Package 'archeofrag'

November 27, 2020

Type Package

Title Refitting and Stratigraphic Analysis in Archeology

Version 0.6.0

Date 2020-11-24

Author Sebastien Plutniak [aut, cre] (<https://orcid.org/0000-0002-6674-3806>)

Maintainer Sebastien Plutniak <sebastien.plutniak@posteo.net>

Description Methods for the stratigraphic analysis of fragmented objects in archeology using ``refitting'' relationships between fragments scattered in stratigraphic layers. Graphs and graph theory are used to model archaeological observations. The package is mainly based on the 'igraph' package for graph analysis. Functions enable to 1) create, manipulate, and simulate fragmentation graphs, 2) measure the cohesion and admixture of stratigraphic layers, and 3) characterise the topology of a specific set of refitting relationships. An empirical dataset is also provided as an example.

License GPL-2

Depends RBGL, igraph

Imports graphics, stats, grDevices, methods

Suggests knitr, rmarkdown

VignetteBuilder knitr

URL https://github.com/sebastien-plutniak/archeofrag

BugReports https://github.com/sebastien-plutniak/archeofrag/issues

NeedsCompilation no

Repository CRAN

Date/Publication 2020-11-27 11:40:10 UTC

R topics documented:

`archeofrag-package` *Archeofrag: tools to analyse the fragmentation of archeological objects*

Description

The Archeofrag package implements a comprehensive method to analyse fragmentation processes in archeology, based on the fragments of archeological objects scattered in stratigraphic layers and connected by physical "refittings" relationships. Graphs are used to model these situations. The package is mainly based on the igraph package for graph analysis. It offers functions to manipulate fragment datasets, to measure the properties of the graphs, and to generate artificial cases to compare with archaeological observations. An empirical dataset is also provided as an example.

Details

Package:	archeofrag
Type:	Package
Version:	0.5.4
Date:	2020-09-30
License:	GPL-2

Author(s)

Sebastien Plutniak Maintainer: Sebastien Plutniak <sebastien.plutniak at posteo.net>

See Also

`igraph` `RBGL`

`frag.cycles` *Count the k-cycles in a graph, for cycles =< a given k*

Description

Count the k-cycles in a graph, for cycles =< a given k

Usage

```
frag.cycles(graph, kmax, max.cycles.only=FALSE)
```

Arguments

`graph`	An `igraph` object, must be an undirected graph.
`kmax`	Maximal length of the cycles to detect.
`max.cycles.only`	Logical. If TRUE, the fragments are only reported as parts of their longer cycle.

Details

A cycle can be part of larger cycles: if `max.cycles.only` all the cycles are reported but, if this parameter is True only the larger cycles are reported. A warning reminds that for cycles k > 4 the fragments of a cycle are not necessarily all connected to each other (due to its location in the original object, a fragment can only be connected to a limited number of adjacent fragments).

Value

A data frame with the number of k-cycles for each k values in [3;k].

Author(s)

Sebastien Plutniak <sebastien.plutniak at posteo.net>

See Also

`subgraph_isomorphisms`

Examples

```
g <- frag.simul.process(n.components=20, vertices=50, disturbance=.15)
frag.cycles(g, kmax=4, max.cycles.only=FALSE)
frag.cycles(g, kmax=4, max.cycles.only=TRUE)
```

`frag.diameters` *Diameter distribution for unconnected graph*

Description

Returns the distribution of the diameter values of an unconnected graph.

Usage

```
frag.diameters(graph, cumulative = FALSE)
```

Arguments

`graph`	An `igraph` object.
`cumulative`	Logical. If TRUE the cumulative relative frequency of the diameters is reported.

Details

`frag.diameters` wraps the `igraph` `diameter` function. For graphs representing the fragmentation of archeological objects, the diameter of each component of the graph (i.e. archeological objects) can be interpreted: as a measure of the intensity of the fragmentation (when all the fragments of the initial object are known); as a measure of the scattering of the fragments (when not all the fragments are known);

Value

A numeric vector of the length equal to the maximum diameter value found. The first element is the frequency of the diameter values = 1, the second element is the frequency of diameter values = 2, etc. If `cumulative` is True, the cumulative density is returned.

Author(s)

Sebastien Plutniak <sebastien.plutniak at posteo.net>

See Also

`diameter`

Examples

```
g <- frag.simul.process(n.components=20, vertices=50, disturbance=.15)
frag.diameters(g)
frag.diameters(g, cumulative=TRUE)
```

`frag.edges.weighting` *Weighting of an archeofrag graph*

Description

Weighting of the edges of an archeofrag fragmentation graph.

Usage

```
frag.edges.weighting(graph, layer.attr)
```

Arguments

`graph`	An undirected `igraph` object.
`layer.attr`	Character. The name of the vertice attribute giving the layer of each fragment.

Details

Considering two layers, internal (within a layer) and external relations (between the two layers) are distinguished. The weights of their edges is computed in two different ways. Three subgraphs are first generated, one for each layer and one only for the external relations.

The weight of an intra-layer edge (E) is equal to the sum of the degrees (d) of the vertices (i and j) it connects:

W_intra(E_ij)= di + dj

For an inter-layers edge, the same calculation is done but with a modifier to account for the balance of information available for each layer:

W_inter(E_ij)= (di + dj) * (1 + (1 / (1 + sum(W_intra 1)/sum(W_intra 2))))

with `W_intra 1` the internal weights within layer 1 and `W_intra 2` the internal weights within layer 2.

An error message is displayed if the vertices "layer" attribute has more than two layers.

Value

The input graph, with an additional "weight" edge attribute.

Author(s)

Sebastien Plutniak <sebastien.plutniak at posteo.net>

See Also

`transitivity`

Examples

```
g <- frag.simul.process(n.components=20, vertices=50, disturbance=.15)
frag.edges.weighting(g , "layer")
```

`frag.get.layers` *Extracts the subgraph of each selected stratigraphic layers.*

Description

Extracts the subgraph of each selected stratigraphic layers.

Usage

```
frag.get.layers(graph, layer.attr, sel.layers)
```

Arguments

`graph`	An undirected `igraph` object.
`layer.attr`	Character. The name of the vertices attribute giving the layer of each fragment.
`sel.layers`	Character. The identifier(s) of the stratigraphic layers to retrieve.

Details

This function is only a convenient function to extract the subgraphs of selected stratigraphic layers. A graph is created for each layer in the vertex attribute given by the `layer.attr` argument.

Value

A list with a graph for each selected stratigraphic layer.

Author(s)

Sebastien Plutniak <sebastien.plutniak at posteo.net>

Examples

```
g <- frag.simul.process(n.components=20, vertices=50, disturbance = .15)
V(g)$layers <- c(rep("layer1", 20), rep("layer2", 20), rep("layer3", 10))
frag.get.layers(g, layer.attr="layers", sel.layers=c("layer1", "layer2"))
```

`frag.get.layers.pair` *Extracts the subraph corresponding to a pair of stratigraphic layers.*

Description

Extracts the subraph corresponding to a pair of stratigraphic layers.

Usage

```
frag.get.layers.pair(graph, layer.attr, sel.layers, size.mini=2,
                     mixed.components.only=FALSE)
```

Arguments

`graph`	An `igraph` object.
`layer.attr`	Character. The name of the vertices attribute giving the layer of each fragment.
`sel.layers`	A numeric vector of length 2 with the name of the stratigraphic layer selected for extraction.
`size.mini`	A minimal number of vertices for the components to include in the resulting graph.
`mixed.components.only`	Logical. If TRUE, only the components with fragments from the two selected layers are returned. If FALSE, all the components of the two layers are extracted.

Details

The default setting of the `mixed.components.only` argument is FALSE, for convenience for other measurements.

Value

An undirected graph object.

Author(s)

Sebastien Plutniak <sebastien.plutniak at posteo.net>

Examples

```
g <- frag.simul.process(n.components=20, vertices=50, disturbance=.15)
V(g)$layers <- c(rep("layer1", 20), rep("layer2", 20), rep("layer3", 10))

frag.get.layers.pair(g, "layers", sel.layers=c("layer2","layer3"),
                     size.mini=2, mixed.components.only=FALSE)
frag.get.layers.pair(g, "layers", sel.layers=c("layer2","layer3"),
                     size.mini=2, mixed.components.only=TRUE)
```

`frag.get.parameters` *Returns a series of descriptive statistics for a fragmentation graph*

Description

Returns a series of descriptive statistics for a fragmentation graph.

Usage

```
frag.get.parameters(graph, layer.attr)
```

Arguments

`graph`	An `igraph` undirected graph.
`layer.attr`	Character. The name of the vertices attribute giving the layer of each fragment.

Details

This function is a convenient function to get general information about a fragmentation graph. It is particularly handy for setting the parameters of the `frag.simul.process` function. It returns the number of components, vertices, and edges, the balance (proportion of fragments in the smaller layer), components balance (proportion of components in the poorest layer), the disturbance, the aggregation factor, and if the graph is planar or not. The aggregation factor reflects the diversity of the components edge count. The factor is calculated by: 1 - 1/(1 + sd(edge counts of the components)).

Value

A list with the values for the parameters.

Author(s)

Sebastien Plutniak <sebastien.plutniak at posteo.net>

See Also

frag.get.layers.pair, frag.simul.process

Examples

```
g <- frag.simul.process(n.components=20, vertices=50)
frag.get.parameters(g, "layer")
```

frag.graph.plot *Plot a fragmentation graph*

Description

A function to plot the graph made by the archeofrag package.

Usage

```
frag.graph.plot(graph, layer.attr, ...)
```

Arguments

graph	An igraph undirected object with a "frag_type" attribute.
layer.attr	Character. The name of the vertices attribute giving the layer of each fragment.
...	Optional arguments sent to plot.igraph.

Details

This function is a wrapper for the plot.igraph method for igraph objects, using the fruchterman-reingold algorithm. The layout of the graph is modified in function of the value of the "frag_type" graph attribute: for "connection and similarity relations" value, igraph' component_wise layout modifier is used.

Value

Returns NULL and plot the graph.

Author(s)

Sebastien Plutniak <sebastien.plutniak at posteo.net>

See Also

plot.igraph, component_wise

Examples

```
g <- frag.simul.process(n.components=20, vertices=50, disturbance=.15)
frag.graph.plot(g, "layer")
```

`frag.layers.admixture` *Admixture of two stratigraphic layers*

Description

Evaluate how reliable is the distinction between two layers (or any other kind of archaeological spatial unit).

Usage

```
frag.layers.admixture(graph, layer.attr)
```

Arguments

`graph`	An undirected `igraph` object.
`layer.attr`	Character. The name of the vertex attribute giving the layer of each fragment.

Details

Given two layers (or other spatial unit) containing connected fragments, this function evaluate how reliable is their distinction, using the formula:

`1 -cohesion(layer 1) -cohesion(layer 2)`

So, in this method, the admixture of two layers is equal to the cohesion of an arbitrary third layers defined by the fragments and connection at the intersection of the two layers. This function must be used after weighting the edges with `frag.edges.weighting`. Results range in [0;1] with 0 for completely independent layers and values towards 1 for mixed layers.

An error message is displayed if the vertex "layer" attribute has more than two layers.

Value

A numerical value reflecting the admixture of the two layers.

Author(s)

Sebastien Plutniak <sebastien.plutniak at posteo.net>

See Also

`frag.edges.weighting`, `frag.simul.process`, `boyerMyrvoldPlanarityTest`

Examples

```
g <- frag.simul.process(n.components=20, vertices=50, disturbance = .15)
g <- frag.edges.weighting(g, layer.attr="layer")
frag.layers.admixture(g, "layer")
```

`frag.layers.cohesion` *Cohesion measure for two layers*

Description

Returns a cohesion value for each stratigraphic layer. Must be used after weighting the edges with `frag.edges.weighting`.

Usage

```
frag.layers.cohesion(graph, layer.attr)
```

Arguments

`graph`	An undirected `igraph` object.
`layer.attr`	Character. The name of the vertices attribute giving the layer of each fragment.

Details

This function returns a vector with a cohesion value for each layer.

The cohesion value for a given layer is given by:

`sum(W_inLayer) / (sum(w_in) + sum(W_ex))`

with W_inLayer for the weights of the internal edges of the layer under consideration, W_in for the weights of all the internal edges, and W_ex for the weights of the external edges.

The measure integrates the balance between the information about each layer. Results range in [0;1] with 0 for two layers only with inter-layers connection relationships, and 1 if there is no inter-layers relationships and important inequal information about the two layers.

An error message is displayed if the vertice "layer" attribute has more or less than two layers.

Value

A numeric vector with the cohesion values ([0;1]) for each layer of the graph.

Author(s)

Sebastien Plutniak <sebastien.plutniak at posteo.net>

See Also

`frag.edges.weighting`

Examples

```
g <- frag.simul.process(n.components=20, vertices=50, disturbance=.1)
g <- frag.edges.weighting(g, layer.attr="layer")
cohesion.values <- frag.layers.cohesion(g, layer.attr="layer")
```

`Frag.object-class` *Class* "Frag.object"

Description

A class for archaeological "fragmentation" dataset. This class construction aims to ensure that the data have been properly built before performing the next steps of the analysis. A convenient constructor function, make_frag_object, is provided.

Objects from the Class

Objects can be created by calls of the form:

`make_frag_object(cr,sr,fragments)`.

Slots

`cr.df`: Object of class "matrix" ("data.frame" are allowed and automatically converted)

`sr.df`: Object of class "matrix" ("data.frame" are allowed and automatically converted)

`fragments.df`: Object of class "data.frame"

`frag_type`: Object of class "character"

Methods

make_cr_graph `signature(object = "Frag.object")`:

Makes an undirected graph representing the "connection" relations between archaeological fragments. Cf. make_cr_graph

make_crsr_graph `signature(object = "Frag.object")`:

Makes an undirected graph combining the "connection" and "similarity" relations between archaeological fragments. Cf. make_crsr_graph

show `signature(object = "Frag.object")`:

show method for Frag.object

Author(s)

Sebastien Plutniak <sebastien.plutniak at posteo.net>

Examples

```
showClass("Frag.object")
```

`frag.path.lengths` *Path length distribution for unconnected graph*

Description

Path length distribution for unconnected graph

Usage

```
frag.path.lengths(graph, cumulative=FALSE)
```

Arguments

`graph`	An `igraph` object.
`cumulative`	Logical. If TRUE, the cumulative relative frequency of the path lengths is returned.

Details

This function is a wrapper of `igraph` `distance_table` returning the frequency of path lengths in undirected and unconnected graphs. In the context of archaeological fragmentation analysis, path lengths are interpreted to characterise the properties of fragmentation within a layer.

Value

A numeric vector having the same length than the maximum path length founded. The first element of the vector is the frequency of the paths of length 1, the second element is the frequency of the paths of length 2, etc.

Author(s)

Sebastien Plutniak <sebastien.plutniak at posteo.net>

See Also

`distance_table`

Examples

```
g <- frag.simul.process(n.components=20, vertices=50, disturbance = .15)
frag.path.lengths(g)
frag.path.lengths(g, cumulative=TRUE)
```

`frag.relations.by.layers` *Summary of the connection relationships between fragments within and between stratigraphic layers.*

Description

Return a matrix with the number of relations within and between stratigraphic layers.

Usage

```
frag.relations.by.layers(graph, layer.attr)
```

Arguments

`graph`	An `igraph` object.
`layer.attr`	Character. The name of the vertices attribute giving the layer of each fragment.

Details

This function is a handy method to summarize the distribution of the relations within and between stratigraphic layers.

Value

A symmetrical matrix with the number of relations within and between the layers.

Author(s)

Sebastien Plutniak <sebastien.plutniak at posteo.net>

Examples

```
g <- frag.simul.process(n.components=20, vertices=50, disturbance=.15)
frag.relations.by.layers(g, "layer")
```

`frag.simul.process` *Simulate the fragmentation of archaeological objects scattered in two stratigraphic layers.*

Description

Simulate the fragmentation of archaeological objects scattered in two stratigraphic layers.

Usage

```
frag.simul.process(initial.layers=2, n.components, vertices=Inf,
                   edges=Inf, balance=.5, components.balance=.5,
                   disturbance=0, aggreg.factor=0, planar=TRUE)
```

Arguments

`initial.layers`	Integer (1 or 2). Number of hypothetical stratigraphic layers to use as initial condition.
`n.components`	Integer. Number of objets to fragment (connected components).
`vertices`	Integer. Number of fragments (vertices).
`edges`	Integer. Number of connection relationships between fragments (edges).
`balance`	Numeric]0;1[. Proportion of fragments in the first layer.
`components.balance`	Numeric]0;1[. Proportion of components in the first layer (used only when initial.layers=2).
`disturbance`	Numeric [0;1]. Proportion of fragments to randomly move from a layer to the other.
`aggreg.factor`	Numeric [0;1]. Higher values increase the likelihood that the biggest components are selected when adding fragments or connections.
`planar`	Logical. If TRUE, generates a planar graph.

Details

This function simulates the fragmentation of archeological objects within and between two adjacent stratigraphic layers. Fragments are represented by vertices and the "connection" relationships ("refittings") between them are represented by edges.

Some parameters are optional or depend on other parameters (messages are displayed accordingly). Namely, if two `initial.layers` are set, then only one of the `vertices` and `edges` parameters can be used. Using only one layer as initial condition enables to constraint the graph with both vertices and edges number. The `disturbance` determines the proportion of fragments to "move" from a layer to the other one. Consequently, it generates inter-layers relationships. Note that the `balance` parameter determines the proportion of fragments in the first layer before the application of the disturbance process.

Note that the simulator is faster with 2 `initial.layers` and when not using the planarity constraint.

Value

An igraph object with a "frag_type" graph attribute (with the value "cr", for "connection relationship") and three vertices attributes: "name" (vertices identifiers), "layer" (with "1" and "2" values), and "object.id" (components identifiers).

Author(s)

Sebastien Plutniak <sebastien.plutniak at posteo.net>

See Also

boyerMyrvoldPlanarityTest

Examples

```
frag.simul.process(n.components=20, vertices=50, disturbance=.15)

g <- frag.simul.process(initial.layers=1,
                            n.components=20,
                            vertices=50,
                            edges=40,
                            balance=.5,
                            components.balance=.5,
                            disturbance=.1,
                            planar=TRUE)
plot(g, vertex.color=factor(V(g)$layer),
     vertex.size=4, vertex.label=NA)
```

LiangAbu *Dataset: Archeological relationships between pottery fragments in Liang Abu*

Description

Liang Abu is an archaeological site in East Kalimantan, Indonesia. This data set describes the relations between pottery fragments founded during the excavations (2009-2012). Two types of relations are defined.

- A connection relation refers to a physical connection between two fragments which were part of a same object.
- A similarity relation between fragments is defined if there is an acceptable likelihood that those fragments were part of a same object.

The data set is composed of three tables, `cr.df`, `sr.df`, `fragments.info`.

- `cr.df`: "connection" relations between fragments.
- `sr.df`: "similarity" relations between fragments.
- `fragments.info`: contextual informations concerning each fragment.

Usage

```
data(LiangAbu)
```

Format

- `cr.df` is a 56x2 matrix. Each line describes a connection relationship between two fragments. There respective unique identifiers are given in column "frg_id1" and in column "frg_id2".
- `sr.df` is a 147x2 matrix. Column "frg_id" gives a fragment unique identifier, column "su_id" gives an unique identifier for the group of similar fragments it belongs (similarity unit).
- `fragments.info` is 177x6 data frame:
  - frg_id: fragment unique identifier
  - layer: stratigraphic layer
  - zmin: minimal depth where the fragment was found
  - zmax: maximal depth where the fragment was found
  - square: square where the fragment was found
  - sherd.type: type of pottery sherd

References

Plutniak, Sebastien, Astolfo Araujo, Simon Puaud, Jean-Georges Ferrie, Adhi Agus Oktaviana, Bambang Sugiyanto, Jean-Michel Chazine et Francois-Xavier Ricaut [2015], "Borneo as a half empty pot: Pottery assemblage from Liang Abu, East Kalimantan, Quaternary International, doi: 10.1016/j.quaint.2015.11.080.

Examples

```
data(LiangAbu)
head(fragments.info)
```

---

make_crsr_graph *Makes a "connection" relations graph including the "similarity" relations.*

Description

Takes a `frag.object` in argument and returns an undirected graph representing the relations between archaeological fragments. "Connection" and "similiarity" relations are combined. A "connection" relation refers to a physical connection between two fragments which were part of a same object. A "similarity" relation between fragments is defined if there is an acceptable likelihood that those fragments were part of a same object.

Usage

```
make_crsr_graph(object)
```

Arguments

object	A frag.object.

Details

A complementary function to the make_cr_graph function. This function is devoted to take into account the "similarity" relations as well. This additional features is motivated by the fact that "similarity" relations more frequently documented in the archaeological datasets than the "connection" relations.

The function returns an undirected graph of "igraph" class, using the "fragments" data frame of the frag.object to set the vertices attributes.

Both "connection" and "similarity" relations are included in the resulting graph. A "type" edge attribute is set with a character "cr" value for "connection" relationships and "sr" for the "similarity" relationships. Edge weights are not set by this function, and it is recommended to use the frag.edges.weighting function. A "frag_type" graph attribute is set with a "connection and similarity" value.

Value

An undirected of "igraph" class graph.

Author(s)

Sebastien Plutniak <sebastien.plutniak at posteo.net>

See Also

make_frag_object, make_cr_graph

Examples

```
cr.df <- matrix(c(1,2, 1,3, 2,3, 4,5, 4,6, 7,8), ncol=2, byrow=TRUE)
sr.df <- matrix( c(1,1, 9,1, 10,1, 11,2, 12,2, 13,2), ncol=2, byrow=TRUE)
fragments.df <- data.frame(1:13, letters[1:13])

cr_g <- make_frag_object(cr=cr.df, fragments=fragments.df)
crsr_g <- make_frag_object(cr=cr.df, sr=sr.df, fragments=fragments.df)

make_cr_graph(cr_g)
make_cr_graph(crsr_g)
```

make_cr_graph *Make a "connection" relations graph.*

Description

Takes a frag.object and returns an undirected graph representing the "connection" relations between archaeological fragments. A "connection" relation refers to a physical connection between two fragments which were part of a same object.

Usage

```
make_cr_graph(object)
```

Arguments

object	A frag.object object.

Details

Returns an undirected graph of "igraph" class. The "fragments" data frame of the frag.object is used to set the vertices attributes.

Value

An undirected "igraph" class graph. A "frag_type" graph attribute is set with a "connection" value, describing the type of the resultant fragmentation graph.

Author(s)

Sebastien Plutniak <sebastien.plutniak at posteo.net>

See Also

make_frag_object

Examples

```
cr.df <- matrix(c(1,2, 1,3, 2,3, 4,5, 4,6, 7,8), ncol=2, byrow=TRUE)
sr.df <- matrix( c(1,1, 9,1, 10,1, 11,2, 12,2, 13,2), ncol=2, byrow=TRUE)
fragments.df <- data.frame(1:13, letters[1:13])

cr_g <- make_frag_object(cr=cr.df, fragments=fragments.df)
crsr_g <- make_frag_object(cr=cr.df, sr=sr.df, fragments=fragments.df)

make_cr_graph(cr_g)
make_cr_graph(crsr_g)
```

make_frag_object *Makes a "frag.object" object.*

Description

Makes a "frag.object" object.

Usage

```
make_frag_object(cr, sr, fragments)
```

Arguments

cr	A matrix or a data frame with two columns giving the vertex id of each pair of connected fragments.
sr	Optional. A matrix or a data frame with two columns: the first gives the fragment id, the second gives the "similarity group" id. Optional if mode is "cr".
fragments	A matrix or a data frame providing informations about each fragment. The first column must contain the fragments id.

Details

This function controls the data set construction before to make the graph properly with make_cr_graph or make_crsr_graph.

Value

An object of "frag.object" class.

Author(s)

Sebastien Plutniak <sebastien.plutniak at posteo.net>

Examples

```
cr.df <- matrix(c(1,2, 1,3, 2,3, 4,5, 4,6, 7,8), ncol=2, byrow=TRUE)
sr.df <- matrix( c(1,1, 9,1, 10,1, 11,2, 12,2, 13,2), ncol=2, byrow=TRUE)
fragments.df <- data.frame(1:13, letters[1:13])

make_frag_object(cr=cr.df, fragments=fragments.df)
make_frag_object(cr=cr.df, sr=sr.df, fragments=fragments.df)
```

Index