

# Package: ConnMatTools (via r-universe)

August 28, 2024

**Type** Package

**Title** Tools for Working with Connectivity Data

**Version** 0.3.5

**Date** 2017-02-03

**Author** David M. Kaplan <dmkaplan2000@gmail.com> [cre,aut], Marco Andrello <marco.andrello@gmail.com> [ctb]

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

**Description** Collects several different methods for analyzing and working with connectivity data 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, mcmc

**Collate** 'ConnMatTools.R' 'eigs.R' 'Jacobi\_EPT\_reserve\_selection.R' 'utils.R' 'connectivity\_estimation.R' 'connectivity\_estimation.distributions.R' 'connectivity\_estimation.finite\_settlement.R' 'connectivity\_estimation.multinomial.R' 'connectivity\_estimation.multiple.R' 'datasets.R' 'dpr\_model.R' 'estimating\_marked\_egg\_fraction.R' 'retentionStats.R' 'jacobi\_etal\_2012.R'

**License** GPL (>= 2)

**RoxygenNote** 7.0.2

**Repository** <https://dmkaplan2000.r-universe.dev>

**RemoteUrl** <https://github.com/dmkaplan2000/connmattools>

**RemoteRef** HEAD

**RemoteSha** 8c6627ad976713994d5e41608f7d7c4dbe094259

## Contents

betasVectorDefault	2
BevertonHolt	3
chile.loco	4
ConnMatTools	5
d.mix.dists.func	6
d.rel.conn.beta.prior	7
d.rel.conn.dists.func	10
d.rel.conn.finite.settlement	13
d.rel.conn.multinomial.unnorm	15
d.rel.conn.multiple	18
d.rel.conn.unif.prior	21
damselish.lods	23
DispersalPerRecruitModel	24
DPRHomerangeGravity	26
dual.mark.transmission	28
eigs	30
gammaParamsConvert	31
hockeyStick	32
laplacianConnMat	33
localRetention	34
mergeSubpops	34
optim.rel.conn.dists	35
optimalSplitConnMat	37
prob.marked	39
protectedAreaSelection	40
qualitySubpops	41
r.marked.egg.fraction	42
recSplitConnMat	45
reducedConnMat	46
relativeLocalRetention	47
selfRecruitment	48
settlerRecruitSlopeCorrection	49
splitConnMat	50
stepfun.hist	51
subpopsVectorToList	52
<b>Index</b>	<b>54</b>

---

betasVectorDefault	<i>Compute vector of beta values</i>
--------------------	--------------------------------------

---

### 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 <code>conn.mat</code> ).
steps	number of beta values to return. Defaults to 10.
cycles	how many cycles of $2\pi$ 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., Andre, C., Doos, 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):  $\text{slope} * \text{settlers} / (1 + \text{slope} * \text{settlers} / R_{\text{max}})$

**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.

**Format**

A square 89x89 matrix with real, positive elements.

**Author(s)**

David M. Kaplan <dmkaplan2000@gmail.com>

**References**

Garavelli L, Kaplan DM, Colas F, Stotz W, Yannicelli B, Lett C (2014) Identifying appropriate spatial scales for marine conservation and management using a larval dispersal model: The case of *Concholepas concholepas* (loco) in Chile. *Progress in Oceanography* 124:42-53. doi:10.1016/j.pocean.2014.03.011

---

ConnMatTools

*Tools for working with connectivity matrices*

---

## Description

Tools for Working with Connectivity Data

## Details

Collects several different methods for analyzing and working with connectivity data 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: 0.3.5  
Date: 2017-02-02  
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.
- Kaplan DM, Cuif M, Fauvelot C, Vigliola L, Nguyen-Huu T, Tiavouane J and Lett C (in press) Uncertainty in empirical estimates of marine larval connectivity. *ICES Journal of Marine Science*. doi:10.1093/icesjms/fsw182.
- 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](#), [d.rel.conn.beta.prior](#)

**Examples**

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

---

d.mix.dists.func	<i>Returns probability density function (PDF) for a mix of marked and unmarked individuals</i>
------------------	--

---

**Description**

This function returns a probability density function (PDF) for scores for a mix of marked and unmarked individuals with known fraction of marked individuals. The distributions for marked individuals and for unmarked individuals must be known.

**Usage**

```
d.mix.dists.func(d.unmarked, d.marked)
```

**Arguments**

d.unmarked	A function representing the PDF of unmarked individuals. Must be normalized so that it integrates to 1 for the function to work properly.
d.marked	A function representing the PDF of marked individuals. Must be normalized so that it integrates to 1 for the function to work properly.

**Value**

A function representing the PDF of observations drawn from the mixed distribution of marked and unmarked individuals. The function takes two arguments: p.marked, the fraction of marked individuals in the distribution; and obs, a vector of observed score values.

**Author(s)**

David M. Kaplan <dmkaplan2000@gmail.com>

**References**

Kaplan DM, Cuif M, Fauvelot C, Vigliola L, Nguyen-Huu T, Tiavouane J and Lett C (in press) Uncertainty in empirical estimates of marine larval connectivity. ICES Journal of Marine Science. doi:10.1093/icesjms/fsw182.

**See Also**

See also [d.rel.conn.dists.func](#), [optim.rel.conn.dists](#).

---

d.rel.conn.beta.prior *Estimate the probability distribution of relative connectivity values assuming a Beta-distributed prior*

---

### Description

These functions calculate the probability density function (`d.rel.conn.beta.prior`), the probability distribution function (aka the cumulative distribution function; `p.rel.conn.beta.prior`) and the quantile function (`q.rel.conn.beta.prior`) for the relative (to all settlers at the destination site) connectivity value for larval transport between a source and destination site given a known fraction of marked individuals (i.e., eggs) in the source population. A non-uniform prior is used for the relative connectivity value.

### Usage

```
d.rel.conn.beta.prior(  
  phi,  
  p,  
  k,  
  n,  
  prior.shape1 = 0.5,  
  prior.shape2 = prior.shape1,  
  prior.func = function(phi) dbeta(phi, prior.shape1, prior.shape2),  
  ...  
)  
  
p.rel.conn.beta.prior(  
  phi,  
  p,  
  k,  
  n,  
  prior.shape1 = 0.5,  
  prior.shape2 = prior.shape1,  
  prior.func = function(phi) dbeta(phi, prior.shape1, prior.shape2),  
  ...  
)  
  
q.rel.conn.beta.prior.func(  
  p,  
  k,  
  n,  
  prior.shape1 = 0.5,  
  prior.shape2 = prior.shape1,  
  prior.func = function(phi) dbeta(phi, prior.shape1, prior.shape2),  
  N = 1000,  
  ...  
)
```

```

q.rel.conn.beta.prior(
  q,
  p,
  k,
  n,
  prior.shape1 = 0.5,
  prior.shape2 = prior.shape1,
  prior.func = function(phi) dbeta(phi, prior.shape1, prior.shape2),
  N = 1000,
  ...
)

```

### Arguments

<code>phi</code>	Vector of fractions of individuals (i.e., eggs) from the source population settling at the destination population
<code>p</code>	Fraction of individuals (i.e., eggs) marked in the source population
<code>k</code>	Number of marked settlers found in sample
<code>n</code>	Total number of settlers collected
<code>prior.shape1</code>	First shape parameter for Beta distributed prior. Defaults to 0.5.
<code>prior.shape2</code>	Second shape parameter for Beta distributed prior. Defaults to being the same as <code>prior.shape1</code> .
<code>prior.func</code>	Function for prior distribution. Should take one parameter, <code>phi</code> , and return a probability. Defaults to <code>function(phi) dbeta(phi, prior.shape1, prior.shape2)</code> . If this is specified, then inputs <code>prior.shape1</code> and <code>prior.shape2</code> are ignored.
<code>...</code>	Extra arguments for the <a href="#">integrate</a> function used for normalization of probability distributions.
<code>N</code>	Number of points at which to estimate cumulative probability function for reverse approximation of quantile distribution. Defaults to 1000.
<code>q</code>	Vector of quantiles

### Details

The prior distribution for relative connectivity `phi` defaults to a Beta distribution with both shape parameters equal to 0.5. This is the Reference or Jeffreys prior for a binomial distribution parameter. Both shape parameters equal to 1 corresponds to a uniform prior.

Estimations of the probability distribution are based on numerical integration using the [integrate](#) function, and therefore are accurate to the level of that function. Some modification of the default arguments to that function may be necessary to achieve good results for certain parameter values.

### Value

Vector of probabilities or quantiles, or a function in the case of `q.rel.conn.beta.prior.func`.



**Functions**

- `d.rel.conn.beta.prior`: Returns the probability density for relative connectivity between a pair of sites
- `p.rel.conn.beta.prior`: Returns the cumulative probability distribution for relative connectivity between a pair of sites
- `q.rel.conn.beta.prior.func`: Returns a function to estimate quantiles for the probability distribution function for relative connectivity between a pair of sites.
- `q.rel.conn.beta.prior`: Estimates quantiles for the probability distribution function for relative connectivity between a pair of sites

**Author(s)**

David M. Kaplan <dmkaplan2000@gmail.com>

**References**

Kaplan DM, Cuif M, Fauvelot C, Vigliola L, Nguyen-Huu T, Tiavouane J and Lett C (in press) Uncertainty in empirical estimates of marine larval connectivity. ICES Journal of Marine Science. doi:10.1093/icesjms/fsw182.

**See Also**

Other connectivity estimation: [d.rel.conn.dists.func\(\)](#), [d.rel.conn.finite.settlement\(\)](#), [d.rel.conn.multinomial.unnorm\(\)](#), [d.rel.conn.multiple\(\)](#), [d.rel.conn.unif.prior\(\)](#), [dual.mark.transmission.optim.rel.conn.dists\(\)](#), [r.marked.egg.fraction\(\)](#)

**Examples**

```
library(ConnMatTools)

k <- 10 # Number of marked settlers among sample
n.obs <- 87 # Number of settlers in sample

p <- 0.4 # Fraction of eggs that was marked
phi <- seq(0.001,1-0.001,length.out=101) # Values for relative connectivity

# Probability distribution assuming infinite settler pool and uniform prior
drc <- d.rel.conn.unif.prior(phi,p,k,n.obs)
qrc <- q.rel.conn.unif.prior(c(0.025,0.975),p,k,n.obs) # 95% confidence interval

# Probability distribution assuming infinite settler pool and using reference/Jeffreys prior
drp <- d.rel.conn.beta.prior(phi,p,k,n.obs)
prp <- p.rel.conn.beta.prior(phi,p,k,n.obs)
qrp <- q.rel.conn.beta.prior(c(0.025,0.975),p,k,n.obs) # 95% confidence interval

# Make a plot of different distributions
# black = Jeffreys prior; red = uniform prior
# Jeffreys prior draws distribution slightly towards zero
plot(phi,drp,type="l",main="Probability of relative connectivity values",
      xlab=expression(phi),ylab="Probability density")
```

```
lines(phi,drc,col="red")
abline(v=qrp,col="black",lty="dashed")
abline(v=qrc,col="red",lty="dashed")
```

---

d.rel.conn.dists.func *Functions for estimating the probability distribution for relative connectivity given a pair of distributions for scores for marked and unmarked individuals*

---

## Description

These functions return functions that calculate the probability density function (d.rel.conn.dists.func), the probability distribution function (aka the cumulative distribution function; p.rel.conn.dists.func) and the quantile function (q.rel.conn.dists.func) for relative connectivity given a set of observed score values, distributions for unmarked and marked individuals, and an estimate of the fraction of all eggs marked at the source site, p.

## Usage

```
d.rel.conn.dists.func(
  obs,
  d.unmarked,
  d.marked,
  p = 1,
  N = max(100, min(5000, 2 * length(obs))),
  prior.shape1 = 0.5,
  prior.shape2 = prior.shape1,
  prior.func = function(phi) dbeta(phi, prior.shape1, prior.shape2),
  ...
)
```

```
p.rel.conn.dists.func(
  obs,
  d.unmarked,
  d.marked,
  p = 1,
  N = max(100, min(5000, 2 * length(obs))),
  prior.shape1 = 0.5,
  prior.shape2 = prior.shape1,
  prior.func = function(phi) dbeta(phi, prior.shape1, prior.shape2),
  ...
)
```

```
q.rel.conn.dists.func(
  obs,
  d.unmarked,
  d.marked,
```

```

    p = 1,
    N = max(100, min(5000, 2 * length(obs))),
    prior.shape1 = 0.5,
    prior.shape2 = prior.shape1,
    prior.func = function(phi) dbeta(phi, prior.shape1, prior.shape2),
    ...
  )

```

### Arguments

obs	Vector of observed score values for potentially marked individuals
d.unmarked	A function representing the PDF of unmarked individuals. Must be normalized so that it integrates to 1 for the function to work properly.
d.marked	A function representing the PDF of marked individuals. Must be normalized so that it integrates to 1 for the function to work properly.
p	Fraction of individuals (i.e., eggs) marked in the source population. Defaults to 1.
N	number of steps between 0 and 1 at which to approximate likelihood function as input to <a href="#">approxfun</a> . Defaults to 2*length(obs) so long as that number is comprised between 100 and 5000.
prior.shape1	First shape parameter for Beta distributed prior. Defaults to 0.5.
prior.shape2	Second shape parameter for Beta distributed prior. Defaults to being the same as prior.shape1.
prior.func	Function for prior distribution. Should take one parameter, phi, and return a probability. Defaults to function(phi) dbeta(phi, prior.shape1, prior.shape2). If this is specified, then inputs prior.shape1 and prior.shape2 are ignored.
...	Additional arguments for the <a href="#">integrate</a> function.

### Details

The normalization of the probability distribution is carried out using a simple, fixed-step trapezoidal integration scheme. By default, the number of steps between relative connectivity values of 0 and 1 defaults to 2\*length(obs) so long as that number is comprised between 100 and 5000.

### Value

A function that takes one argument (the relative connectivity for d.rel.conn.dists.func and p.rel.conn.dists.func; the quantile for q.rel.conn.dists.func) and returns the probability density, cumulative probability or score value, respectively. The returned function accepts both vector and scalar input values.

### Functions

- d.rel.conn.dists.func: Returns a function that is PDF for relative connectivity
- p.rel.conn.dists.func: Returns a function that is cumulative probability distribution for relative connectivity
- q.rel.conn.dists.func: Returns a function that is quantile function for relative connectivity

**Author(s)**

David M. Kaplan <dmkaplan2000@gmail.com>

**References**

Kaplan DM, Cuif M, Fauvelot C, Vigliola L, Nguyen-Huu T, Tiavouane J and Lett C (in press) Uncertainty in empirical estimates of marine larval connectivity. ICES Journal of Marine Science. doi:10.1093/icesjms/fsw182.

**See Also**

Other connectivity estimation: [d.rel.conn.beta.prior\(\)](#), [d.rel.conn.finite.settlement\(\)](#), [d.rel.conn.multinomial.unnorm\(\)](#), [d.rel.conn.multiple\(\)](#), [d.rel.conn.unif.prior\(\)](#), [dual.mark.transmission.optim.rel.conn.dists\(\)](#), [r.marked.egg.fraction\(\)](#)

**Examples**

```
library(ConnMatTools)
data(damselfish.lods)

# Histograms of simulated LODs
l <- seq(-1,30,0.5)
h.in <- hist(damselfish.lods$in.group,breaks=1)
h.out <- hist(damselfish.lods$out.group,breaks=1)

# PDFs for marked and unmarked individuals based on simulations
d.marked <- stepfun.hist(h.in)
d.unmarked <- stepfun.hist(h.out)

# Fraction of adults genotyped at source site
p.adults <- 0.25

# prior.shape1=1 # Uniform prior
prior.shape1=0.5 # Jeffreys prior

# Fraction of eggs from one or more genotyped parents
p <- dual.mark.transmission(p.adults)$p

# PDF for relative connectivity
D <- d.rel.conn.dists.func(damselfish.lods$real.children,
                          d.unmarked,d.marked,p,
                          prior.shape1=prior.shape1)

# Estimate most probable value for relative connectivity
phi.mx <- optim.rel.conn.dists(damselfish.lods$real.children,
                              d.unmarked,d.marked,p)$phi

# Estimate 95% confidence interval for relative connectivity
Q <- q.rel.conn.dists.func(damselfish.lods$real.children,
                          d.unmarked,d.marked,p,
                          prior.shape1=prior.shape1)
```

```

# Plot it up
phi <- seq(0,1,0.001)
plot(phi,D(phi),type="l",
      xlim=c(0,0.1),
      main="PDF for relative connectivity",
      xlab=expression(phi),
      ylab="Probability density")

abline(v=phi.mx,col="green",lty="dashed")
abline(v=Q(c(0.025,0.975)),col="red",lty="dashed")

```

---

```
d.rel.conn.finite.settlement
```

*Estimate the probability distribution for the number of settlers originating at a site given a sample from a finite settler pool*

---

### Description

These functions calculate the probability mass function (`d.rel.conn.finite.settlement`), the cumulative distribution function (`p.rel.conn.finite.settlement`) and the quantile function (`q.rel.conn.finite.settlement`) for the true number of settlers at a site that originated in a particular site given a known fraction of marked eggs among the eggs originating at the source site, a sample of settlers at the destination site, a known fraction of which are marked, and a finite settler pool of known size.

### Usage

```

d.rel.conn.finite.settlement(
  n.origin,
  p,
  k,
  n.obs,
  n.settlers,
  prior.n.origin = 1
)

```

```

p.rel.conn.finite.settlement(
  n.origin,
  p,
  k,
  n.obs,
  n.settlers,
  prior.n.origin = 1
)

```

```

q.rel.conn.finite.settlement(q, p, k, n.obs, n.settlers, prior.n.origin = 1)

```

**Arguments**

<code>n.origin</code>	Vector of integers of possible numbers of settlers in the cohort that originated at the site of marking. All values should be integers $\leq n.settlers$ .
<code>p</code>	Fraction of individuals (i.e., eggs) marked in the source population
<code>k</code>	Number of marked settlers in sample
<code>n.obs</code>	Total number of settlers collected
<code>n.settlers</code>	Total number of settlers at the destination site from which the <code>n.obs</code> ( $\leq n.settlers$ ) settlers are collected
<code>prior.n.origin</code>	A prior probability mass function for the number of settlers in the cohort originating at the site of marking. Must be a scalar or a vector of length <code>n.settlers+1</code> . Defaults to 1.
<code>q</code>	Vector of quantiles

**Details**

The relative connectivity between the source and destination sites is calculated as `n.origin/n.settlers`.

**Value**

A vector of probabilities or quantiles.

**Functions**

- `d.rel.conn.finite.settlement`: Returns the probability mass function for the numbers of settlers in the cohort that originated at the source site (i.e., site of marking).
- `p.rel.conn.finite.settlement`: Returns the cumulative distribution function for the numbers of settlers in the cohort that originated at the source site (i.e., site of marking).
- `q.rel.conn.finite.settlement`: Returns quantiles of the cumulative distribution function for the numbers of settlers in the cohort that originated at the source site (i.e., site of marking).

**Author(s)**

David M. Kaplan <dmkaplan2000@gmail.com>

**References**

Kaplan DM, Cuif M, Fauvelot C, Vigliola L, Nguyen-Huu T, Tiavouane J and Lett C (in press) Uncertainty in empirical estimates of marine larval connectivity. ICES Journal of Marine Science. doi:10.1093/icesjms/fsw182.

**See Also**

Other connectivity estimation: `d.rel.conn.beta.prior()`, `d.rel.conn.dists.func()`, `d.rel.conn.multinomial.unnc`, `d.rel.conn.multiple()`, `d.rel.conn.unif.prior()`, `dual.mark.transmission()`, `optim.rel.conn.dists()`, `r.marked.egg.fraction()`

**Examples**

```

library(ConnMatTools)

k <- 10 # Number of marked settlers among sample
n.obs <- 87 # Number of settlers in sample
n.settlers <- 100 # Total size of settler pool

p <- 0.4 # Fraction of eggs that was marked
phi <- seq(0,1,length.out=101) # Values for relative connectivity

# Probability distribution assuming infinite settler pool and uniform prior
drc <- d.rel.conn.unif.prior(phi,p,k,n.obs)
prc <- p.rel.conn.unif.prior(phi,p,k,n.obs)
qrc <- q.rel.conn.unif.prior(c(0.025,0.975),p,k,n.obs) # 95% confidence interval

# Test with finite settlement function and large (approx. infinite) settler pool
# Can be a bit slow for large settler pools
dis <- d.rel.conn.finite.settlement(0:(7*n.obs),p,k,n.obs,7*n.obs)

# Quantiles
qis <- q.rel.conn.finite.settlement(c(0.025,0.975),p,k,n.obs,7*n.obs)

# Finite settler pool
dfs <- d.rel.conn.finite.settlement(0:n.settlers,p,k,n.obs,n.settlers)

# Quantiles for the finite settler pool
qfs <- q.rel.conn.finite.settlement(c(0.025,0.975),p,k,n.obs,n.settlers)

# Make a plot of different distributions
plot(phi,drc,type="l",main="Probability of relative connectivity values",
      xlab=expression(phi),ylab="Probability density")
lines(phi,prc,col="blue")
lines((0:(7*n.obs))/(7*n.obs),dis*(7*n.obs),col="black",lty="dashed")
lines((0:n.settlers)/n.settlers,dfs*n.settlers,col="red",lty="dashed")
abline(v=qrc,col="black")
abline(v=qis/(7*n.obs),col="black",lty="dashed")
abline(v=qfs/n.settlers,col="red",lty="dashed")

```

---

```
d.rel.conn.multinomial.unnorm
```

*Calculates unnormalized probability density for relative connectivity values from multiple distinct sites*

---

**Description**

This functions calculates the unnormalized probability density function for the relative (to all settlers at the destination site) connectivity value for larval transport between multiple source sites to a destination site. An arbitrary number of source sites can be evaluated.

**Usage**

```
d.rel.conn.multinomial.unnorm(
  phis,
  ps,
  ks,
  n.sample,
  log = FALSE,
  dirichlet.prior.alphas = 1/(length(phis) + 1)
)
```

**Arguments**

<code>phis</code>	Vector of fractions of individuals (i.e., eggs) from the source populations settling at the destination population
<code>ps</code>	Vector of fractions of individuals (i.e., eggs) marked in each of the source populations
<code>ks</code>	Vector of numbers of marked settlers from each source population found in the sample
<code>n.sample</code>	Vector of total numbers of settlers collected
<code>log</code>	Boolean indicating whether or not to return the log probability density. Defaults to FALSE.
<code>dirichlet.prior.alphas</code>	Parameter value for a Dirichlet prior distribution for the <code>phis</code> . Can be a single value for a Dirichlet prior with uniform parameters, or a vector of length = <code>length(phis)+1</code> . Defaults to <code>1/(length(phis)+1)</code> , the value for the "reference distance" non-informative prior of Berger et al. 2015.

**Details**

As this function returns the unnormalized probability density, it must be normalized somehow to produce a true probability density. This can be achieved using a variety of approaches, including brute force integration of the unnormalized probability density and MCMC algorithms.

**Value**

The unnormalized probability density value. If `log=TRUE`, then the logarithm of the probability density value will be returned.

**Author(s)**

David M. Kaplan <dmkaplan2000@gmail.com>

**References**

Kaplan DM, Cuif M, Fauvelot C, Vigliola L, Nguyen-Huu T, Tiavouane J and Lett C (in press) Uncertainty in empirical estimates of marine larval connectivity. ICES Journal of Marine Science. doi:10.1093/icesjms/fsw182.



Berger JO, Bernardo JM, Sun D (2015) Overall Objective Priors. *Bayesian Analysis* 10:189-221.  
doi:10.1214/14-BA915

### See Also

Other connectivity estimation: [d.rel.conn.beta.prior\(\)](#), [d.rel.conn.dists.func\(\)](#), [d.rel.conn.finite.settlementer.dists.func\(\)](#), [d.rel.conn.multiple\(\)](#), [d.rel.conn.unif.prior\(\)](#), [dual.mark.transmission\(\)](#), [optim.rel.conn.dists\(\)](#), [r.marked.egg.fraction\(\)](#)

### Examples

```
library(ConnMatTools)

ps <- c(0.7,0.5) # Fraction of eggs "marked" at each source site
ks <- c(4,5) # Number of marked settlers among sample from each source site
n.sample <- 20 # Total sample size. Must be >= sum(ks)

phis0 = runif(3,min=0.05)
phis0 = phis0 / sum(phis0)
phis0 = phis0[1:2] # Don't include relative connectivity of unknown sites

nbatch=1e4

library(mcmc)
ans = metrop(d.rel.conn.multinomial.unnorm,
             initial=phis0,nbatch=nbatch,scale=0.1,
             log=TRUE,ps=ps,ks=ks,n.sample=n.sample)
# A more serious test would adjust blen and scale to improve results, and would repeat
# multiple times to get results from multiple MCMC chains.

# Plot marginal distribution of relative connectivity from first site
h=hist(ans$batch[,1],xlab="Rel. Conn., Site 1",
       main="Relative Connectivity for Source Site 1")

# For comparison, add on curve that would correspond to single site calculation
phi = seq(0,1,length.out=40)
d1 = d.rel.conn.beta.prior(phi,ps[1],ks[1],n.sample)

lines(phi,d1*nbatch*diff(h$breaks)[1],col="red",lwd=5)

# Image plot of bivariate probability density
t=table(cut(ans$batch[,1],phi),cut(ans$batch[,2],phi))
image(t,col=heat.colors(12)[12:1],xlab="Rel. Conn., Site 1",ylab="Rel. Conn., Site 2")

# Add line indicate region above which one can never find results as that would
# lead to a total connectivity great than 1
abline(1,-1,col="black",lty="dashed",lwd=3)
```

---

d.rel.conn.multiple     *Functions for estimating the probability distribution of relative connectivity values as a weighted sum over possible input parameters*

---

### Description

These functions calculate the probability density function (`d.rel.conn.multiple`), the probability distribution function (aka the cumulative distribution function; `p.rel.conn.multiple`) and the quantile function (`q.rel.conn.multiple`) for the relative (to all settlers at the destination site) connectivity value for larval transport between a source and destination site. This version allows one to input multiple possible fractions of individuals (i.e., eggs) marked at the source site, multiple possible numbers of settlers collected and multiple possible marked individuals observed in the sample. This gives one the possibility to produce ensemble averages over different input parameter values with different probabilities of being correct.

### Usage

```
d.rel.conn.multiple(  
  phi,  
  ps,  
  ks,  
  ns,  
  weights = 1,  
  d.rel.conn = d.rel.conn.beta.prior,  
  ...  
)
```

```
p.rel.conn.multiple(  
  phi,  
  ps,  
  ks,  
  ns,  
  weights = 1,  
  p.rel.conn = p.rel.conn.beta.prior,  
  ...  
)
```

```
q.rel.conn.multiple.func(  
  ps,  
  ks,  
  ns,  
  weights = 1,  
  p.rel.conn = p.rel.conn.beta.prior,  
  N = 1000,  
  ...  
)
```

```

q.rel.conn.multiple(
  q,
  ps,
  ks,
  ns,
  weights = 1,
  p.rel.conn = p.rel.conn.beta.prior,
  N = 1000,
  ...
)

```

### Arguments

phi	Vector of fractions of individuals (i.e., eggs) from the source population settling at the destination population
ps	Vector of fractions of individuals (i.e., eggs) marked in the source population
ks	Vector of numbers of marked settlers found in sample
ns	Vector of total numbers of settlers collected
weights	Vector of weights for each set of p, k and n values
d.rel.conn	Function to use to calculate probability density for individual combinations of ps, ks and ns. Defaults to <a href="#">d.rel.conn.beta.prior</a> . Could also be <a href="#">d.rel.conn.unif.prior</a> .
...	Additional arguments for the function d.rel.conn or p.rel.conn
p.rel.conn	Function to use to calculate cumulative probability distribution for individual combinations of ps, ks and ns. Defaults to <a href="#">p.rel.conn.beta.prior</a> . Could also be <a href="#">p.rel.conn.unif.prior</a> .
N	Number of points at which to estimate cumulative probability function for reverse approximation of quantile distribution. Defaults to 1000.
q	Vector of quantiles

### Details

If ps, ks, ns and weights can be scalars or vectors of the same length (or lengths divisible into that of the largest input parameter). weights are normalized to sum to 1 before being used to sum probabilities from each individual set of input parameters.

### Value

Vector of probabilities or quantiles, or a function in the case of [q.rel.conn.multiple.func](#)

### Functions

- [d.rel.conn.multiple](#): Estimates quantiles for the probability distribution function for relative connectivity between a pair of sites for multiple possible p, k and n values.
- [p.rel.conn.multiple](#): Estimates the cumulative probability distribution for relative connectivity between a pair of sites for multiple possible p, k and n values.

- `q.rel.conn.multiple.func`: Returns a function to estimate quantiles for the probability distribution function for relative connectivity between a pair of sites for multiple possible  $p$ ,  $k$  and  $n$  values.
- `q.rel.conn.multiple`: Estimates quantiles for the probability distribution function for relative connectivity between a pair of sites for multiple possible  $p$ ,  $k$  and  $n$  values.

### Author(s)

David M. Kaplan <dmkaplan2000@gmail.com>

### References

Kaplan DM, Cuif M, Fauvelot C, Vigliola L, Nguyen-Huu T, Tiavouane J and Lett C (in press) Uncertainty in empirical estimates of marine larval connectivity. ICES Journal of Marine Science. doi:10.1093/icesjms/fsw182.

### See Also

Other connectivity estimation: [d.rel.conn.beta.prior\(\)](#), [d.rel.conn.dists.func\(\)](#), [d.rel.conn.finite.settlement.dists.func\(\)](#), [d.rel.conn.multinomial.unnorm\(\)](#), [d.rel.conn.unif.prior\(\)](#), [dual.mark.transmission\(\)](#), [optim.rel.conn.dists\(\)](#), [r.marked.egg.fraction\(\)](#)

### Examples

```
library(ConnMatTools)

# p values have uniform probability between 0.1 and 0.4
p <- seq(0.1,0.8,length.out=100)

# Weights the same for all except first and last, which are halved
w <- rep(1,length(p))
w[1]<-0.5
w[length(w)]<-0.5

n <- 20 # Sample size
k <- 2 # Marked individuals in sample

# phi values to use for plotting distribution
phi <- seq(0,1,0.01)

prior.shape1 = 1 # Uniform prior
# prior.shape1 = 0.5 # Jeffreys prior

# Plot distribution
plot(phi,d.rel.conn.multiple(phi,p,k,n,w,prior.shape1=prior.shape1),
      main="Probability density for relative connectivity",
      xlab=expression(phi),
      ylab="Probability density",
      type="l")

# Add standard distributions for max and min p values
```

```

lines(phi,d.rel.conn.beta.prior(phi,min(p),k,n,prior.shape1=prior.shape1),
      col="red",lty="dashed")
lines(phi,d.rel.conn.beta.prior(phi,max(p),k,n,prior.shape1=prior.shape1),
      col="red",lty="dashed")

# Add some quantiles
q = q.rel.conn.multiple(c(0.025,0.25,0.5,0.75,0.975),
                      p,k,n,w,prior.shape1=prior.shape1)
abline(v=q,col="green")

```

---

d.rel.conn.unif.prior *Estimate the probability distribution of relative connectivity values assuming a uniform prior distribution*

---

### Description

These functions calculate the probability density function (`d.rel.conn.unif.prior`), the probability distribution function (aka the cumulative distribution function; `p.rel.conn.unif.prior`) and the quantile function (`q.rel.conn.unif.prior`) for the relative (to all settlers at the destination site) connectivity value for larval transport between a source and destination site given a known fraction of marked individuals (i.e., eggs) in the source population. A uniform prior is used for the relative connectivity value.

### Usage

```

d.rel.conn.unif.prior(phi, p, k, n, log = FALSE, ...)
p.rel.conn.unif.prior(phi, p, k, n, log = FALSE, ...)
q.rel.conn.unif.prior(q, p, k, n, log = FALSE, ...)

```

### Arguments

phi	Vector of fractions of individuals (i.e., eggs) from the source population settling at the destination population
p	Fraction of individuals (i.e., eggs) marked in the source population
k	Number of marked settlers found in sample
n	Total number of settlers collected
log	If TRUE, returns natural logarithm of probabilities, except for <code>q.rel.conn.unif.prior</code> , which expects log of probabilities as inputs
...	Extra arguments to Beta distribution functions. See <a href="#">dbeta</a> for details. For expert use only.
q	Vector of quantiles

**Details**

Estimations of the probability distribution are derived from the Beta distribution (see [dbeta](#)) and should be exact to great precision.

**Value**

Vector of probabilities or quantiles.

**Functions**

- `d.rel.conn.unif.prior`: Returns the probability density for relative connectivity between a pair of sites
- `p.rel.conn.unif.prior`: Returns the cumulative probability distribution for relative connectivity between a pair of sites
- `q.rel.conn.unif.prior`: Estimates quantiles for the probability distribution function for relative connectivity between a pair of sites

**Author(s)**

David M. Kaplan <dmkaplan2000@gmail.com>

**References**

Kaplan DM, Cuif M, Fauvelot C, Vigliola L, Nguyen-Huu T, Tiavouane J and Lett C (in press) Uncertainty in empirical estimates of marine larval connectivity. ICES Journal of Marine Science. doi:10.1093/icesjms/fsw182.

**See Also**

Other connectivity estimation: [d.rel.conn.beta.prior\(\)](#), [d.rel.conn.dists.func\(\)](#), [d.rel.conn.finite.settlement\(\)](#), [d.rel.conn.multinomial.unnorm\(\)](#), [d.rel.conn.multiple\(\)](#), [dual.mark.transmission\(\)](#), [optim.rel.conn.dists\(\)](#), [r.marked.egg.fraction\(\)](#)

**Examples**

```
library(ConnMatTools)

k <- 10 # Number of marked settlers among sample
n.obs <- 87 # Number of settlers in sample
n.settlers <- 100 # Total size of settler pool

p <- 0.4 # Fraction of eggs that was marked
phi <- seq(0,1,length.out=101) # Values for relative connectivity

# Probability distribution assuming infinite settler pool and uniform prior
drc <- d.rel.conn.unif.prior(phi,p,k,n.obs)
prc <- p.rel.conn.unif.prior(phi,p,k,n.obs)
qrc <- q.rel.conn.unif.prior(c(0.025,0.975),p,k,n.obs) # 95% confidence interval

# Test with finite settlement function and large (approx. infinite) settler pool
```

```

# Can be a bit slow for large settler pools
dis <- d.rel.conn.finite.settlement(0:(7*n.obs),p,k,n.obs,7*n.obs)

# Quantiles
qis <- q.rel.conn.finite.settlement(c(0.025,0.975),p,k,n.obs,7*n.obs)

# Finite settler pool
dfs <- d.rel.conn.finite.settlement(0:n.settlers,p,k,n.obs,n.settlers)

# Quantiles for the finite settler pool
qfs <- q.rel.conn.finite.settlement(c(0.025,0.975),p,k,n.obs,n.settlers)

# Make a plot of different distributions
plot(phi,drc,type="l",main="Probability of relative connectivity values",
      xlab=expression(phi),ylab="Probability density")
lines(phi,prc,col="blue")
lines((0:(7*n.obs))/(7*n.obs),dis*(7*n.obs),col="black",lty="dashed")
lines((0:n.settlers)/n.settlers,dfs*n.settlers,col="red",lty="dashed")
abline(v=qrc,col="black")
abline(v=qis/(7*n.obs),col="black",lty="dashed")
abline(v=qfs/n.settlers,col="red",lty="dashed")

```

---

damselfish.lods

*Sample LOD score data for simulated and real parent-child pairs*


---

## Description

This dataset contains both simulated and real 'log of the odds ratio' (LOD) scores for potential parent-child pairs of humbug damselfish (*Dascyllus aruanus*) from New Caledonia. Data was generated using **FaMoz**. In all cases, results are for the potential parent with the highest LOD score for a given larval fish (child). Simulated data is based on artificial children generated from either real potential parent-pairs (the 'in' group) or artificial parents generated from observed allelic frequencies (the 'out' group).

## Format

A list with 3 elements:

**in.group** 5000 maximum LOD scores for simulated children from random crossing of real potential parents

**out.group** 5000 maximum LOD scores for simulated children from random crossing of artificial potential parents based on observed allelic frequencies

**real.children** Maximum LOD scores for 200 real juvenile fish

## Author(s)

David M. Kaplan <dmkaplan2000@gmail.com>

**References**

Gerber S, Chabrier P, Kremer A (2003) FAMOZ: a software for parentage analysis using dominant, codominant and uniparentally inherited markers. *Molecular Ecology Notes* 3:479-481. doi:10.1046/j.1471-8286.2003.00439.x

Kaplan et al. (submitted) Uncertainty in marine larval connectivity estimation

**See Also**

See also [d.rel.conn.dists.func](#)

---

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 <a href="#">hockeyStick</a> .
...	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 <a href="#">hockeyStick</a> .
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

- 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
- 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](#)

---

 dual.mark.transmission

*Fraction of eggs marked for male and female mark transmission*


---

### Description

Estimates the fraction of eggs produced at the source site that are the result of crossing parents, one or both of which have been genotyped. Based on the assumption that probability of breeding between pairs of individuals is completely independent of whether or not one or more of those individuals was genotyped.

### Usage

```
dual.mark.transmission(p.female, p.male = p.female)
```

### Arguments

p.female	Fraction of all adult females genotyped in the source population
p.male	Fraction of all adult males genotyped in the source population. Defaults to be equal to p.female

### Value

A list with the following elements:

**prob.matrix** 2x2 matrix with probabilities for producing offspring with male or female known or unknown parents

**p** fraction of all eggs produced at source site that will come from at least one genotyped parent

**p.female.known** Fraction of eggs with a single known female parent among all eggs that have one or more known parents

**p.male.known** Fraction of eggs with a single known male parent among all eggs that have one or more known parents

**p.two.known.parents** Fraction of eggs with two known parents among all eggs that have one or more known parents

### Author(s)

David M. Kaplan <dmkaplan2000@gmail.com>

### References

Kaplan DM, Cuif M, Fauvelot C, Vigliola L, Nguyen-Huu T, Tiavouane J and Lett C (in press) Uncertainty in empirical estimates of marine larval connectivity. ICES Journal of Marine Science. doi:10.1093/icesjms/fsw182.

**See Also**

Other connectivity estimation: `d.rel.conn.beta.prior()`, `d.rel.conn.dists.func()`, `d.rel.conn.finite.settlements.dists.func()`, `d.rel.conn.multinomial.unnorm()`, `d.rel.conn.multiple()`, `d.rel.conn.unif.prior()`, `optim.rel.conn.dists()`, `r.marked.egg.fraction()`

**Examples**

```
library(ConnMatTools)
data(damselfish.lods)

# Histograms of simulated LODs
l <- seq(-1,30,0.5)
h.in <- hist(damselfish.lods$in.group,breaks=1)
h.out <- hist(damselfish.lods$out.group,breaks=1)

# PDFs for marked and unmarked individuals based on simulations
d.marked <- stepfun.hist(h.in)
d.unmarked <- stepfun.hist(h.out)

# Fraction of adults genotyped at source site
p.adults <- 0.25

# prior.shape1=1 # Uniform prior
prior.shape1=0.5 # Jeffreys prior

# Fraction of eggs from one or more genotyped parents
p <- dual.mark.transmission(p.adults)$p

# PDF for relative connectivity
D <- d.rel.conn.dists.func(damselfish.lods$real.children,
                          d.unmarked,d.marked,p,
                          prior.shape1=prior.shape1)

# Estimate most probable value for relative connectivity
phi.mx <- optim.rel.conn.dists(damselfish.lods$real.children,
                              d.unmarked,d.marked,p)$phi

# Estimate 95% confidence interval for relative connectivity
Q <- q.rel.conn.dists.func(damselfish.lods$real.children,
                          d.unmarked,d.marked,p,
                          prior.shape1=prior.shape1)

# Plot it up
phi <- seq(0,1,0.001)
plot(phi,D(phi),type="l",
      xlim=c(0,0.1),
      main="PDF for relative connectivity",
      xlab=expression(phi),
      ylab="Probability density")

abline(v=phi.mx,col="green",lty="dashed")
abline(v=Q(c(0.025,0.975)),col="red",lty="dashed")
```

---

`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

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

### Value

A list with at least the following two items:

<code>values</code>	A set of eigenvalues
<code>vectors</code>	A matrix of eigenvectors

### Author(s)

David M. Kaplan <dmkaplan2000@gmail.com>

**See Also**

See also [arpack](#)

---

gammaParamsConvert	<i>Gamma distribution shape and scale parameters from mean and standard deviation, or vice-versa</i>
--------------------	--

---

**Description**

Calculates shape and scale parameters for a gamma distribution from the mean and standard deviation of the distribution, or vice-versa. One supplies either mean and sd or shape and scale and the function returns a list with all four parameter values.

**Usage**

```
gammaParamsConvert(...)
```

**Arguments**

... This function can be run either supplying mean and sd, or supplying shape and scale, but not both pairs of parameters.

**Value**

A list with mean, sd, shape and scale parameters of the corresponding gamma distribution.

**Author(s)**

David M. Kaplan <dmkaplan2000@gmail.com>

**Examples**

```
library(ConnMatTools)
mn <- 1
sd <- 0.4
l <- gammaParamsConvert(mean=mn, sd=sd)
x <- seq(0, 2, length.out=50)

# Plot gamma and normal distributions - for sd << mean, the two should be very close
plot(x, dgamma(x, l$shape, scale=l$scale),
     main="Normal versus Gamma distributions", type="l")
lines(x, dnorm(x, l$mean, l$sd), col="red")
```

---

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

**See Also**

See also [optimalSplitConnMat](#),

---

optim.rel.conn.dists    *Maximum-likelihood estimate for relative connectivity given two distributions for scores for marked and unmarked individuals*

---

**Description**

This function calculates the value for relative connectivity that best fits a set of observed score values, a pair of distributions for marked and unmarked individuals and an estimate of the fraction of eggs marked in the source population,  $p$ .

**Usage**

```
optim.rel.conn.dists(
  obs,
  d.unmarked,
  d.marked,
  p = 1,
  phi0 = 0.5,
  method = "Brent",
  lower = 0,
  upper = 1,
  ...
)
```

**Arguments**

obs	Vector of observed score values for potentially marked individuals
d.unmarked	A function representing the PDF of unmarked individuals. Must be normalized so that it integrates to 1 for the function to work properly.
d.marked	A function representing the PDF of marked individuals. Must be normalized so that it integrates to 1 for the function to work properly.

p	Fraction of individuals (i.e., eggs) marked in the source population
phi0	Initial value for $\phi$ , the fraction of settlers at the destination population that originated at the source population, for <code>optim</code> function. Defaults to 0.5.
method	Method variable for <code>optim</code> function. Defaults to "Brent".
lower	Lower limit for search for fraction of marked individuals. Defaults to 0.
upper	Upper limit for search for fraction of marked individuals. Defaults to 1.
...	Additional arguments for the <code>optim</code> function.

**Value**

A list with results of optimization. Optimal fraction of marked individuals is in phi field. Negative log-likelihood is in the neg.log.prob field. See `optim` for other elements of list.

**Author(s)**

David M. Kaplan <dmkaplan2000@gmail.com>

**References**

Kaplan DM, Cuif M, Fauvelot C, Vigliola L, Nguyen-Huu T, Tiavouane J and Lett C (in press) Uncertainty in empirical estimates of marine larval connectivity. ICES Journal of Marine Science. doi:10.1093/icesjms/fsw182.

**See Also**

Other connectivity estimation: `d.rel.conn.beta.prior()`, `d.rel.conn.dists.func()`, `d.rel.conn.finite.settlement()`, `d.rel.conn.multinomial.unnorm()`, `d.rel.conn.multiple()`, `d.rel.conn.unif.prior()`, `dual.mark.transmission()`, `r.marked.egg.fraction()`

**Examples**

```
library(ConnMatTools)
data(damselfish.lods)

# Histograms of simulated LODs
l <- seq(-1,30,0.5)
h.in <- hist(damselfish.lods$in.group,breaks=1)
h.out <- hist(damselfish.lods$out.group,breaks=1)

# PDFs for marked and unmarked individuals based on simulations
d.marked <- stepfun.hist(h.in)
d.unmarked <- stepfun.hist(h.out)

# Fraction of adults genotyped at source site
p.adults <- 0.25

# prior.shape1=1 # Uniform prior
prior.shape1=0.5 # Jeffreys prior

# Fraction of eggs from one or more genotyped parents
```

```

p <- dual.mark.transmission(p.adults)$p

# PDF for relative connectivity
D <- d.rel.conn.dists.func(damselfish.lods$real.children,
                          d.unmarked,d.marked,p,
                          prior.shape1=prior.shape1)

# Estimate most probable value for relative connectivity
phi.mx <- optim.rel.conn.dists(damselfish.lods$real.children,
                              d.unmarked,d.marked,p)$phi

# Estimate 95% confidence interval for relative connectivity
Q <- q.rel.conn.dists.func(damselfish.lods$real.children,
                          d.unmarked,d.marked,p,
                          prior.shape1=prior.shape1)

# Plot it up
phi <- seq(0,1,0.001)
plot(phi,D(phi),type="l",
      xlim=c(0,0.1),
      main="PDF for relative connectivity",
      xlab=expression(phi),
      ylab="Probability density")

abline(v=phi.mx,col="green",lty="dashed")
abline(v=Q(c(0.025,0.975)),col="red",lty="dashed")

```

---

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**

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

**Value**

A list with the following elements:

<code>betas</code>	Vector of all beta values tested
<code>num.subpops</code>	Vector of number of subpopulations found for each value of beta
<code>qualities</code>	Vector of the quality statistic for each subpopulation division
<code>subpops</code>	A matrix with dimensions $\text{dim}(\text{conn.mat})[2] \times \text{length}(\text{betas})$ indicating which subpopulation each site belongs to
<code>best.splits</code>	A list indicating for each number of subpopulations, which column of <code>subpops</code> contains the division with the lowest quality statistic. E.g., <code>best.splits[["4"]]\$index</code> 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  $C_{i,j}$  is dispersal from  $i$  to  $j$ , whereas in this R package, the connectivity matrix is oriented so that  $C_{i,j}$  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., Andre, C., Doos, 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)
```

---

prob.marked	<i>Returns probability a set of observations correspond to marked individuals</i>
-------------	---

---

**Description**

This function returns the probability each of a set of observations corresponds to a marked individual given the distribution of scores for unmarked and marked individuals and the fraction of individuals that are marked.

**Usage**

```
prob.marked(obs, d.unmarked, d.marked, phi = 0.5, p = 1)
```

**Arguments**

obs	A vector of score values for a random sample of (marked and unmarked) individuals from the population
d.unmarked	A function representing the PDF of unmarked individuals. Must be normalized so that it integrates to 1 for the function to work properly.
d.marked	A function representing the PDF of marked individuals. Must be normalized so that it integrates to 1 for the function to work properly.

phi	The fraction of settlers at the destination population that originated at the source population. Defaults to 0.5, which would correspond to an even sample of marked and unmarked individuals.
p	Fraction of individuals (i.e., eggs) marked in the source population. Defaults to 1.

**Value**

A vector of the same size as obs containing the probability that each individual is marked

**Author(s)**

David M. Kaplan <dmkaplan2000@gmail.com>

**References**

Kaplan DM, Cuif M, Fauvelot C, Vigliola L, Nguyen-Huu T, Tiavouane J and Lett C (in press) Uncertainty in empirical estimates of marine larval connectivity. ICES Journal of Marine Science. doi:10.1093/icesjms/fsw182.

**See Also**

See also [d.rel.conn.dists.func](#), [optim.rel.conn.dists](#).

---

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 <a href="#">eigs</a> 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

*Quality measure for subpopulation division*

---

**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 <a href="#">subpopsVectorToList</a> 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., Andre, C., Doos, K., and Jonsson, P. R. 2012. Identification of subpopulations from connectivity matrices. *Ecography*, 35: 1004-1016.

### See Also

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

---

r.marked.egg.fraction *Estimates of fraction of eggs marked accounting for variability in reproductive output*

---

### Description

This function estimates the fraction of eggs "marked" at a site (where the "mark" could be micro-chemical or genetic) taking into account uncertainty in female (and potentially male in the case of dual genetic mark transmission) reproductive output. It generates a set of potential values for the fraction of eggs marked assuming that reproductive output of each marked or unmarked mature individual is given by a random variable drawn from a single probability distribution with known mean and standard deviation (or equivalently coefficient of variation) **and** that the numbers of marked and unmarked individuals are large enough that the central limit theorem applies and, therefore, their collective reproductive outputs are reasonably well described by a gamma distribution whose mean and standard deviation are appropriately scaled based on the number of individual reproducers. The function also returns the total egg production corresponding to each fraction of marked eggs, needed for estimating absolute connectivity values (i.e., elements of the connectivity matrix needed for assessing population persistence).

**Usage**

```
r.marked.egg.fraction(
  n,
  n.females,
  n.marked.females = round(n.females * p.marked.females),
  mean.female = 1,
  cv.female,
  dual = FALSE,
  male.uncert = FALSE,
  n.males = n.females,
  n.marked.males = tryCatch(round(n.males * p.marked.males), error = function(e)
    n.marked.females),
  mean.male = mean.female,
  cv.male = cv.female,
  p.marked.females,
  p.marked.males = p.marked.females
)
```

**Arguments**

n	Number of random values to estimates
n.females	Total number of mature females in the population
n.marked.females	Number of marked females in population
mean.female	Mean egg production of each mature female. Defaults to 1.
cv.female	Coefficient of variation of reproductive output of an individual mature female
dual	Logical variable. If TRUE, then the fraction of marked eggs is calculated assuming dual (male and female) mark transmission. Defaults to FALSE.
male.uncert	Logical variable. If TRUE, then variability in male sperm output is also taken into account when estimating the number of marked eggs. Defaults to FALSE.
n.males	Total number of mature males in the population. Only used if dual=TRUE. Defaults to being equal to n.females.
n.marked.males	Number of marked males in population. Only used if dual=TRUE. Defaults to being equal to n.marked.females.
mean.male	Mean sperm production of each mature male. Only used if dual=TRUE and male.uncert=TRUE. Defaults to being equal to mean.female.
cv.male	Coefficient of variation of reproductive output of an individual mature male. Only used if dual=TRUE and male.uncert=TRUE. Defaults to being equal to cv.female.
p.marked.females	Fraction of marked females in population. Can be supplied instead of n.marked.females. Ignored if n.marked.females is given.
p.marked.males	Fraction of marked males in population. Can be supplied instead of n.marked.males. Only used if dual=TRUE. Ignored if n.marked.males is given.

**Value**

A list with the following elements:

**p** Vector of length *n* with estimates for fraction of marked eggs

**eggs** Vector of length *n* with estimates for total egg production

**marked.eggs** Vector of length *n* with estimates for total number of marked eggs produced

**sperm** Only returned if `dual=TRUE`. If `male.uncert=FALSE`, then a scalar equal to `n.males`. Otherwise, a vector of length *n* with estimates for total sperm production

**marked.sperm** Only returned if `dual=TRUE`. If `male.uncert=FALSE`, then a scalar equal to `n.marked.males`. Otherwise, a vector of length *n* with estimates for total marked sperm production

**Author(s)**

David M. Kaplan <dmkaplan2000@gmail.com>

**References**

Kaplan DM, Cuif M, Fauvelot C, Vigliola L, Nguyen-Huu T, Tiavouane J and Lett C (in press) Uncertainty in empirical estimates of marine larval connectivity. ICES Journal of Marine Science. doi:10.1093/icesjms/fsw182.

**See Also**

Other connectivity estimation: [d.rel.conn.beta.prior\(\)](#), [d.rel.conn.dists.func\(\)](#), [d.rel.conn.finite.settlement](#), [d.rel.conn.multinomial.unnorm\(\)](#), [d.rel.conn.multiple\(\)](#), [d.rel.conn.unif.prior\(\)](#), [dual.mark.transmission](#), [optim.rel.conn.dists\(\)](#)

**Examples**

```
library(ConnMatTools)

n.females <- 500
n.marked.females <- 100
p.marked.females <- n.marked.females/n.females
mn <- 1
cv <- 1
# Numbers of males and marked males and variance in male sperm production
# assumed the same as values for females

# Random values from distribution of pure female mark transmission
F=r.marked.egg.fraction(1000,n.females=n.females,n.marked.females=n.marked.females,
                      mean.female=mn,cv.female=cv)

# Random values from distribution of dual female-male mark transmission, but
# fraction of marked eggs only depends on fraction of marked males
Fm=r.marked.egg.fraction(1000,n.females=n.females,n.marked.females=n.marked.females,
                       mean.female=mn,cv.female=cv,dual=TRUE,male.uncert=FALSE)

# Random values from distribution of dual female-male mark transmission, with
# fraction of marked eggs depending on absolute marked and unmarked sperm output
```

```

FM=r.marked.egg.fraction(1000,n.females=n.females,n.marked.females=n.marked.females,
                        mean.female=mn,cv.female=cv,dual=TRUE,male.uncert=TRUE)

# Plot of pure female mark transmission
hist(F$p,50,main="Female mark transmission",
     xlab="Fraction of marked eggs",
     ylab="Frequency")

# Female+male mark transmission, but no variability in male mark transmission
h <- hist(Fm$p,50,main="Female+male mark transmission, no male uncert.",
         xlab="Fraction of marked eggs",
         ylab="Frequency")
hh <- hist((1-p.marked.females)*F$p + p.marked.females,
         breaks=c(-Inf,h$breaks,Inf),plot=FALSE)
lines(hh$mids,hh$counts,col="red")

# Plot of pure female mark transmission
h <- hist(FM$p,50,plot=FALSE)
hh <- hist(Fm$p,
         breaks=c(-Inf,h$breaks,Inf),plot=FALSE)
plot(h,ylim=c(0,1.1*max(hh$counts,h$counts)),
     main="Female+Male mark transmission, male uncert.",
     xlab="Fraction of marked eggs",
     ylab="Frequency")
lines(hh$mids,hh$counts,col="red")

```

---

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 <a href="#">subpopsVectorToList</a> .
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 <a href="#">splitConnMat</a>

**Author(s)**

David M. Kaplan <dmkaplan2000@gmail.com>

**References**

Jacobi, M. N., Andre, C., Doos, 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 <a href="#">subpopsVectorToList</a> 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., Andre, C., Doos, 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

<code>conn.mat</code>	a square connectivity matrix.
<code>slope</code>	slope at the origin of the settler-recruit relationship. Only interesting to fix this argument if it is a vector of length = <code>dim(conn.mat)[2]</code> (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).
<code>natural.LEP</code>	value of lifetime-egg-production (LEP), also known as eggs-per-recruit, in the absence of fishing. Can be a vector of length = <code>dim(conn.mat)[2]</code> . Defaults to 1.
<code>critical.FLEP</code>	Fraction of <code>natural.LEP</code> at which collapse occurs. Defaults to 0.35.
<code>use.arpack</code>	Boolean determining if calculation is to be done with <a href="#">arpack</a> function from the <a href="#">igraph</a> package. This is much quicker for large matrices, but requires <a href="#">igraph</a> . Defaults to TRUE, but will use <code>eigen</code> instead if <a href="#">igraph</a> 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](mailto: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

*Split connectivity matrix into subpopulations*


---

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

**See Also**

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

---

stepfun.hist

*Create a probability density step function from a histogram object*

---

**Description**

This function creates a step function from the bars in a histogram object. By default, the step function will be normalized so that it integrates to 1.

**Usage**

```
stepfun.hist(h, ..., normalize = TRUE)
```

**Arguments**

h	an object of type histogram
...	Additional arguments for the default <a href="#">stepfun</a> function.
normalize	Boolean indicating whether or not to normalize the output stepfun so that it integrates to 1. Defaults to TRUE. If FALSE, then the function will integrate to $\text{sum}(h\$counts)$

**Value**

A function of class stepfun. The height of the steps will be divided by the distance between breaks and possibly the total count.

**Author(s)**

David M. Kaplan <dmkaplan2000@gmail.com>

**See Also**

See also [d.rel.conn.dists.func](#), [optim.rel.conn.dists](#).

**Examples**

```
library(ConnMatTools)
data(damselfish.lods)

# Histograms of simulated LODs
l <- seq(-1,30,0.5)
h.in <- hist(damselfish.lods$in.group,breaks=1)
h.out <- hist(damselfish.lods$out.group,breaks=1)

# PDFs for marked and unmarked individuals based on simulations
d.marked <- stepfun.hist(h.in)
```

```

d.unmarked <- stepfun.hist(h.out)

# Fraction of adults genotyped at source site
p.adults <- 0.25

# prior.shape1=1 # Uniform prior
prior.shape1=0.5 # Jeffreys prior

# Fraction of eggs from one or more genotyped parents
p <- dual.mark.transmission(p.adults)$p

# PDF for relative connectivity
D <- d.rel.conn.dists.func(damselfish.lods$real.children,
                          d.unmarked,d.marked,p,
                          prior.shape1=prior.shape1)

# Estimate most probable value for relative connectivity
phi.mx <- optim.rel.conn.dists(damselfish.lods$real.children,
                               d.unmarked,d.marked,p)$phi

# Estimate 95% confidence interval for relative connectivity
Q <- q.rel.conn.dists.func(damselfish.lods$real.children,
                           d.unmarked,d.marked,p,
                           prior.shape1=prior.shape1)

# Plot it up
phi <- seq(0,1,0.001)
plot(phi,D(phi),type="l",
      xlim=c(0,0.1),
      main="PDF for relative connectivity",
      xlab=expression(phi),
      ylab="Probability density")

abline(v=phi.mx,col="green",lty="dashed")
abline(v=Q(c(0.025,0.975)),col="red",lty="dashed")

```

---

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

## \* connectivity estimation

- d.rel.conn.beta.prior, 7
- d.rel.conn.dists.func, 10
- d.rel.conn.finite.settlement, 13
- d.rel.conn.multinomial.unnorm, 15
- d.rel.conn.multiple, 18
- d.rel.conn.unif.prior, 21
- dual.mark.transmission, 28
- optim.rel.conn.dists, 35
- r.marked.egg.fraction, 42

## \* data

- chile.loco, 4
- damselish.lods, 23

## \* package

- ConnMatTools, 5

approxfun, 11

arpack, 30, 31, 49, 50

betasVectorDefault, 2, 38

BevertonHolt, 3, 25, 27

chile.loco, 4

ConnMatTools, 5

d.mix.dists.func, 6

d.rel.conn.beta.prior, 6, 7, 12, 14, 17, 19,  
20, 22, 29, 36, 44

d.rel.conn.dists.func, 6, 9, 10, 14, 17, 20,  
22, 24, 29, 36, 40, 44, 51

d.rel.conn.finite.settlement, 9, 12, 13,  
17, 20, 22, 29, 36, 44

d.rel.conn.multinomial.unnorm, 9, 12, 14,  
15, 20, 22, 29, 36, 44

d.rel.conn.multiple, 9, 12, 14, 17, 18, 22,  
29, 36, 44

d.rel.conn.unif.prior, 9, 12, 14, 17, 19,  
20, 21, 29, 36, 44

damselish.lods, 23

dbeta, 21, 22

DispersalPerRecruitModel, 24, 27, 33

DPRHomerangeGravity, 26

dual.mark.transmission, 9, 12, 14, 17, 20,  
22, 28, 36, 44

eigen, 30

eigs, 30, 41, 50

gammaParamsConvert, 31

hockeyStick, 24–27, 32

igraph, 30, 49

integrate, 8, 11

laplacianConnMat, 33

localRetention, 34

mergeSubpops, 34, 39

optim, 36

optim.rel.conn.dists, 6, 9, 12, 14, 17, 20,  
22, 29, 35, 40, 44, 51

optimalSplitConnMat, 3, 6, 35, 37, 42, 46, 51

p.rel.conn.beta.prior, 19

p.rel.conn.beta.prior  
(d.rel.conn.beta.prior), 7

p.rel.conn.dists.func  
(d.rel.conn.dists.func), 10

p.rel.conn.finite.settlement  
(d.rel.conn.finite.settlement),  
13

p.rel.conn.multiple  
(d.rel.conn.multiple), 18

p.rel.conn.unif.prior, 19

p.rel.conn.unif.prior  
(d.rel.conn.unif.prior), 21

prob.marked, 39

protectedAreaSelection, 40

q.rel.conn.beta.prior  
    (d.rel.conn.beta.prior), 7  
q.rel.conn.beta.prior.func, 8  
q.rel.conn.dists.func  
    (d.rel.conn.dists.func), 10  
q.rel.conn.finite.settlement  
    (d.rel.conn.finite.settlement),  
    13  
q.rel.conn.multiple  
    (d.rel.conn.multiple), 18  
q.rel.conn.multiple.func, 19  
q.rel.conn.unif.prior, 21  
q.rel.conn.unif.prior  
    (d.rel.conn.unif.prior), 21  
qualitySubpops, 39, 41, 47, 52, 53  
  
r.marked.egg.fraction, 9, 12, 14, 17, 20,  
    22, 29, 36, 42  
recSplitConnMat, 39, 45, 51–53  
reducedConnMat, 46  
relativeLocalRetention, 41, 42, 47  
  
selfRecruitment, 48  
settlerRecruitSlopeCorrection, 49  
splitConnMat, 38, 39, 45, 46, 50  
stepfun, 51  
stepfun.hist, 51  
subpopsVectorToList, 34, 42, 45, 46, 51, 52