

Package ‘ConnMatTools’

August 4, 2014

Type Package

Title Tools for working with connectivity matrices

Version 0.1.5

Date 2013-07-09

Author Marco Andrello <marco.andrello@gmail.com>

Maintainer David M. Kaplan <dmkaplan2000@gmail.com>

Description ConnMatTools collects several different methods for analyzing and working with connectivity matrices in R. Though primarily oriented towards marine larval dispersal, many of the methods are general and useful for terrestrial systems as well.

URL <https://github.com/dmkaplan2000/ConnMatTools.git>

Suggests igraph

Collate 'ConnMatTools.R' 'eigs.R' 'Jacobi_EPT_reserve_selection.R'
'datasets.R' 'dpr_model.R' 'retentionStats.R' 'jacobi_etal_2012.R'

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2014-06-21 09:01:40

R topics documented:

betasVectorDefault	2
BevertonHolt	3
chile.loco	4
ConnMatTools	4
DispersalPerRecruitModel	5
DPRHomerangeGravity	7

eigs	8
hockeyStick	9
laplacianConnMat	10
localRetention	11
mergeSubpops	12
optimalSplitConnMat	13
protectedAreaSelection	14
qualitySubpops	16
recSplitConnMat	17
reducedConnMat	17
relativeLocalRetention	19
selfRecruitment	19
settlerRecruitSlopeCorrection	20
splitConnMat	21
subpopsVectorToList	22

Index **23**

betasVectorDefault *Compute vector of beta values*

Description

Helper function to compute a set of beta values using formula used in Jacobi et al. (2012).

Usage

```
betasVectorDefault(n, steps = 10, cycles = 3/4, coeff = 0.8, pwr = 3)
```

Arguments

n	numerator of formula from Jacobi et al. (2012). Normally will be the number of columns in the connectivity matrix if one normalizes the columns (otherwise, it would typically be $N^2 / \text{sum}(\text{conn.mat})$, where N is the number of columns of conn.mat.
steps	number of beta values to return. Defaults to 10.
cycles	how many cycles of 2π to do.
coeff	coefficient in front of sine function
pwr	exponent in denominator

Value

vector of beta values

Author(s)

David M. Kaplan <dmkaplan2000@gmail.com>

References

Jacobi, M. N., André, C., Döös, K., and Jonsson, P. R. 2012. Identification of subpopulations from connectivity matrices. *Ecography*, 35: 1004-1016.

See Also

See also [optimalSplitConnMat](#)

BevertonHolt

Beverton-Holt settler-recruit relationship

Description

Calculates recruitment based on the settler-recruit relationship from Beverton & Holt (1957): $\text{slope} * \text{settlers} / (1 + \text{slope} * \text{settlers})$

Usage

```
BevertonHolt(S, slope = 1/0.35, Rmax = 1)
```

Arguments

S	a vector of settlement values, 1 for each site.
slope	slope at the origin of the settler-recruit relationship. Can be a vector of same length as S.
Rmax	maximum recruitment value.

Details

slope and Rmax can both either be scalars or vectors of the same length as S.

Value

A vector of recruitment values.

Author(s)

David M. Kaplan <dmkaplan2000@gmail.com>

References

Beverton RJH, Holt SJ (1957) On the dynamics of exploited fish populations. H.M.S.O., London. 533 pp.

`chile.loco`*Connectivity matrix for loco (Concholepas concholepas) from Chile*

Description

Sample connectivity matrix representing potential larval dispersal of loco (*Concholepas concholepas*) from Chile. The matrix is for 89 sites along the coast of Chile and is derived from a theoretical larval transport model.

Author(s)

David M. Kaplan <dmkaplan2000@gmail.com>

`ConnMatTools`*Tools for working with connectivity matrices*

Description

Tools for working with connectivity matrices

Details

ConnMatTools collects several different methods for analyzing and working with connectivity matrices in R. Though primarily oriented towards marine larval dispersal, many of the methods are general and useful for terrestrial systems as well.

Package:	ConnMatTools
Type:	Package
Version:	0.1.5
Date:	2013-07-09
License:	GPL (>= 2)
LazyLoad:	no

Author(s)

David M. Kaplan <dmkaplan2000@gmail.com>

Marco Andrello <marco.andrello@gmail.com>

References

Jacobi, M. N., and Jonsson, P. R. 2011. Optimal networks of nature reserves can be found through eigenvalue perturbation theory of the connectivity matrix. *Ecological Applications*, 21: 1861-1870.

Jacobi, M. N., Andre, C., Doos, K., and Jonsson, P. R. 2012. Identification of subpopulations from connectivity matrices. *Ecography*, 35: 1004-1016.

Gruss, A., Kaplan, D. M., and Lett, C. 2012. Estimating local settler-recruit relationship parameters for complex spatially explicit models. *Fisheries Research*, 127-128: 34-39.

Kaplan, D. M., Botsford, L. W., and Jorgensen, S. 2006. Dispersal per recruit: An efficient method for assessing sustainability in marine reserve networks. *Ecological Applications*, 16: 2248-2263.

White, J. W. 2010. Adapting the steepness parameter from stock-recruit curves for use in spatially explicit models. *Fisheries Research*, 102: 330-334.

Gruss A, Kaplan DM, Hart DR (2011) Relative Impacts of Adult Movement, Larval Dispersal and Harvester Movement on the Effectiveness of Reserve Networks. *PLoS ONE* 6:e19960

Beverton RJH, Holt SJ (1957) On the dynamics of exploited fish populations. H.M.S.O., London. 533 pp.

See Also

See [optimalSplitConnMat](#)

Examples

```
## Not run: optimalSplitConnMat(CM)
```

DispersalPerRecruitModel

Population dynamics model based on lifetime-egg-production

Description

This function implements the marine population dynamics model described in Kaplan et al. (2006). This model is most appropriate for examining equilibrium dynamics of age-structured populations or temporal dynamics of semelparous populations.

Usage

```
DispersalPerRecruitModel(LEP, conn.mat, recruits0, timesteps = 10,
  settler.recruit.func = hockeyStick, ...)
```

Arguments

LEP	a vector of lifetime-egg-production (LEP; also known as eggs-per-recruit (EPR)) for each site.
conn.mat	a square connectivity matrix. $\dim(\text{conn.mat}) = \text{rep}(\text{length}(\text{LEP}), 2)$
recruits0	a vector of initial recruitment values for each site.
timesteps	a vector of timesteps at which to record egg production, settlement and recruitment.
settler.recruit.func	a function to calculate recruitment from the number of settlers at each site. Defaults to hockeyStick .
...	additional arguments to settler.recruit.func. Typically Rmax and slope.

Value

A list with the following elements:

eggs	egg production for the timesteps in timesteps
settlers	Similar for settlement
recruits	Similar for recruitment

Author(s)

David M. Kaplan <dmkaplan2000@gmail.com>

References

Kaplan, D. M., Botsford, L. W., and Jorgensen, S. 2006. Dispersal per recruit: An efficient method for assessing sustainability in marine reserve networks. *Ecological Applications*, 16: 2248-2263.

See Also

See also [BevertonHolt](#), [hockeyStick](#)

Examples

```
library(ConnMatTools)
data(chile.loco)

# Get appropriate collapse slope
# critical.FLEP=0.2 is just an example
slope <- settlerRecruitSlopeCorrection(chile.loco,critical.FLEP=0.2)

# Make the middle 20 sites a reserve
# All other sites: scorched earth
n <- dim(chile.loco)[2]
LEP <- rep(0,n)
nn <- round(n/2)-9
LEP[nn:(nn+19)] <- 1

Rmax <- 1

recruits0 <- rep(Rmax,n)

# Use DPR model
ret <- DispersalPerRecruitModel(LEP,chile.loco,recruits0,1:20,slope=slope,Rmax=Rmax,
                               settler.recruit.func=BevertonHolt)
image(1:n,1:20,ret$settlers,xlab="sites",ylab="timesteps",
      main=c("Settlement","click to proceed"))
locator(1)
plot(ret$settlers[,20],xlab="sites",ylab="equilibrium settlement",
     main="click to proceed")
locator(1)

# Same, but with a uniform Laplacian dispersal matrix and hockeyStick
```

```

cm <- laplacianConnMat(n,10,0,"circular")
ret <- DispersalPerRecruitModel(LEP,cm,recruits0,1:20,slope=1/0.35,Rmax=Rmax)
image(1:n,1:20,ret$settlers,xlab="sites",ylab="timesteps",
      main=c("Settlement","click to proceed"))
locator(1)
plot(ret$settlers[,20],xlab="sites",ylab="equilibrium settlement")

```

DPRHomerangeGravity *Extended DPR population dynamics model to include homerange movement*

Description

This function implements the marine population dynamics model described in Gruss et al. (2011). The model is an extension of the dispersal-per-recruit model in Kaplan et al. (2006) to include movement in a homerange and a gravity model for fishing effort redistribution.

Usage

```

DPRHomerangeGravity(larval.mat, adult.mat, recruits0, f0, timesteps = 10,
  settler.recruit.func = hockeyStick, LEP.of.f = function(f) 1 - f,
  YPR.of.f = function(f) f, gamma = 0, gravity.ts.interval = 1, ...)

```

Arguments

larval.mat	a square larval connectivity matrix. $\dim(\text{larval.mat}) = \text{rep}(\text{length}(\text{recruits0}), 2)$
adult.mat	a square adult homerange movement matrix. $\dim(\text{adult.mat}) = \text{rep}(\text{length}(\text{recruits0}), 2)$. adult.mat must be properly normalized so that each column sums to 1.
recruits0	a vector of initial recruitment values for each site.
f0	a vector of initial real fishing mortalities for each site.
timesteps	a vector of timesteps at which to record egg production, settlement and recruitment.
settler.recruit.func	a function to calculate recruitment from the number of settlers at each site. Defaults to hockeyStick .
LEP.of.f	a function that returns lifetime-egg-productions given a vector of fishing rates.
YPR.of.f	a function that returns yields-per-recruit given a vector of fishing rates.
gamma	exponent for the gravity model. Defaults to 0, i.e., no gravity model.
gravity.ts.interval	number of timesteps between updates of gravity model. Defaults to 1, i.e., every timestep.
...	additional arguments to settler.recruit.func.

Value

A list with the following elements:

eggs	egg production for the timesteps in timesteps
settlers	Similar for settlement
recruits	Similar for recruitment
fishing.mortality	Real spatial distribution of fishing mortality
effective.fishing.mortality	Effective fishing mortality taking into account adult movement
yield	Real spatial distribution of yield
effective.yield	Effective yield indicating where fish biomass caught originates from

Author(s)

David M. Kaplan <dmkaplan2000@gmail.com>

References

Grüss A, Kaplan DM, Hart DR (2011) Relative Impacts of Adult Movement, Larval Dispersal and Harvester Movement on the Effectiveness of Reserve Networks. PLoS ONE 6:e19960

Kaplan, D. M., Botsford, L. W., and Jorgensen, S. 2006. Dispersal per recruit: An efficient method for assessing sustainability in marine reserve networks. Ecological Applications, 16: 2248-2263.

See Also

See also [BevertonHolt](#), [hockeyStick](#), [DispersalPerRecruitModel](#)

eigs

Compute some eigenvalues of a matrix

Description

This function computes a limited number of eigenvalues and eigenvectors of a matrix. It uses [arpack](#) function from the [igraph](#) package. If this package is not available, it will use the standard [eigen](#) function to do the calculation, but will issue a warning.

Usage

```
eigs(M, nev = min(dim(M)[1] - 1, 1), sym = sum(abs(M - t(M)))/sum(abs(M)) <
  1e-10, which = "LM", use.arpack = TRUE, options.arpack = NULL)
```


Arguments

M	a matrix.
nev	number of eigenvalues and eigenvectors to return
sym	A boolean indicating if matrix is symmetric or not. Defaults to checking if this is the case or not.
which	A character string indicating which eigenvalues to return. Defaults to "LM", meaning largest magnitude eigenvalues. If not using arpack , then "SM" is also a possibility to return the smallest magnitude eigenvalues. If using arpack , then a number of options are possible, though they are not all guaranteed to work for all use cases. See that function for more details.
use.arpack	Boolean determining if calculation is to be done with arpack function from the igraph package. This is much quicker for large matrices, but requires igraph . Defaults to TRUE, but will use eigen instead if igraph is not found.
options.arpack	Additional options for arpack . See that function for details. Not all options are compatible with this function.

Value

A list with at least the following two items:

values	A set of eigenvalues
vectors	A matrix of eigenvectors

Author(s)

David M. Kaplan <dmkaplan2000@gmail.com>

See Also

See also [arpack](#)

hockeyStick	<i>Hockey-stick settler-recruit relationship</i>
-------------	--

Description

Calculates recruitment based on a settler-recruit relationship that increases linearly until it reaches a maximum values.

Usage

```
hockeyStick(S, slope = 1/0.35, Rmax = 1)
```

Arguments

S	a vector of settlement values, 1 for each site.
slope	slope at the origin of the settler-recruit relationship. Can be a vector of same length as S.
Rmax	maximum recruitment value.

Details

slope and Rmax can both either be scalars or vectors of the same length as S.

Value

A vector of recruitment values.

Author(s)

David M. Kaplan <dmkaplan2000@gmail.com>

References

Kaplan, D. M., Botsford, L. W., and Jorgensen, S. 2006. Dispersal per recruit: An efficient method for assessing sustainability in marine reserve networks. *Ecological Applications*, 16: 2248-2263.

laplacianConnMat	<i>Uniform Laplacian connectivity matrix</i>
------------------	--

Description

This function generates a connectivity matrix that is governed by a Laplacian distribution: $D[i, j] = \exp(-\text{abs}(x[i] - y[j]) - \text{shift})$

Usage

```
laplacianConnMat(num.sites, disp.dist, shift = 0, boundaries = "nothing")
```

Arguments

num.sites	number of sites. Sites are assumed to be aligned on a linear coastline.
disp.dist	dispersal distance in "site" units (i.e., 1 site = 1 unit of distance)
shift	advection distance in "site" units. Defaults to 0.
boundaries	string indicating what to do at boundaries. Defaults to "nothing". Possible values include: "nothing", "conservative" and "circular"

Details

The boundary argument can have the following different values: "nothing" meaning do nothing special with boundaries; "conservative" meaning force columns of matrix to sum to 1; and "circular" meaning wrap edges.

Value

A square connectivity matrix

Author(s)

David M. Kaplan <dmkaplan2000@gmail.com>

References

Kaplan, D. M., Botsford, L. W., and Jorgensen, S. 2006. Dispersal per recruit: An efficient method for assessing sustainability in marine reserve networks. *Ecological Applications*, 16: 2248-2263.

See Also

See also [DispersalPerRecruitModel](#)

Examples

```
library(ConnMatTools)
cm <- laplacianConnMat(100,10,15,"circular")
image(cm)
```

localRetention	<i>Local retention of a connectivity matrix</i>
----------------	---

Description

Local retention is defined as the diagonal elements of the connectivity matrix.

Usage

```
localRetention(conn.mat)
```

Arguments

conn.mat A square connectivity matrix.

Author(s)

David M. Kaplan <dmkaplan2000@gmail.com>

Examples

```
library(ConnMatTools)
data(chile.loco)

sr <- selfRecruitment(chile.loco)
lr <- localRetention(chile.loco)
rlr <- relativeLocalRetention(chile.loco)
```

mergeSubpops	<i>Merge subpopulations</i>
--------------	-----------------------------

Description

This function tries to merge random subpopulations, checking if the result is a better solution to the minimization problem.

Usage

```
mergeSubpops(subpops.lst, conn.mat, beta)
```

Arguments

subpops.lst	A list whose elements are vectors of indices for each subpopulation. See subpopsVectorToList .
conn.mat	A square connectivity matrix. This matrix has typically been normalized and made symmetric prior to using this function.
beta	Controls degree of splitting of connectivity matrix, with larger values generating more subpopulations.

Value

List of the same format as subpops.lst, but with potentially fewer subpopulations.

Author(s)

David M. Kaplan <dmkaplan2000@gmail.com>

References

Jacobi, M. N., André, C., Döös, K., and Jonsson, P. R. 2012. Identification of subpopulations from connectivity matrices. *Ecography*, 35: 1004-1016.

See Also

See also [optimalSplitConnMat](#),

optimalSplitConnMat *Iteratively, optimally split a connectivity matrix*

Description

Algorithm for iteratively determining subpopulations of highly-connected sites. Uses an iterative method described in Jacobi et al. (2012)

Usage

```
optimalSplitConnMat(conn.mat, normalize.cols = TRUE,
  make.symmetric = "mean", remove.diagonal = FALSE, cycles = 2,
  betas = betasVectorDefault(ifelse(normalize.cols, dim(conn.mat)[2],
  prod(dim(conn.mat))/sum(conn.mat))), steps), steps = 10, ...)
```

Arguments

conn.mat	A square connectivity matrix.
normalize.cols	A boolean indicating if columns of conn.mat should be normalized by the sum of all elements in the column. Defaults to TRUE.
make.symmetric	A string indicating how to force conn.mat to be symmetric. "mean" (the default) will replace C_{ij} by $(C_{ij} + C_{ji})/2$. "max" will replace C_{ij} by the maximum of C_{ij} and C_{ji} .
remove.diagonal	A boolean indicating if the diagonal elements of conn.mat should be removed before determining the subpopulations. Defaults to FALSE.
cycles	Number of times to pass over values in betas.
betas	Vector of beta values to try. If not given, will default to <code>betasVectorDefault(dim(conn.mat)[2], steps)</code> .
steps	Number of beta values to produce using betasVectorDefault. Ignored if betas argument is explicitly given.
...	further arguments to be passed to <code>splitConnMat</code>

Value

A list with the following elements:

betas	Vector of all beta values tested
num.subpops	Vector of number of subpopulations found for each value of beta
qualities	Vector of the quality statistic for each subpopulation division
subpops	A matrix with dimensions $\text{dim}(\text{conn.mat})[2] \times \text{length}(\text{betas})$ indicating which subpopulation each site belongs to
best.splits	A list indicating for each number of subpopulations, which column of subpops contains the division with the lowest quality statistic. E.g., <code>best.splits[["4"]]</code> \$index contains the column index of the optimal division of the connectivity matrix into 4 subpopulations.

Note

In Jacobi et al. (2012) paper, the connectivity matrix is oriented so that C_{ij} is dispersal from i to j , whereas in this R package, the connectivity matrix is oriented so that C_{ij} is dispersal from j to i . This choice of orientation is arbitrary, but one must always be consistent. From j to i is more common in population dynamics because it works well with matrix multiplication (e.g., `settlers = conn.mat %*% eggs`).

Author(s)

David M. Kaplan <dmkaplan2000@gmail.com>

References

Jacobi, M. N., André, C., Döös, K., and Jonsson, P. R. 2012. Identification of subpopulations from connectivity matrices. *Ecography*, 35: 1004-1016.

See Also

See also [splitConnMat](#), [recSplitConnMat](#), [mergeSubpops](#), [qualitySubpops](#)

Examples

```
library(ConnMatTools)
data(chile.loco)

num <- prod(dim(chile.loco)) / sum(chile.loco)
betas <- betasVectorDefault(n=num, steps=4)
chile.loco.split <- optimalSplitConnMat(chile.loco, normalize.cols=FALSE,
                                       betas=betas)

# Extra 3rd division
print(paste("Examining split with", names(chile.loco.split$best.splits)[1],
           "subpopulations."))
pops <- subpopsVectorToList(chile.loco.split$subpops[, chile.loco.split$best.splits[[1]]$index])

reduce.loco <- reducedConnMat(pops, chile.loco)

sr <- selfRecruitment(reduce.loco)
lr <- localRetention(reduce.loco)
rlr <- relativeLocalRetention(reduce.loco)
```

protectedAreaSelection

Function to select optimal network of protected areas based on connectivity

Description

This function finds the optimal network of protected areas based on connectivity using the eigenvalue perturbation approach described in Nilsson Jacobi & Jonsson (2011).

Usage

```
protectedAreaSelection(conn.mat, nev = dim(conn.mat)[1], delta = 0.1,
  theta = 0.05, M = 20, epsilon.lambda = 1e-04, epsilon.uv = 0.05,
  only.list = T, ...)
```

Arguments

conn.mat	a square connectivity matrix.
nev	number of eigenvalues and associated eigenvectors to be calculated.
delta	the effect of protecting site <i>i</i> (e.g. increase in survival or fecundity in protected areas relative to unprotected areas). Now a single value, in future it will be possible to specify site-specific values. The perturbation theory used in the construction of the algorithm assumes delta to be small (e.g. delta=0.1). However, higher values give also good results.
theta	the threshold of donor times recipient value that a site must have to be selected.
M	the maximal number of sites selected from each subpopulation even if there are more sites above the threshold theta
epsilon.lambda	Threshold for removing complex eigenvalues.
epsilon.uv	Threshold for removing eigenvectors with elements of opposite signs of comparable magnitude.
only.list	Logical, whether the function return only the list of selected sites or also the predicted impact of each selected site on the eigenvalues
...	Additional arguments for the <code>eigs</code> function.

Value

If only.list is TRUE, just returns the list of selected sites. If FALSE, then result will be a list containing selected sites and predicted impact of each selected site on the eigenvalues.

Author(s)

Marco Andrello <marco.andrello@gmail.com>

References

Jacobi, M. N., and Jonsson, P. R. 2011. Optimal networks of nature reserves can be found through eigenvalue perturbation theory of the connectivity matrix. *Ecological Applications*, 21: 1861–1870.

qualitySubpops	<i>Quality measure for subpopulation division</i>
----------------	---

Description

A measure of the leakage between subpopulations for a given division of the connectivity matrix into subpopulations. This statistic is equal to $1 - \text{mean}(\text{RLR})$ of the reduced connectivity matrix, where RLR=relative local retention ([relativeLocalRetention](#)), i.e., the fraction of settling individuals that originated at their site of settlement.

Usage

```
qualitySubpops(subpops.lst, conn.mat)
```

Arguments

subpops.lst	A list whose elements are vectors of indices for each subpopulation. If a vector of integers is given, then subpopsVectorToList is applied to convert it to a list of subpopulations.
conn.mat	A square connectivity matrix.

Value

The quality statistic.

A smaller value of the quality statistic indicates less leakage.

Author(s)

David M. Kaplan <dmkaplan2000@gmail.com>

References

Jacobi, M. N., André, C., Döös, K., and Jonsson, P. R. 2012. Identification of subpopulations from connectivity matrices. *Ecography*, 35: 1004-1016.

See Also

See also [optimalSplitConnMat](#), [subpopsVectorToList](#), [relativeLocalRetention](#)

recSplitConnMat *Recursively subdivides a set of subpopulations*

Description

This function recursively splits each subpopulation of a list of subpopulations until none of the subpopulations can be split further to improve the minimization.

Usage

```
recSplitConnMat(subpops.lst, conn.mat, beta, ...)
```

Arguments

subpops.lst	A list whose elements are vectors of indices for each subpopulation. See subpopsVectorToList .
conn.mat	A square connectivity matrix. This matrix has typically been normalized and made symmetric prior to using this function.
beta	Controls degree of splitting of connectivity matrix, with larger values generating more subpopulations.
...	further arguments to be passed to splitConnMat

Author(s)

David M. Kaplan <dmkaplan2000@gmail.com>

References

Jacobi, M. N., André, C., Döös, K., and Jonsson, P. R. 2012. Identification of subpopulations from connectivity matrices. *Ecography*, 35: 1004-1016.

See Also

See also [optimalSplitConnMat](#), [splitConnMat](#), [subpopsVectorToList](#)

reducedConnMat *Reduced connectivity matrix according to a set of subpopulations*

Description

Reduces a connectivity matrix based on a set of subpopulations. If there are N subpopulations, then the reduced matrix will have dimensions NxN. The reduced matrix will be ordered according to the order of subpopulations in subpops.lst.

Usage

```
reducedConnMat(subpops.lst, conn.mat)
```

Arguments

<code>subpops.lst</code>	A list whose elements are vectors of indices for each subpopulation. If a vector of integers is given, then subpopsVectorToList is applied to convert it to a list of subpopulations.
<code>conn.mat</code>	A square connectivity matrix.

Value

A reduced connectivity matrix. The sum of all elements of this reduced connectivity matrix will be equal to the sum of all elements of the original connectivity matrix.

Author(s)

David M. Kaplan <dmkaplan2000@gmail.com>

References

Jacobi, M. N., André, C., Döös, K., and Jonsson, P. R. 2012. Identification of subpopulations from connectivity matrices. *Ecography*, 35: 1004-1016.

See Also

See also [qualitySubpops](#)

Examples

```
library(ConnMatTools)
data(chile.loco)

num <- prod(dim(chile.loco)) / sum(chile.loco)
betas <- betasVectorDefault(n=num, steps=4)
chile.loco.split <- optimalSplitConnMat(chile.loco, normalize.cols=FALSE,
                                       betas=betas)

# Extra 3rd division
print(paste("Examining split with", names(chile.loco.split$best.splits)[1],
           "subpopulations."))
pops <- subpopsVectorToList(chile.loco.split$subpops[, chile.loco.split$best.splits[[1]]$index])

reduce.loco <- reducedConnMat(pops, chile.loco)

sr <- selfRecruitment(reduce.loco)
lr <- localRetention(reduce.loco)
rlr <- relativeLocalRetention(reduce.loco)
```

`relativeLocalRetention`*Relative local retention of a connectivity matrix*

Description

Relative local retention is defined as the diagonal elements of the connectivity matrix divided by the sum of the corresponding column of the connectivity matrix.

Usage

```
relativeLocalRetention(conn.mat)
```

Arguments

`conn.mat` A square connectivity matrix.

Author(s)

David M. Kaplan <dmkaplan2000@gmail.com>

Examples

```
library(ConnMatTools)
data(chile.loco)

sr <- selfRecruitment(chile.loco)
lr <- localRetention(chile.loco)
rlr <- relativeLocalRetention(chile.loco)
```

`selfRecruitment`*Self recruitment of a connectivity matrix*

Description

If egg production is uniform over sites, then self recruitment is defined as the diagonal elements of the connectivity matrix divided by the sum of the corresponding row of the connectivity matrix. If not, then the elements of the dispersal matrix must be weighted by the number of eggs produced.

Usage

```
selfRecruitment(conn.mat, eggs = NULL)
```

Arguments

conn.mat	A square connectivity matrix.
eggs	A vector of egg production values for each site. Defaults to NULL, equivalent to assuming all sites have equal egg production.

Author(s)

David M. Kaplan <dmkaplan2000@gmail.com>

Examples

```
library(ConnMatTools)
data(chile.loco)

sr <- selfRecruitment(chile.loco)
lr <- localRetention(chile.loco)
rlr <- relativeLocalRetention(chile.loco)
```

settlerRecruitSlopeCorrection

Correction for slope of settler-recruit relationship

Description

This function corrects the slope of the settler-recruit curve so that the collapse point of the spatially-explicit population model corresponding to the connectivity matrix agrees with that of the global non-spatially-explicit model. Uses the method in White (2010).

Usage

```
settlerRecruitSlopeCorrection(conn.mat, slope = 1, natural.LEP = 1,
  critical.FLEP = 0.35, use.arpack = TRUE)
```

Arguments

conn.mat	a square connectivity matrix.
slope	slope at the origin of the settler-recruit relationship. Only interesting to fix this argument if it is a vector of length = $\dim(\text{conn.mat})[2]$ (i.e., if the slope varies among sites and one wants to globally scale all slopes so that the collapse point matches the global collapse point).
natural.LEP	value of lifetime-egg-production (LEP), also known as eggs-per-recruit, in the absence of fishing. Can be a vector of length = $\dim(\text{conn.mat})[2]$. Defaults to 1.
critical.FLEP	Fraction of natural.LEP at which collapse occurs. Defaults to 0.35.
use.arpack	Boolean determining if calculation is to be done with arpack function from the igraph package. This is much quicker for large matrices, but requires igraph . Defaults to TRUE, but will use eigen instead if igraph is not found.

Value

The slope argument corrected so that collapse happens when LEP is critical. $FLEP * natural.LEP$.

Author(s)

David M. Kaplan <dmkaplan2000@gmail.com>

References

White, J. W. 2010. Adapting the steepness parameter from stock-recruit curves for use in spatially explicit models. *Fisheries Research*, 102: 330-334.

See Also

See also [eigs](#), [arpack](#)

splitConnMat	<i>Split connectivity matrix into subpopulations</i>
--------------	--

Description

This function tries to optimally split a given subpopulation into two smaller subpopulations.

Usage

```
splitConnMat(indices, conn.mat, beta, tries = 5, threshold = 1e-10,
             alpha = 0.1, maxit = 500)
```

Arguments

indices	vector of indices of sites in a subpopulation
conn.mat	a square connectivity matrix. This matrix has typically been normalized and made symmetric prior to using this function.
beta	controls degree of splitting of connectivity matrix, with larger values generating more subpopulations.
tries	how many times to restart the optimization algorithm. Defaults to 5.
threshold	controls when to stop each "try". Defaults to 1e-10.
alpha	controls rate of convergence to solution
maxit	Maximum number of iterations to perform per "try".

Value

List with one or two elements, each containing a vector of indices of sites in a subpopulations

Author(s)

David M. Kaplan <dmkaplan2000@gmail.com>

References

Jacobi, M. N., André, C., Döös, K., and Jonsson, P. R. 2012. Identification of subpopulations from connectivity matrices. *Ecography*, 35: 1004-1016.

See Also

See also [optimalSplitConnMat](#), [recSplitConnMat](#), [subpopsVectorToList](#)

subpopsVectorToList *Convert subpopulation vector to a list of indices*

Description

A helper function to convert a vector of subpopulation identifications into a list appropriate for [recSplitConnMat](#), [qualitySubpops](#), etc.

Usage

```
subpopsVectorToList(x)
```

Arguments

x vector of subpopulation identifications

Details

Note that subpopulations list will be ordered according to the numerical order of the subpopulation indices in the original matrix, which will not necessarily be that of the spatial order of sites in the original connectivity matrix.

Value

A list where each element is a vector of indices for a given subpopulation.

Author(s)

David M. Kaplan <dmkaplan2000@gmail.com>

See Also

See also [recSplitConnMat](#), [qualitySubpops](#)

Index

*Topic **data**

chile.loco, 4

*Topic **package**

ConnMatTools, 4

arpack, 8, 9, 20, 21

betasVectorDefault, 2, 13

BevertonHolt, 3, 6, 8

chile.loco, 4

ConnMatTools, 4

ConnMatTools-package (ConnMatTools), 4

DispersalPerRecruitModel, 5, 8, 11

DPRHomerangeGravity, 7

eigen, 8

eigs, 8, 15, 21

hockeyStick, 5–8, 9

igraph, 8, 9, 20

laplacianConnMat, 10

localRetention, 11

mergeSubpops, 12, 14

optimalSplitConnMat, 3, 5, 12, 13, 16, 17, 22

protectedAreaSelection, 14

qualitySubpops, 14, 16, 18, 22

recSplitConnMat, 14, 17, 22

reducedConnMat, 17

relativeLocalRetention, 16, 19

selfRecruitment, 19

settlerRecruitSlopeCorrection, 20

splitConnMat, 13, 14, 17, 21

subpopsVectorToList, 12, 16–18, 22, 22