Package ‘loop’

February 20, 2015

Type Package
Title loop decomposition of weighted directed graphs for life cycle analysis, providing flexible network plotting methods, and analyzing food chain properties in ecology
Version 1.1
Date 2012-09-30
Author Youhua Chen
Maintainer Youhua Chen <haydi@126.com>
Description The program can perform loop analysis and plot network structure (especially for food webs), including minimum spanning tree, loop decomposition of weighted directed graphs, and other network properties which may be related to food chain properties in ecology.
License GPL-2
LazyLoad yes
Depends grid, MASS
Repository CRAN
Date/Publication 2012-10-01 05:51:43
NeedsCompilation no

R topics documented:

  loop-package .......................................................... 2
  conversion ........................................................... 3
  decomp ............................................................... 4
  find.ranks ............................................................ 5
  fplot ................................................................. 6
  fplot.foodweb ....................................................... 7
  gplot ................................................................. 8
  gplot1 ............................................................... 9
  groupplot .......................................................... 10
  groupplot.foodweb ................................................ 12
Description

The package can (1) perform loop analysis (van Groenendael et al., 1994) in demographic analysis in ecology, which employed the decomposition algorithm of directed weighted graphs presented in Su and Wang (2007); (2) find food web patterns, including longest food chain, largest-weighted food chain, enumeration of all food chains linked to the proposed species in the food network; (3) plot directed/undirected network structure using flexible methods; (4) other statistics, like minimum spanning tree etc. Next version will include more statistics on food web structure.

Details

Package: loop
Type: Package
Version: 1.0
Date: 2012-06-22
License: GPL-2
LazyLoad: yes

Author(s)

Youhua Chen Maintainer: Youhua Chen <haydi@126.com>

References


Examples

```
gemat=matrix(c(13,26,0,11,18,15,0,2),nrow=3,ncol=3)
gemat=matrix(c(13,26,0,12,18,15,0,2),nrow=3,ncol=3)
#search out all loops
decomp(gemat)
#make plots
mat<-matrix(c(1,2,1,3,2,4,3,5,4,5,6,7,8,9,1,9,9,8),nrow=9,ncol=2)
w<-c(3,10,30,50,20,22,9,15,33)
mat<-cbind(mat,w)
gplot(mat)
```

---

### conversion

Converts a graph matrix form into a graph edge form.

#### Description

Graph matrix form is a square matrix in which each element a(ij) denotes the weight of a direction from a row entry (row i) to a column entry (column j).

#### Usage

```
conversion(gemat)
```

#### Arguments

- **gemat**: the square graph matrix

#### Details

Each element must be larger (a weighted link) or equal to zero (no link).

#### Value

Graph edge matrix with first column meaning the starting points, the second column denoting the ending points, the last column denotes the weights for the links.

#### Author(s)

Youhua Chen <haydi@126.com>
References


See Also

decomp, gplot, loop.forward, loop.random

Examples

```r
mat = matrix(c(1, 3, 2, 6, 0, 11, 18, 15, 0, 2), nrow = 3, ncol = 3)
converion(gemat = mat)
```

Description

Decompose the weighted graph into directional single connected graphs and a non-connected graph.

Usage

decomp(gemat)

Arguments

gemat the square graph matrix

Value

Return all single connected directed graphs and a final non-decomposable graph

Author(s)

Youhua Chen <haydi@126.com>

References


**find.ranks**

See Also

`gplot, fplot, fplot.foodweb, loop.forward, loop.random`

Examples

```r
mat=matrix(c(1,3,2,6,0,1,12,18,15,0,2),nrow=3,ncol=3)
#search out all loops
decomp(gemat=mat)
```

```r
find.ranks(gemat=mat)
```

Description

it will return trophic ranks of species in the food webs

Usage

```r
find.ranks(gemat, converted = TRUE)
```

Arguments

- `gemat`: `gemat` is a square matrix
- `converted`: if TRUE, resultant trophic ranks of species will be converted to a list, which is the input format for `fplot.foodweb()` function; if FALSE, resultant trophic ranks of species are present in a vector default is TRUE

Value

it will return a list (converted=TRUE), each list represent one trophic rank, the elements of which are the species being classified into the rank. it can also return a vector (converted=FALSE), each element denote the rank of the species.

Author(s)

Youhua Chen <haydi@126.com>

References


See Also

`fplot.foodweb`
Examples

```r
gemat<-matrix(c(0,5,3,7,0,5,0,8,0,4,3,8,0,1,6,7,0,1,0,2,0,4,6,2,0),5,5)
ranks<-find.ranks(gemat)
fplot.foodweb(gemat=gemat,ranks=ranks,addlabels=TRUE,bg="white",pch=21,pcex=4,lcol=2,weighted=FALSE)
```

Another way to plot network structure based on similar vertex will be grouped together, while dissimilar nodes/vertex will depart from each others.

Description

The method for seperating the nodes in the two-dimensional spaces is the non-dimensional scaling technique, which can take the similarity matrices of the nodes as the input and generate the positions of the nodes in the space.

Usage

```r
fplot(gemat,type="both",metric="jaccard",addlabels=FALSE,scaled=TRUE,weighted=TRUE,pch=20,bg=1,pcex=4,lcol=2,lty=1)
```

Arguments

- `gemat`: standard graph square matrix
- `type`: if `type="both"`, the node similarity is calculated based on the the vertex similarity from the inward/outward links for each pair of nodes. if `type="in"`, the node similarity is calculated based on the the vertex similarity from the inward links for each pair of nodes. if `type="out"`, the node similarity is calculated based on the the vertex similarity from the outward links for each pair of nodes.
- `metric`: node similarity methods, currently supporting two basic similarity indices: "jaccard" and "sorensen".
- `addlabels`: if you want to label each node/vertex, set it’s status as TRUE; default is FALSE
- `scaled`: if you want to the links showing relative weights, set it’s status as TRUE; default is FALSE links with larger weights will have thicker line width, vice versa.
- `weighted`: if TRUE, the links/edges will be weighted based on the cell value present in the matrix of gemat, different edges then will have different line widths for representing them. Otherwise, all edges have the same line width. Default is TRUE
- `pch`: this pch is for nodes/vertex
- `bg`: bg is for nodes/vertex filled background colors, will function when pch=21:25.
- `pcex`: pcex is for nodes/vertex size
- `pcol`: pcol is for nodes/vertex color
- `lty`: lty is the line style for the links
- `lcol`: lcol is the line color for the links
- `tfont`: tfont is the font size for the labels of the nodes
- `tcol`: tcol is the color for the labels of the nodes
**fplot.foodweb**

**Author(s)**
Youhua Chen <haydi@126.com>

**References**


**See Also**

fplot.foodweb, groupplot.foodweb, gplot, gplot1, groupplot

**Examples**

```r
mat = matrix(c(0, 5, 3, 7, 0, 5, 8, 0, 4, 8, 0, 1, 6, 7, 0, 1, 0, 2, 0, 4, 6, 2, 0), 5, 5)
fplot(gemat = mat) #other parameters are set in default
```

**Description**

vertex/nodes/species are arranged in a form of vertical hierarchy, given the species' food rank positions listed in the parameter "ranks". This method thus does not to calculate node similarity based on inward/outward link similarity.

**Usage**

```r
fplot.foodweb(gemat, ranks, addlabels = FALSE, scaled = TRUE, weighted = TRUE, pch = 20, bg = 1, pcol = 3, lty = 1)
```

**Arguments**

- `gemat` standard square graph matrix
- `ranks` must be a list, each list element represented the species in that given rank (list number order), for example list[[1]] means all the species in food web hierarchy rank 1, and so on...
- `addlabels` if you want to label each node/vertex, set it's status as TRUE; default is FALSE
- `scaled` if you want to the links showing relative weights, set it's status as TRUE; default is FALSE links with larger weights will have thicker line width, vice versa.
- `weighted` if TRUE, the links/edges will be weighted based on the cell value present in the matrix of gemat, different edges then will have different line widths for representing them. Otherwise, all edges have the same line width. Default is TRUE
- `pch` this pch is for nodes/vertex
- `bg` bg is for nodes/vertex filled background colors, will function when pch=21:25.
gplot

make a plot for directional graph from edge matrix graph form

Description

can change the link line style and color, can add arrows at the ending points, this function requires an edge matrix form of graphs, compared to another similar function 'gplot1'.

Usage

gplot(edgemat, arrow=TRUE, lty=1, col=8, weighted=TRUE)
gplot1

Arguments

edgemat  graph edge matrix form, without confusion, 'loop' package will have two types for the graph storage: square matrix from (argument is 'gemat') and edge matrix form (argument is 'edgemat').

arrow  default is TRUE, denoting arrows should be added in the plot

lty  line style for each graph link

col  line color for each graph link

weighted  if TRUE, the links/edges will be weighted based on the value present in the 3rd column of edgemat, different edges then will have different line widths for representing them. Otherwise, all edges have the same line width. Default is TRUE

Author(s)

Youhua Chen <haydi@126.com>

References


See Also

decomp, loop.forward, loop.random, gplot1, fplot, groupplot

Examples

#make plots
mat<-matrix(c(1,2,1,3,2,4,3,5,4,6,5,7,8,9,1,9,9,8),nrow=9,ncol=2)
w<-c(3,10,30,50,20,22,9,15,33)
mat<-cbind(mat,w)
gplot(edgemat=mat)

---

gplot1  plot directed graphs directly from square graph matrix

Description

don't need to convert square graph matrix into edge form compared to the function "gplot"

Usage

gplot1(gemat,arrow=TRUE,lty=1,col=8,weighted=TRUE)
Arguments

- **gemat**: standard square graph matrix
- **arrow**: add arrows or not, from the starting node to the ending node, default is TRUE.
- **lty**: line style for the links/edges, default is 1.
- **col**: color for the links/edges, default is 8 (grey).
- **weighted**: if TRUE, the links/edges will be weighted based on the cell value present in the matrix of gemat, different edges then will have different line widths for representing them. Otherwise, all edges have the same line width. Default is TRUE.

Author(s)

Youhua Chen <haydi@126.com>

References


See Also

gplot, fplot, groupplot.

Examples

```r
mat = matrix(c(0, 5, 3, 7, 0, 5, 0, 8, 0, 4, 3, 8, 0, 1, 6, 7, 0, 1, 0, 2, 0, 4, 6, 2, 0), 5, 5)
gplot1(gemat = mat) # other parameters are set in default
```

Description

This function is for plotting different graphs into a same map with different colors, line styles and so on. The method for separating the nodes in the two-dimensional spaces is the non-dimensional scaling technique, which can take the similarity matrices of the nodes as the input and generate the positions of the nodes in the space.

Usage

groupplot(gemat, groups, type = "both", metric = "jaccard", addlabels = FALSE, scaled = TRUE, pch = 20)
Arguments

- **gemat**: standard graph square matrix
- **groups**: must be a list, each list element has the species that belong to a same group.
- **type**: if type="both", the node similarity is calculated based on the the vertex similarity from the inward/outward links for each pair of nodes. if type="in", the node similarity is calculated based on the the vertex similarity from the inward links for each pair of nodes. if type="out", the node similarity is calculated based on the the vertex similarity from the outward links for each pair of nodes.
- **metric**: node similarity methods, currently supporting two basic similarity indices: "jaccard" and "sorensen".
- **addlabels**: if you want to label each node/vertex, set it's status as TRUE; default is FALSE
- **scaled**: if you want to the links showing relative weights, set it's status as TRUE; default is FALSE links with larger weights will have thicker line width, vice versa.
- **pch**: this pch is for nodes/vertex
- **bg**: bg is for nodes/vertex filled background colors, will function when pch=21:25.
- **pcex**: pcex is for nodes/vertex size
- **pcol**: pcol is for nodes/vertex color
- **lty**: lty is the line style for the links
- **lcol**: lcol is the line color for the links
- **tfont**: tfont is the font size for the labels of the nodes
- **tcol**: tcol is the color for the labels of the nodes

Author(s)

Youhua Chen <haydi@126.com>

References


See Also

- `fplot.foodweb`, `groupplot.foodweb`, `gplot`, `gplot1`

Examples

```r
mat=matrix(c(0,5,3,7,0,5,0,8,4,3,8,0,1,6,7,0,1,0,2,0,4,6,2,0),5,5)
glist<-list(c1=c(1:5),c2=c(1,3),c3=c(3,4,5))
pch=c(21,21,21) #length of the parameter vector should be identical to the number of species groups
bg=c(1,2,3)
lcol=c(8,2,4)
groupplot(gemat=mat,groups=glist,pch=pch,lcol=lcol,bg=bg) #other parameters are set in default
```
Description

vertex/nodes/species are arranged in a form of vertical hierarchy, given the species’ food rank positions listed in the parameter "ranks". "groups" are a list for each list element showing a group of species. They should be defined based on the user’s choices. This method thus does not calculate node similarity based on inward/outward link similarity.

Usage

```r
groupplot.foodweb(gemat, ranks, groups, addlabels = FALSE, scaled = TRUE, pch = 20, bg = 1, pcex = 3, pcol = 1, lty = 1, lcol = 8, tfont = 1, tcol = 1)
```

Arguments

- `gemat`: standard square graph matrix
- `ranks`: a list for each list member showing the species in that rank (list number), for example list[[1]] means all the species in food web hierarchy rank 1, and so on...
- `groups`: a list of species groups. Can be overlapped among the list elements. But should be defined by the users.
- `addlabels`: if you want to label each node/vertex, set it’s status as TRUE; default is FALSE
- `scaled`: if you want the links showing relative weights, set it’s status as TRUE; default is FALSE links with larger weights will have thicker line width, vice versa.
- `pch`: this pch is for nodes/vertex
- `bg`: bg is for nodes/vertex filled background colors, will function when pch=21:25.
- `pcex`: pcex is for nodes/vertex size
- `pcol`: pcol is for nodes/vertex color
- `lty`: lty is the line style for the links
- `lcol`: lcol is the line color for the links
- `tfont`: tfont is the font size for the labels of the nodes
- `tcol`: tcol is the color for the labels of the nodes

Author(s)

Youhua Chen <haydi@126.com>

References

See Also

fplot.foodweb, gplot, gplot1, fplot, groupplot

Examples

mat=matrix(c(0,5,3,7,0,0,8,4,3,8,0,1,6,7,0,1,0,2,0,4,6,2,0),5,5)
rlist<-list(c1=c(1,2),c2=c(3,4),c3=5)
glist<-list(c1=c(1:5),c2=c(1,2,3),c3=c(2,4,5))
groupplot.foodweb(gemat=mat,ranks=rlist,groups=glist) #other parameters are set in default
#because the function found that the parameters (especially the colors) are not set
another example that can separate the groups
rlist<-list(c1=c(1,2),c2=c(3,4),c3=5)
glist<-list(c1=c(1:5),c2=c(1,3),c3=c(3,4,5))
pch=c(20,22,24) #length of the parameter vector should be identical to the number of species groups
lcol=c(8,2,4)
groupplot.foodweb(gemat=mat,ranks=rlist,groups=glist,pch=pch,lcol=lcol) #other parameters are set in default

largest.weight

find out the food chain with largest weights among all food chains that are linked to a given species

Description

a food chain with largest weights mean that the summation of weights for each link of the chain is the largest. Note: the longest food chain (with largest number of nodes) does not necessarily have the largest weight.

Usage

largest.weight(gemat, sp)

Arguments

gemat standard square graph matrix

sp the node that is evaluated.

Value

return all candidate pathways and related weights

Author(s)

Youhua Chen <haydi@126.com>

References

See Also

`shortest.chain`, `longest.chain`, `smallest.weight`, `lclw`

Examples

```r
#mat=matrix(c(0,5,3,7,0,5,0,8,0,4,3,8,0,1,6,7,0,1,0,2,0,4,6,2,0),5,5)
#search for species 3
#largest.weight(mat,3)
```

`lclw` find the largest weight food chain among the candidate longest food chains for a given node (species)

Description

Sometimes for a given species, it has many food chains that are equal in chain lengths and all are longest. This function thus will find out the one with largest weight.

Usage

```r
lclw(gemat, sp)
```

Arguments

- `gemat` : standard square graph matrix
- `sp` : the node/species you want to search

Value

return the pathways and corresponding weights in a list

Author(s)

Yohuha Chen <haydi@126.com>

References


See Also

`shortest.chain`, `largest.weight`, `smallest.weight`, `longest.chain`
longest.chain

Examples

#mat=matrix(c(0,5,3,7,0,5,0,8,0,4,3,8,0,1,6,7,0,1,0,2,0,4,6,2,0),5,5)
#search for species 3
#lclw(gemat=mat,sp=3)
#longest chain isn't necessary to have largest weights
#largest.weight(gemat=mat,sp=3)
#longest.chain(gemat=mat,sp=3)

longest.chain find out the food chain with longest length among all food chains that are linked to a given species

Description

Sometimes the longest chain is not unique, thus will return all candidates with longest length

Usage

longest.chain(gemat, sp)

Arguments

gemat standard square graph matrix
sp the node/species you want to search

Author(s)

Youhua Chen <haydi@126.com>

References


See Also

shortest.chain, largest.weight, smallest.weight, lclw

Examples

#mat=matrix(c(0,5,3,7,0,5,0,8,0,4,3,8,0,1,6,7,0,1,0,2,0,4,6,2,0),5,5)
#search for species 3
#lclw(gemat=mat,sp=3)
#longest chain isn't necessary to have largest weights
#largest.weight(gemat=mat,sp=3)
#longest.chain(gemat=mat,sp=3)
loop.forward

forward method to decompose the weighted graph into single connected graphs

Description

At each step, only choose the first one of all candidate nodes when finding next node in the process of making a single connected graph. This is an internal function.

Usage

loop.forward(gemat)

Arguments

gemat  square graph matrix form

Author(s)

Youhua Chen <haydi@126.com>

References


See Also

decomp, gplot, loop.random

loop.random

a random method to decompose the weighted graph into single connected graphs

Description

At each step, a random one of all candidate nodes was chosen when finding next node in the process of making a single connected graph. This method can allow us to obtain different decomposition results possibly. In contrast, the method used in 'loop.forward' function will always return only one possibility of decomposition. This algorithm is the one employed in the function 'decomp'. Next version will use function 'uniquepaths' to get all possible decompositions at the same time.
mst.primm

Usage

loop.random(gemat)

Arguments

gemat square graph matrix

Author(s)

Youhua Chen <haydi@126.com>

References


See Also

decom.pplot.gplot1, loop.forward

_________________________________________________________________

mst.primm

Primm's algorithm to search minimum spanning tree

Description

Each time search two nodes with lowest weights in an iterative manner

Usage

mst.primm(gemat)

Arguments

gemat standard graph square matrix

Value

return a matrix in edge form, the first column is the starting node, the second column is the ending node

Author(s)

Youhua Chen <haydi@126.com>
nmds.ordination

References


See Also

groupplot

Examples

mat = matrix(c(0, 5, 3, 7, 0, 5, 8, 0, 4, 3, 0, 1, 6, 7, 0, 1, 0, 2, 0, 4, 6, 2, 0), 5, 5)
mst.primm(mat)

nmds.ordination

perform non-dimensional scaling of the nodes by employing the function 'isoMDS' from package 'MASS'

Description

this is to get the coordinates of each node/vertex in the two-diemsional spaces, which are used in plotting functions, including "fplot" and "groupplot".

Usage

nmds.ordination(gemat, type = "both", metric = "jaccard")

Arguments

gemat standard square graph matrix

type if type="both", the node similarity is calculated based on the the vertex similarity from the inward/outward links for each pair of nodes. if type="in", the node similarity is calculated based on the the vertex similarity from the inward links for each pair of nodes. if type="out", the node similarity is calculated based on the the vertex similarity from the outward links for each pair of nodes.

metric node similarity methods, currently supporting two basic similarity indices: "jaccard" and "sorensen".

Value

coord coordinates of each node in the space

names names for the nodes

Author(s)

Youhua Chen <haydi@126.com>
References


See Also

fplot, groupplot, node.similarity

Examples

```r
mat=matrix(c(0,5,3,7,0,5,0,8,0,4,3,8,0,1,6,7,0,1,0,2,0,4,6,2,0),5,5)
nmds.ordination(gemat=mat)
```

```r
node.similarity(gemat, type = "both", metric = "jaccard")
```

Description

The pair similarity of nodes is defined as the exterior nodes shared by both focused nodes (i.e., there must be links between the focused nodes and the exterior nodes).

Usage

```r
node.similarity(gemat, type = "both", metric = "jaccard")
```

Arguments

- **gemat** standard square graph matrix
- **type** if type="both", the node similarity is calculated based on the the vertex similarity from the inward/outward links for each pair of nodes. if type="in", the node similarity is calculated based on the the vertex similarity from the inward links for each pair of nodes. if type="out", the node similarity is calculated based on the the vertex similarity from the outward links for each pair of nodes.
- **metric** node similarity methods, currently supporting two basic similarity indices: "jaccard" and "sorensen".

Value

will return a symmetric similarity matrices

Author(s)

Youhua Chen <haydi@126.com>
References


See Also

nmds.ordination

Examples

mat = matrix(c(0,5,3,7,0,5,0,0,4,3,8,0,1,0,7,0,1,0,0,0,4,6,2,0),5,5)
# compare the differences for each type of links
node.similarity(gemat=mat,type="in")
node.similarity(gemat=mat,type="out")
node.similarity(gemat=mat,type="both")

Description

the food chains can allow circles/loop. This function will return all possible food chains, and some duplicated chains are generated as well. If you want to get unique food chains, try the function 'uniquepaths'

Usage

pathways(gemat, bsp)

Arguments

gemat: standard square graph matrix
bsp: the species/node you want to search. If input is a vector, then the function will return the food chains for each element species in the vector. Please note that all pathways/chains are started with the focused vertext/node/species, ended by a circle/loops; or upto the top of the network without a circle/loop.

Value

return a list showing all food chains by presenting each node in that chain. Also, the weights for each link will be returned as well.

Author(s)

Youhua Chen <haydi@126.com>
rank.nodes

References


See Also

uniquepaths, loop.forward, loop.random

Examples

```r
mat = matrix(c(0, 5, 3, 7, 0, 5, 0, 8, 0, 4, 3, 8, 0, 1, 6, 7, 0, 1, 0, 2, 0, 4, 6, 2, 0), 5, 5)
pathways(mat, bsp = 2)
# a vector of species
pathways(mat, bsp = c(1, 3, 5))
```

<table>
<thead>
<tr>
<th>rank.nodes</th>
<th>rank nodes based on the inward links/outward links/both links</th>
</tr>
</thead>
</table>

Description

rank each node in the graph based on their connection status with other nodes, the standard is the inward/outward link numbers.

Usage

```r
rank.nodes(gemat, type = "both")
```

Arguments

- `gemat`: standard square graph matrix
- `type`: if `type="both"`, the function will rank the nodes based on total link number, by counting both inward and outward links for each node; if `type="in"`, the function will rank the nodes based on inward link number for each node; if `type="out"`, the function will rank the nodes based on outward link number for each node;

Value

return the sorted node vector in decreasing form with names showing the nodes, while the vector elements are the link numbers.

Author(s)

Youhua Chen <haydi@126.com>

References

shortest.chain

See Also

node.similarity, nmds.ordination

Examples

mat = matrix(c(0, 5, 3, 7, 0, 5, 0, 8, 0, 4, 3, 8, 0, 1, 6, 7, 0, 1, 0, 2, 0, 4, 6, 2, 0, 5, 5)
rank.nodes(gemat = mat, type = "in")
rank.nodes(gemat = mat, type = "out")
rank.nodes(gemat = mat, type = "both")

shortest.chain  find out the food chain with shortest length among all food chains that are linked to a given species

Description

not including self-loop (i.e., the node itself as a chain)

Usage

shortest.chain(gemat, sp)

Arguments

gemat          standard square graph matrix
sp             the node/species you want to search

Author(s)

Youhua Chen <haydi@126.com>

References


See Also

longest.chain, largest.weight, smallest.weight, lclw

Examples

mat = matrix(c(0, 5, 3, 7, 0, 5, 0, 8, 0, 4, 3, 8, 0, 1, 6, 7, 0, 1, 0, 2, 0, 4, 6, 2, 0, 5, 5)
shortest.chain(gemat = mat, sp = 4)
**smallest.weight**

**find out the food chain with shortest weight among all food chains that are linked to a given species**

**Description**

not including self-loop (i.e., the node itself as a chain)

**Usage**

```
smallest.weight(gemat, sp)
```

**Arguments**

- **gemat** standard square graph matrix
- **sp** the node/species you want to search

**Author(s)**

Youhua Chen <haydi@126.com>

**References**


**See Also**

- `shortest.chain`, `largest.weight`, `longest.chain`, `lclw`

**Examples**

```
mat=matrix(c(0,5,3,7,0,5,0,8,0,4,3,8,0,1,6,7,0,1,0,2,0,4,6,2,0,5,5)
shortest.chain(gemat=mat, sp=4)
```

**uniquepaths**

**enumerate all unique food chains that are linked to a given species/node**

**Description**

the food chains can allow circles/loop. This function will return all unique food chains that involved the focused species as a node, and the duplicated chains are deleted. It is recommended to use rather than the function 'pathways'.
Usage

`uniquepaths(gemat, sp)`

Arguments

gemat standard square graph matrix
sp the species you want to search.

Value

return a list showing all food chains by presenting each node in that chain. Also, the weights for each link will be returned as well.

Author(s)

Youhua Chen <haydi@126.com>

References


See Also

`pathways`

Examples

```r
mat = matrix(c(0, 5, 3, 7, 0, 5, 0, 8, 0, 4, 3, 8, 0, 1, 6, 7, 0, 1, 0, 2, 0, 4, 6, 2, 0), 5, 5)
uniquepaths(mat, sp=2)
```
Index

*Topic \textasciitilde kwd1
  conversion, 3

*Topic \textasciitilde kwd2
  conversion, 3

conversion, 3

decomp, 4, 9, 16, 17

find.ranks, 5
fplot, 5, 6, 8–10, 13, 19
fplot.foodweb, 5, 7, 11, 13

gplot, 4, 5, 7, 8, 10, 11, 13, 16, 17
gplotl, 7, 9, 11, 13, 17
groupplot, 7–10, 10, 13, 18, 19
groupplot.foodweb, 7, 8, 11, 12

largest.weight, 13, 14, 15, 22, 23
lclw, 14, 15, 22, 23
longest.chain, 14, 15, 22, 23
loop (loop-package), 2
loop-package, 2
loop.forward, 4, 5, 9, 16, 17, 21
loop.random, 4, 5, 9, 16, 16, 21

mst.primm, 17

nmds.ordination, 18, 20, 22
node.similarity, 19, 19, 22

pathways, 20, 24

rank.nodes, 21

shortest.chain, 14, 15, 22, 23
smallest.weight, 14, 15, 22, 23

uniquepaths, 21, 23