generated using the software.

SELES software (http://seles.info/ Fall and Fall, 2001) is distributed with the package, and is used to produce the MPG and perform a generalization of the Voronoi tessellation used in GOC models. Routines also depend on the `sp`, `raster`, `igraph` and optionally `rgeos` packages (Pebesma and Bivand, 2005, Csardi and Nepusz, 2006, Hijmans and van Etten, 2011, Bivand and Rundel, 2012).

An R vignette detailing the use of this package for landscape connectivity modelling is in preparation.

Please cite this package using the Galpern et al. (2012) reference below.

Author(s)

Paul Galpern, Andrew Fall, Micheline Manseau

Maintainer: Paul Galpern <pgalpern@gmail.com>

References

Primary references:

Fall, A., M.-J. Fortin, M. Manseau, D. O'Brien. Spatial graphs: Principles and applications for habitat connectivity. Ecosystems. 10:448:461.

Galpern, P., M. Manseau, P.J. Wilson. (2012) Grains of connectivity: analysis at multiple spatial scales in landscape genetics. Molecular Ecology. 21:3996-4009.

Supporting packages and software:

Bivand, R.S. and C. Rundel. (2012). rgeos: Interface to Geometry Engine - Open Source (GEOS). R package version 0.2-1.

Csardi, G. and T. Nepusz. (2006). The igraph software package for complex network research. InterJournal Complex Systems 1695.

Fall, A. and J. Fall. (2001). A domain-specific language for models of landscape dynamics. Ecological Modelling 141:1-18.

Hijmans, R.J. and J. van Etten. (2011). raster: Geographic analysis and modeling with raster data. R package version 1.9-58.

Pebesma, E.J. and R.S. Bivand. (2005). Classes and methods for spatial data in R. R News 5:http://cran.r-project.org/doc/Rnews/.

`gsGOC` *Produce a grains of connectivity model at multiple scales (patch-based or lattice GOC)*

Description

Given a `gsMPG` object produce a grains of connectivity (GOC) model at multiple scales (resistance thresholds) by scalar analysis. Patch-based or lattice GOC modelling can be done with this function.

Usage

```
gsGOC(gsMPG, nThresh = NULL, doThresh = NULL, weight = "lcpPerimWeight",
    sp = FALSE, verbose = 3)
```

Arguments

`gsMPG`	A `gsMPG` object produced by `gsMPG`. For lattice GOC `gsMPG` must be run with patch set as an integer value.
`nThresh`	Optional. An integer giving the number of thresholds (or scales) at which to create GOC models. Thresholds are selected to produce a maximum number of unique grains (i.e. models). `nThresh` thresholds are also approximately evenly spread between 0 and the threshold at which all patches or focal points on the landscape are connected. This is a simple way to get a representative subset of all possible GOC models. Provide either `nThresh` or `doThresh` not both.
`doThresh`	Optional. A vector giving the link thresholds at which to create GOC models. Use `link{gsThreshold}` to identify thresholds of interest. Provide either `nThresh` or `doThresh` not both.
`weight`	A string giving the link weight or attribute to use for threshold. `"lcpPerimWeight"` uses the accumulated resistance or least-cost path distance from the perimeters of patches as the link weight. `"eucPerimWeight"` use the Euclidean distance from the perimeters of patches as the link weight.

sp	Logical. If TRUE the rgeos package is used to create a vector of class SpatialPolygonsDataFrame describing the finest grain of connectivity. This is very useful for visualizing grains of connectivity models, especially for print purposes. Equally, using the maptools or rgdal packages these polygons can be exported as shapefiles for use in other GIS applications. But, please see details.
verbose	Set verbose=1 for no progress information to console.

Details

This function can take a long time to run when sp=TRUE. Time taken is dependent on the dimensions of the gsMPG$voronoi raster. Also, as of this release (May, 2012) there was still a memory leak in rgeos caused by its parent GEOS library. In extreme circumstances sp=TRUE may fail or cause a crash of the R process.

Value

A gsGOC object, consisting of a list of objects.

The main elements:
$voronoi is a raster describing the regions of proximity in resistance units around the focal patches or points (RasterLayer)
$voronoiSP is a vector representation of these regions of proximity (SpatialPolygons; if sp=TRUE)
$summary summarizes the grains of connectivity generated and their properties
$th is a list of length nThresh or length(doThresh) giving the GOC graph at each threshold.

Each element of $th contains a $goc object giving the GOC graph as class igraph. Vertex attributes describes qualities of each polygon including the coordinates of each polygon centroid, the area of these polygons, and the original patch IDs in the MPG that are included in each polygon. All areal measurements are given as raster cell counts. A variety of edge attributes are also given in the GOC graph. See gsGOCDistance for more information.

Note

Researchers should consider whether the use of a patch-based GOC or a lattice GOC model is appropriate based on the patch-dependency of the organism under study. Patch-based models make most sense when animals are restricted to, or dependent on, a resource patch. Lattice models can be used as a generalized and functional approach to scaling resistance surfaces.

See gsMPG for warning related to areal measurements.

Author(s)

Paul Galpern (<pgalpern@gmail.com>)

References

Fall, A., M.-J. Fortin, M. Manseau, D. O'Brien. (2007) Spatial graphs: Principles and applications for habitat connectivity. Ecosystems. 10:448:461

Galpern, P., M. Manseau, P.J. Wilson. (2012) Grains of connectivity: analysis at multiple spatial scales in landscape genetics. Molecular Ecology 21:3996-4009.

See Also

gsMPG, gsGOCVisualize, gsGOCDistance, gsGOCPoint

Examples

```
## Not run:

## Load raster landscape
tiny <- raster(system.file("extdata/tiny.asc", package="grainscape"))

## Create a resistance surface from a raster using an is-becomes reclassifyifyification
tinyCost <- reclassifyify(tiny, rcl=cbind(c(1, 2, 3, 4), c(1, 5, 10, 12)))

## Produce a patch-based MPG where patches are resistance features=1
tinyPatchMPG <- gsMPG(cost=tinyCost, patch=tinyCost==1)


## Extract a representative subset of 5 grains of connectivity
tinyPatchGOC <- gsGOC(tinyPatchMPG, nThresh=5)

## Examine the properties of the GOC graph of grain 3 of 5
print(tinyPatchGOC$th[[3]]$goc, vertex=TRUE, edge=TRUE)

## Extract specified grains of connectivity and produce a vector SpatialPolygons
## representation of the finest grain of connectivity (Threshold=0)
tinyPatchGOC <- gsGOC(tinyPatchMPG, doThresh=c(0, 20, 40), sp=TRUE)


## End(Not run)
```

---

gsGOCCorridor — *Visualize corridors between two points using a grains of connectivity (GOC) tessellation at a given scale in vector format*

Description

Given a series of GOC models built at different scales in a gsGOC object, visualize the corridor (or shortest path) between two points using one of the tessellations (i.e. scales) in these models. Visualization is exclusively in vector format. gsGOC must be run using the sp=TRUE option.

Usage

```
gsGOCCorridor(gsGOC, whichThresh, coords, doPlot=FALSE, weight="meanWeight")
```

Arguments

gsGOC	A gsGOC object created by gsGOC
whichThresh	Integer giving the index of the threshold to visualize.

coords	A two column matrix or a SpatialPoints object giving coordinates at the end points of the corridor
doPlot	Logical. If TRUE plots a vector visualization of the corridor at the given scale. For full control, manually produce plots using the outputs of this function.
weight	The GOC graph link weight to use in calculating the distance. Please see details in gsGOCDistance.

Value

A list object:

$voronoiSP vector representation of polygons in the tessellation (SpatialPolygonsDataFrame)
$linksSP vector representation of links in the grains of connectivity graph (SpatialLinesDataFrame)
$nodesSP vector representation of the nodes in the grains of connectivity graph (SpatialPoints)
$shortestLinksSP vector representation of the links in the shortest path between coordinates (SpatialLines)
$shortestNodesSP vector representation of the nodes in the shortest path between coordinates (SpatialPoints)
$corridorLength gives the length of the shortest path between coordinates in accumulated resistance units

Author(s)

Paul Galpern (<pgalpern@gmail.com>)

References

Fall, A., M.-J. Fortin, M. Manseau, D. O'Brien. (2007) Spatial graphs: Principles and applications for habitat connectivity. Ecosystems. 10:448:461

Galpern, P., M. Manseau, P.J. Wilson. (2012) Grains of connectivity: analysis at multiple spatial scales in landscape genetics. Molecular Ecology 21:3996-4009.

See Also

gsGOC gsGOCVisualize

Examples

```
## Not run:
## Load raster landscape
tiny <- raster(system.file("extdata/tiny.asc", package="grainscape"))

## Create a resistance surface from a raster using an is-becomes reclassification
tinyCost <- reclassify(tiny, rcl=cbind(c(1, 2, 3, 4), c(1, 5, 10, 12)))


## Produce a patch-based MPG where patches are resistance features=1
tinyPatchMPG <- gsMPG(cost=tinyCost, patch=tinyCost==1)
```

```
## Extract a representative subset of 5 grains of connectivity using sp=TRUE
tinyPatchGOC <- gsGOC(tinyPatchMPG, nThresh=5, sp=TRUE)

## Quick visualization of a corridor
corridorStartEnd <- rbind(c(10,10), c(90,90))
gsGOCCorridor(tinyPatchGOC, whichThresh=3, coords=corridorStartEnd, doPlot=TRUE)

## More control over a corridor visualization
tinyPatchCorridor <- gsGOCCorridor(tinyPatchGOC, whichThresh=3, coords=corridorStartEnd)
plot(tinyPatchCorridor$voronoiSP, col="lightgrey", border="white", lwd=2)
plot(tinyPatchCorridor$linksSP, col="darkred", lty="dashed", add=TRUE)
plot(tinyPatchCorridor$nodesSP, col="darkred", pch=21, bg="white", add=TRUE)
plot(tinyPatchCorridor$shortestLinksSP, col="darkred", lty="solid", lwd=2, add=TRUE)
plot(tinyPatchCorridor$shortestNodesSP, col="darkred", pch=21, bg="darkred", add=TRUE)
mtext(paste("Corridor shortest path length:",
            round(tinyPatchCorridor$corridorLength, 2),
            "resistance units"), side=1)


## End(Not run)
```

gsGOCDistance — *Find the grains of connectivity network distance*

Description

Given a gsGOC object find the shortest network distance between pairs of points using the GOC graph. This can be used as an effective distance for landscape connectivity assessments.

Usage

```
gsGOCDistance(gsGOC, coords, weight = "meanWeight")
```

Arguments

gsGOC	A gsGOC object produced by gsGOC
coords	A two column matrix or a SpatialPoints object giving the coordinates of points of interest
weight	The GOC graph link weight to use in calculating the distance. Please see details for explanation.

Details

GOC graphs with the following links weights can be used:

weight="meanWeight" links represent the mean weight of all links connecting MPG nodes in neighbouring polygons. It is recommended.
weight="maxWeight" links represent the maximum weight of all links connecting MPG nodes in neighbouring polygons.
weight="minWeight" links represent the minimum weight of all links connecting MPG nodes in neighbouring polygons.
weight="medianWeight" links represent the median weight of all links connecting MPG nodes in

neighbouring polygons.
weight="numEdgesWeight" links represent the number of links connecting MPG nodes in neighbouring polygons.
weight="eucCentroidWeight" links represent the Euclidean distances between the centroids of neighbouring polygons.

Value

A list object giving a distance matrix for each threshold in the gsGOC object. Distance matrices give the pairwise grains of connectivity network distances between sampling locations. Matrix indices correspond to rows in the coords matrix.

Author(s)

Paul Galpern (<pgalpern@gmail.com>)

References

Fall, A., M.-J. Fortin, M. Manseau, D. O'Brien. (2007) Spatial graphs: Principles and applications for habitat connectivity. Ecosystems. 10:448:461

Galpern, P., M. Manseau, P.J. Wilson. (2012) Grains of connectivity: analysis at multiple spatial scales in landscape genetics. Molecular Ecology 21:3996-4009.

See Also

gsGOC, gsGOCPoint

Examples

```
## Not run:

## Load raster landscape
tiny <- raster(system.file("extdata/tiny.asc", package="grainscape"))

## Create a resistance surface from a raster using an is-becomes reclassifyification
tinyCost <- reclassify(tiny, rcl=cbind(c(1, 2, 3, 4), c(1, 5, 10, 12)))

## Produce a patch-based MPG where patches are resistance features=1
tinyPatchMPG <- gsMPG(cost=tinyCost, patch=tinyCost==1)

## Extract a representative subset of 5 grains of connectivity
tinyPatchGOC <- gsGOC(tinyPatchMPG, nThresh=5)

## Three sets of coordinates in the study area
loc <- cbind(c(30, 60, 90), c(30, 60, 90))

## Find the GOC network distance matrices between these poitns
## for each of the 5 grains of connectivity
tinyDist <- gsGOCDistance(tinyPatchGOC, loc)

## End(Not run)
```

`gsGOCPoint` *Identify the polygons containing locations in grains of connectivity (GOC) tessellations*

Description

Given a `gsGOC` object identify the polygon containing a location at multiple scales.

Usage

```
gsGOCPoint(gsGOC, coords)
```

Arguments

`gsGOC`	A `gsGOC` object produced by `gsGOC`
`coords`	A two column matrix or a `SpatialPoints` object giving the coordinates of points of interest

Value

A list object with elements: `$pointPolygon` is a matrix with elements giving the id of the polygon from the `gsGOC`, where rows give points of interest and columns give thresholds
`$pointTotalPatchArea` is a matrix with elements giving the area of patches in a polygon (in cell counts), where rows give points of interest and columns give thresholds
`$pointTotalCoreArea` is the same for core area of patches
`$pointECS` gives the patch area (in cell counts) averaged for all points of interest (defined by O'Brien et al. 2006)
`$pointECSCore` is the same for the core area of patches

Note

See `gsMPG` for warning related to areal measurements.

Author(s)

Paul Galpern (<`pgalpern@gmail.com`>)

References

Fall, A., M.-J. Fortin, M. Manseau, D. O'Brien. (2007) Spatial graphs: Principles and applications for habitat connectivity. Ecosystems. 10:448:461

Galpern, P., M. Manseau, P.J. Wilson. (2012) Grains of connectivity: analysis at multiple spatial scales in landscape genetics. Molecular Ecology 21:3996-4009.

O'Brien, D., M. Manseau, A. Fall, and M.-J. Fortin. (2006) Testing the importance of spatial configuration of winter habitat for woodland caribou: An application of graph theory. Biological Conservation 130:70-83.

See Also

gsGOC, gsGOCDistance

Examples

```
## Not run:
## Load raster landscape
tiny <- raster(system.file("extdata/tiny.asc", package="grainscape"))

## Create a resistance surface from a raster using an is-becomes reclassification
tinyCost <- reclassify(tiny, rcl=cbind(c(1, 2, 3, 4), c(1, 5, 10, 12)))

## Produce a patch-based MPG where patches are resistance features=1
tinyPatchMPG <- gsMPG(cost=tinyCost, patch=tinyCost==1)

## Extract a representative subset of 5 grains of connectivity
tinyPatchGOC <- gsGOC(tinyPatchMPG, nThresh=5)

## Three sets of coordinates in the study area
loc <- cbind(c(30, 60, 90), c(30, 60, 90))

## Find the GOC polygon containing these three locations
## for each of the 5 grains of connectivity
tinyPts <- gsGOCPoint(tinyPatchGOC, loc)

## End(Not run)
```

`gsGOCVisualize` *Visualize grains of connectivity (GOC) tessellations at a given scale*

Description

Given a series of GOC models built at different scales in a `gsGOC` object, visualize one the tessellations (i.e. scales) in these models. Visualization is by default in raster format. Vector based visualization is also possible.

Usage

```
gsGOCVisualize(gsGOC, whichThresh, sp = FALSE, doPlot=FALSE)
```

Arguments

`gsGOC`	A `gsGOC` object created by `gsGOC`
`whichThresh`	Integer giving the index of the threshold to visualize.
`sp`	Logical. If `TRUE` then produce a `SpatialPolygonsDataFrame` representation of the selected threshold. Requires also running `gsGOC` with `sp=TRUE`, and that the `rgeos` package is installed.
`doPlot`	Logical. If `TRUE` plots a raster (or vector if `sp=TRUE`) of the Voronoi tessellation at `whichThresh` for quick visualizations. For full control, manually produce plots using the `$voronoi` or `$voronoiSP` objects created by this function.

Value

A list object:

`$summary` giving the properties of the visualized scale of the GOC model
`$voronoi` giving the tessellation (`RasterLayer`)
`$centroids` the centroids of the polygons in the tessellation (`SpatialPoints`)
`$voronoiSP` vector representation of polygons in the tessellation (`SpatialPolygonsDataFrame`; if `sp=TRUE`)

Author(s)

Paul Galpern (<pgalpern@gmail.com>)

References

Fall, A., M.-J. Fortin, M. Manseau, D. O'Brien. (2007) Spatial graphs: Principles and applications for habitat connectivity. Ecosystems. 10:448:461

Galpern, P., M. Manseau, P.J. Wilson. (2012) Grains of connectivity: analysis at multiple spatial scales in landscape genetics. Molecular Ecology 21:3996-4009.

See Also

gsGOC

Examples

```
## Not run:
## Load raster landscape
tiny <- raster(system.file("extdata/tiny.asc", package="grainscape"))

## Create a resistance surface from a raster using an is-becomes reclassification
tinyCost <- reclassify(tiny, rcl=cbind(c(1, 2, 3, 4), c(1, 5, 10, 12)))


## Produce a patch-based MPG where patches are resistance features=1
tinyPatchMPG <- gsMPG(cost=tinyCost, patch=tinyCost==1)

## Extract a representative subset of 5 grains of connectivity
tinyPatchGOC <- gsGOC(tinyPatchMPG, nThresh=5)

## Very quick visualization at the finest scale/grain/threshold
## Producing plot on the default graphics device
gsGOCVisualize(tinyPatchGOC, whichThresh=1, doPlot=TRUE)

## Visualize the model at the finest scale/grain/threshold
## Manual control of plotting
plot(gsGOCVisualize(tinyPatchGOC, whichThresh=1)$voronoi,
   col=sample(rainbow(100)), legend=FALSE, main="Threshold 1")

## Extract a representative subset of 5 grains of connectivity for vector visualization
tinyPatchGOC <- gsGOC(tinyPatchMPG, nThresh=5, sp=TRUE)
```

```
## Visualize the model at a selected scale/grain/threshold using vector polygons
plot(tinyPatchMPG$patchId, col="grey", legend=FALSE)
plot(gsGOCVisualize(tinyPatchGOC, whichThresh=3, sp=TRUE)$voronoiSP,
  add=TRUE, lwd=2)


## End(Not run)
```

`gsGraphDataFrame` *Produce a data.frame containing the structure and associated attributes for a gsMPG, gsGOC, or igraph object*

Description

Given a gsMPG, gsGOC, or any igraph object produce a data.frame containing the node (vertex) and link (edge) structure as well as the associated attributes for these. This provides an easy way to create data tables describing graphs, particularly helpful for users unfamiliar with the structure of igraph objects

Usage

```
gsGraphDataFrame(gsObj)
```

Arguments

`gsObj`	A `gsMPG`, `gsGOC`, or `igraph` object

Value

A list object:

`$v` giving node (vertex) names and associated attributes
`$e` giving link (edge) lists and associated attributes

Please see `gsMPG` and `gsGOC` for details about the attributes.

For `gsGOC` objects which typically contain multiple thresholds, an enumerated list of the same length as the number of thresholds is returned each containing `$v` and `$e` elements.

Author(s)

Paul Galpern (<pgalpern@gmail.com>)

References

Fall, A., M.-J. Fortin, M. Manseau, D. O'Brien. (2007) Spatial graphs: Principles and applications for habitat connectivity. Ecosystems. 10:448:461

Galpern, P., M. Manseau, P.J. Wilson. (2012) Grains of connectivity: analysis at multiple spatial scales in landscape genetics. Molecular Ecology 21:3996-4009.

See Also

gsMPG, gsGOC

Examples

```
## Not run:
## Load raster landscape
tiny <- raster(system.file("extdata/tiny.asc", package="grainscape"))

## Create a resistance surface from a raster using an is-becomes reclassification
tinyCost <- reclassify(tiny, rcl=cbind(c(1, 2, 3, 4), c(1, 5, 10, 12)))


## Produce a patch-based MPG where patches are resistance features=1
tinyPatchMPG <- gsMPG(cost=tinyCost, patch=tinyCost==1)

## Extract a representative subset of 5 grains of connectivity
tinyPatchGOC <- gsGOC(tinyPatchMPG, nThresh=5)

## Create a data.frame with the structure and attributes of a gsMPG object
tinyPatchMPG_df <- gsGraphDataFrame(tinyPatchMPG)

## Create a data.frame with the structure and attributes of a gsGOC object
tinyPatchGOC_df <- gsGraphDataFrame(tinyPatchGOC)

## Create a data.frame with the structure and attributes of any igraph object
gsGraphDataFrame(tinyPatchGOC$th[[1]]$goc)


## End(Not run)
```

---

gsMPG *Extract a minimum planar graph (MPG) model from a landscape resistance surface (using SELES)*

Description

This function is used to extract a minimum planar graph (MPG) and it is also the first step in grains of connectivity (GOC) modelling. Both patch-based and lattice MPGs can be extracted. The function calls `SELES`, a Windows-based executable distributed with the package, to extract the graph.

Usage

```
gsMPG(cost, patch, sa = NULL, outputFolder = NULL, filterPatch = NULL,
    spreadFactor = 0, selesPath = system.file("SELES", package = "grainscape"))
```

Arguments

cost	A raster of class RasterLayer giving a landscape resistance surface, where the values of each raster cell are proportional to the resistance to movement, dispersal, or gene flow for an organism in the landscape feature they represent. Missing values NA are acceptable (but see below). Negative values are not. To extract an MPG with Euclidean links (i.e. and not least-cost path links) set `cost[] <- 1`.
patch	A raster of class RasterLayer for a patch-based analysis OR an integer for a lattice analysis. If a raster is given it must be of the same extent, origin and projection as cost and be binary, without missing values, where patches=1 and non-patches=0. For lattice analyses, an integer gives the spacing in raster cells between focal points in the lattice.
sa	Optional. A raster of class RasterLayer of the same extent, origin and projection as cost indicating the study area (i.e. cells on the landscape to include in the analysis). If not supplied sa is the full extent of cost. To mask out areas of the landscape to exclude from analysis (e.g. at the edges of a map), supply a binary raster where included cells=1 and excluded cells=0.
outputFolder	Optional. If not supplied this function creates files for use by SELES in a temporary folder placed in the R working directory that is deleted following successful execution. Another location may be specified instead. If supplied the location is not deleted after the analysis completes. This can be useful for debugging purposes.
filterPatch	Optional. Remove patches from the analysis that are smaller than a given number of cells.
spreadFactor	Optional. Fine-grained control over the accuracy of Voronoi polygons. To reduce accuracy and increase speed, set this as spreadFactor=10 or spreadFactor=100.
selesPath	Optional. The location of the SELES installation. By default this is the folder in the package installation.

Details

Use this function to create a minimum planar graph (MPG) that can be further analyzed using igraph routines.

It is also the first step in grains of connectivity (GOC) modelling.

Value

A gsMPG object, consisting of a list of objects.

The main elements:
$mpg is the minimum planar graph as class igraph
$patchId is the input patch raster with patch cells assigned to their id (RasterLayer)
$voronoi is the Voronoi tessellation of the patches and resistance surface (RasterLayer)
$lcpPerimWeight gives the paths of the links between patches and their accumulated costs (RasterLayer)
$lcpLinkId gives the paths of the links between patches and their id (RasterLayer)

`$lcpPerimType` gives the paths of the links between patches and their type (`RasterLayer`; see notes)
`$mpgPlot` provides a quick way of visualizing the mpg (`RasterLayer`)

The `$mpg` has useful vertex and edge attributes. Vertex attributes give attributes of patches including patch area, the area of patch edges, the core area of each patch, and the coordinates of the patch centroid. All areal measurements are given as raster cell counts. Edge attributes give attributes of the graph links including link weights giving accumulated resistance/least-cost path distance, Euclidean distance, and the start and end coordinates of each link.

Note

SELES has been compiled for Windows. Therefore use of `gsMPG` is limited to Windows-based platforms.

Researchers should consider whether the use of a patch-based MPG or a lattice MPG model is appropriate based on the patch-dependency of the organism under study. Patch-based models make most sense when animals are restricted to, or dependent on, a resource patch. Lattice models can be used as a generalized and functional approach to scaling resistance surfaces.

Four types of links are identified in the MPG (1=Nearest neighbour; 2=Minimum spanning tree; 3=Gabriel; 4=Delaunay;)

Areal measurements are given as raster cell counts. If the raster projection is one where cell sizes are approximately constant in area (e.g. UTM), or the raster covers a relatively small geographic extent (e.g. < 1000 km in dimension) areal measurements will often be adequate. Reprojection of rasters should be considered to minimize these effects in other cases (see `projectRaster`).

Author(s)

Paul Galpern (<pgalpern@gmail.com>), Andrew Fall

References

Fall, A., M.-J. Fortin, M. Manseau, D. O'Brien. (2007) Spatial graphs: Principles and applications for habitat connectivity. Ecosystems. 10:448:461

Galpern, P., M. Manseau, P.J. Wilson. (2012) Grains of connectivity: analysis at multiple spatial scales in landscape genetics. Molecular Ecology 21:3996-4009.

See Also

`gsGOC`, `gsThreshold`

Examples

```
## Not run:

## Load raster landscape
tiny <- raster(system.file("extdata/tiny.asc", package="grainscape"))
```

```
## Create a resistance surface from a raster using an is-becomes reclassifyification
tinyCost <- reclassify(tiny, rcl=cbind(c(1, 2, 3, 4), c(1, 5, 10, 12)))

## Produce a patch-based MPG where patches are resistance features=1
tinyPatchMPG <- gsMPG(cost=tinyCost, patch=tinyCost==1)

## Explore the graph structure and link attributes
print(tinyPatchMPG$mpg, edge=TRUE)

## Explore the patch/node attributes
print(tinyPatchMPG$mpg, vertex=TRUE)

## Find the mean patch area (see igraph manual for use of V() and E())
mean(V(tinyPatchMPG$mpg)$patchArea)

## Quick visualization of the MPG
plot(tinyPatchMPG$mpgPlot, col=c("grey", "black"), legend=FALSE)

## Visualize the minimum spanning tree of the MPG
plot(tinyPatchMPG$patchId, col="black", legend=FALSE)
plot(tinyPatchMPG$lcpPerimType %in% c(1,2), add=TRUE, legend=FALSE, col=c(NA, "grey"))

## Additional graph extraction scenarios
## Produce a lattice MPG where focal points are spaced 10 cells apart
tinyLatticeMPG <- gsMPG(cost=tinyCost, patch=10)

## Produce a patch-based MPG with a study area consisting of half of the map
tinySa <- tinyCost
tinySa[] <- 1
tinySa[1:5000] <- 0
tinyPatchMPG <- gsMPG(cost=tinyCost, patch=tinyCost==1, sa=tinySa)


## End(Not run)
```

---

gsThreshold *Produce a minimum planar graph (MPG) at multiple scales*

Description

This function performs a scalar analysis of a minimum planar graph (MPG) by building the graph at a series of link thresholds. As the threshold value increases more nodes in the graph become connected, forming increasingly fewer components, until the graph becomes connected (e.g. Brooks, 2003). N.B. Grains of connectivity (GOC) done by gsGOC is also a scalar analysis using Voronoi tessellations rather than patches (see Galpern et al., 2012).

Usage

```
gsThreshold(gsMPG, weight = "lcpPerimWeight", nThresh = NULL, doThresh = NULL)
```

Arguments

gsMPG	A gsMPG object produced by gsMPG.
weight	A string giving the link weight or attribute to use for threshold. "lcpPerimWeight" uses the accumulated resistance or least-cost path distance from the perimeters of patches as the link weight. "eucPerimWeight" use the Euclidean distance from the perimeters of patches as the link weight.
nThresh	Optional. An integer giving the number of thresholds (or scales) at which to create GOC models. Thresholds are selected to produce a maximum number of unique grains (i.e. models). nThresh thresholds are also approximately evenly spread between 0 and the threshold at which all patches or focal points on the landscape are connected. This is a simple way to get a representative subset of all possible GOC models. Provide either nThresh or doThresh not both.
doThresh	Optional. A vector giving the link thresholds at which to create GOC models. Use link{gsThreshold} to identify thresholds of interest. Provide either nThresh or doThresh not both.

Value

A list object with the following elements:
$summary summarizes the thresholded graphs generated and their properties
$th is a list of length nThresh or length(doThresh) giving the thresholded graph (class igraph) at each threshold.

Note

See gsMPG for warning related to areal measurements.

Author(s)

Paul Galpern (<pgalpern@gmail.com>)

References

Brooks, C.P. (2003) A scalar analysis of landscape connectivity. Oikos 102:433-439.

Fall, A., M.-J. Fortin, M. Manseau, D. O'Brien. (2007) Spatial graphs: Principles and applications for habitat connectivity. Ecosystems. 10:448:461

Galpern, P., M. Manseau, P.J. Wilson. (2012) Grains of connectivity: analysis at multiple spatial scales in landscape genetics. Molecular Ecology 21:3996-4009.

See Also

gsMPG

Examples

```
## Not run:
# Load raster landscape
tiny <- raster(system.file("extdata/tiny.asc", package="grainscape"))

## Create a resistance surface from a raster using an is-becomes reclassification
tinyCost <- reclassify(tiny, rcl=cbind(c(1, 2, 3, 4), c(1, 5, 10, 12)))

## Produce a patch-based MPG where patches are resistance features=1
tinyPatchMPG <- gsMPG(cost=tinyCost, patch=tinyCost==1)

## Threshold this graph at a representative subset of 10 thresholds
tinyThresh <- gsThreshold(tinyPatchMPG, nThresh=10)

## Examine the properties of one of these threshold graphs
print(tinyThresh$th[[7]], vertex=TRUE, edge=TRUE)

## End(Not run)
```

Index