

# Package ‘memgene’

February 21, 2022

**Type** Package

**Title** Spatial Pattern Detection in Genetic Distance Data Using Moran's Eigenvector Maps

**Version** 1.0.2

**Date** 2022-02-21

**Author** Pedro Peres-Neto, Paul Galpern

**Maintainer** Paul Galpern <pgalpern@ucalgary.ca>

**Description** Can detect relatively weak spatial genetic patterns by using Moran's Eigenvector Maps (MEM) to extract only the spatial component of genetic variation. Has applications in landscape genetics where the movement and dispersal of organisms are studied using neutral genetic variation.

**License** GPL ( $\geq 2$ )

**Imports** ade4, gdistance, raster, sp, vegan

**Suggests** adegenet, geosphere, knitr

**VignetteBuilder** knitr

**BuildVignettes** true

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2022-02-21 19:00:02 UTC

## R topics documented:

memgene-package	2
codomToPropShared	3
mgForward	4
mgLandscape	5
mgMap	8
mgMEM	10
mgQuick	12
mgRDA	14
mgVarPart	15
<b>Index</b>	<b>18</b>

---

memgene-package      *Spatial pattern detection in genetic distance data using Moran's Eigenvector Maps*

---

## Description

Memgene can detect relatively weak spatial genetic patterns by using Moran's Eigenvector Maps (MEM) to extract only the spatial component of genetic variation. Memgene has applications in landscape genetics where the movement and dispersal of organisms are studied using neutral genetic variation.

## Details

Package: memgene  
Type: Package  
Version: 1.0  
Date: 2014-06-07  
License: GPL (>=2)

## Author(s)

Paul Galpern (pgalpern@ucalgary.ca)  
Pedro Peres-Neto (peres-neto.pedro@uqam.ca)  
Maintainer: Paul Galpern (pgalpern@ucalgary.ca)

## References

Galpern, P., Peres-Neto, P., Polfus, J., and Manseau, M. 2014. MEMGENE: Spatial pattern detection in genetic distance data. Submitted.

## Examples

```
## The basic interface to MEMGENE is mgQuick()
?mgQuick

## For landscape genetic analysis with MEMGENE see mgLandscape()
?mgLandscape
```

---

codomToPropShared	<i>Find proportion of shared alleles genetic distances from a codominant alleles matrix</i>
-------------------	---

---

### Description

Given a matrix with two adjacent columns for each locus (e.g. LOCUS1a, LOCUS1b, LOCUS2a, LOCUS2b, ...) containing codominant alleles, where individual genotypes are in rows, find the proportion of shared alleles (Bowcock et al., 1994) among individuals using functions in the adegenet package.

This is a convenience function that wraps adegenet routines.

Note that any type of genetic distance matrix can be used in MEMGENE, and the proportion of shared alleles metric is not a requirement.

### Usage

```
codomToPropShared(alleles, missingData = c(-98,-99), genind=FALSE)
```

### Arguments

alleles	A matrix with two adjacent columns for each locus containing codominant alleles
missingData	A vector of any length giving the values in the alleles matrix representing missing data (NA also represents missing)
genind	Return a genind object rather than the proportion of shared alleles genetic distance matrix. A genind object can be used by various functions in the adegenet package.

### Details

First prepares the alleles matrix into a format that can be converted using functions in the adegenet package to the genind format. propShared is then run on this object.

### Value

Returns a genetic distance matrix using the proportion of shared alleles metric (Bowcock et al., 1994)

### Author(s)

Pedro Peres-Neto (peres-neto.pedro@uqam.ca)  
Paul Galpern (pgalpern@ucalgary.ca)

## References

Bowcock AM, Ruizlinares A, Tomfohrde J, et al. 1994 High resolution of human evolutionary trees with polymorphic microsatellites. *Nature*, 368, 455-457.

## Examples

```
radialData <- read.csv(system.file("extdata/radial.csv", package="memgene"))
radialGen <- radialData[, -c(1,2)]
radialDM <- codomToPropShared(radialGen)
```

---

mgForward

*Forward selection of MEM eigenvectors against genetic distance data*


---

## Description

This function calls mgRDA repeatedly in order to identify a reduced set of all MEM eigenvectors (i.e. spatial patterns).

## Usage

```
mgForward(genD, vectorsMEM, perm = 100, alpha = 0.05)
```

## Arguments

genD	A symmetrical distance matrix giving the genetic distances among individual genotypes or populations
vectorsMEM	A matrix giving a set of MEM eigenvectors
perm	The number of permutations in a randomization test
alpha	The 1-alpha level for forward selection

## Details

A wrapper for [mgRDA](#) designed for forward selection

## Value

A list  
 \$selectedMEM gives the indices of the input vectorsMEM that were selected and can then be used in a call to mgRDA(..., full=TRUE)

## Author(s)

Pedro Peres-Neto (peres-neto.pedro@uqam.ca)  
 Paul Galpern (pgalpern@ucalgary.ca)

**Examples**

```
## Not run:

## Prepare the radial data for analysis
radialData <- read.csv(system.file("extdata/radial.csv", package="memgene"))
radialGen <- radialData[, -c(1,2)]
radialXY <- radialData[, 1:2]
if (require(adegenet)) {
  radialDM <- codomToPropShared(radialGen)
} else {
  stop("adegenet package required to produce genetic distance matrix in example.")
}

## Find MEM eigenvectors given sampling locations
## by first finding the Euclidean distance matrix
radialEuclid <- dist(radialXY)
radialMEM <- mgMEM(radialEuclid)

## Forward select significant MEM eigenvectors using RDA
## Positive MEM eigenvectors (positive spatial autocorrelation) first
radialPositive <- mgForward(radialDM,
  radialMEM$vectorsMEM[, radialMEM$valuesMEM > 0])
## Negative MEM eigenvectors (negative spatial autocorrelation) second
radialNegative <- mgForward(radialDM,
  radialMEM$vectorsMEM[, radialMEM$valuesMEM < 0])

## Summarize the selected MEM eigenvectors
allSelected <- cbind(radialMEM$vectorsMEM[, radialMEM$valuesMEM > 0][
  , na.omit(radialPositive$selectedMEM)],
  radialMEM$vectorsMEM[, radialMEM$valuesMEM < 0][
  , na.omit(radialNegative$selectedMEM)])

## Use the selected MEM eigenvectors in a final model
radialAnalysis <- mgRDA(radialDM, allSelected, full=TRUE)

## End(Not run)
```

**Description**

Use least-cost path distances among sampling locations on a resistance surface, rather than Euclidean distances (as in [mgQuick](#)), to extract MEM eigenvectors. The goal is to compare multiple resistance surfaces (i.e. representing alternative hypotheses about landscape resistance) in terms of the proportion of variation in genetic distance they explain. This is often a goal in landscape genetic

analysis. By default Euclidean distances (i.e. representing a surface with no landscape resistance) are also analyzed unless `euclid=FALSE`.

The analysis steps are as follows:

1. Find MEM eigenvectors given a distance matrix extracted from the coordinates (`coords`). In the case of a resistance surface the distances are least-cost paths among sampling locations found using the function `gdistance::costDistance`. In the Euclidean case Euclidean distances are used. For all distance matrices a minimum spanning tree of the locations is found, followed by truncation of the tree (see [mgMEM](#))
2. Perform separate forward selections of positive and negative MEM eigenvectors against genetic distance (`genD`), to identify a significant subset, using parameters `forwardPerm` as the number of permutations and `forwardAlpha` as the alpha level for a significant eigenvector. NOTE: The number of permutations `forwardPerm` is set at 100 by default to reduce analysis time for exploratory analyses. This number should be increased for final analyses.
3. Use variation partitioning against the genetic distance matrix to find the proportion of variation in genetic distance explained by the selected positive and negative MEM eigenvectors (i.e. fraction [a] representing spatial genetic variation explained by the resistance surface hypothesis) and the matrix of coordinates (i.e. fraction [c] representing spatial genetic variation not explained by the resistance hypothesis). These [a] and [c] fractions can be used to inform model selection (see below).

### Usage

```
mgLandscape(resistance, genD, coords, euclid=TRUE, forwardPerm=100,
            forwardAlpha=0.05, finalPerm=1000, verbose=TRUE)
## S3 method for class 'mgLandscape'
print(x, ...)
```

### Arguments

<code>resistance</code>	A <code>RasterLayer</code> produced by the <a href="#">raster</a> package in a planar (i.e. not longitude/latitude) projection giving the hypothesized resistance to movement of landscape features (all cells must be either missing as NA or $\geq 1$ ) To test multiple resistance hypotheses provide a <code>RasterStack</code> or <code>RasterBrick</code> also produced by the <a href="#">raster</a> package.
<code>genD</code>	A symmetrical distance matrix giving the genetic distances among individual genotypes or populations
<code>coords</code>	A two column matrix or <code>data.frame</code> of x and y coordinates of sampling locations of individual genotypes or populations. Must be in the same planar projection as the resistance surface. Geographic coordinates (i.e. longitude/latitude) must be projected before use.
<code>euclid</code>	If TRUE will test the Euclidean distances among sampling locations in addition to the resistance surface(s) supplied. Including a Euclidean surface is recommended as a null model.

forwardPerm	The number of permutations in the randomization test for the forward selection of MEM eigenvectors. The default forwardPerm=100 is sufficient for exploratory purposes, however this should be increased for final analyses.
forwardAlpha	The 1-alpha level for the forward selection process
finalPerm	The number of permutations to test the significance of the [a], [c] and [abc] fractions.
verbose	If TRUE then report progress to the console
x	An object of class mgLandscape produced by the mgLandscape function
...	Additional parameters passed to print

### Value

A code\$summary table giving the results of the variation partitioning. The following table provides an interpretation of each of the fractions returned:

Proportion of variation in genetic distance that is... (RsQAdj)
[abc] explained by spatial predictors
[a] spatial and explained by selected patterns in the model
[c] spatial and explained by coordinates not patterns in the model
[b] spatial and confounded between the model and coordinates
[d] residual not explained by the spatial predictors

A good model will have a relatively high [a] fraction and relatively low [c] fraction indicating that the selected patterns in the landscape model have captured a large proportion of the spatial variation in genetic distance.

### Author(s)

Pedro Peres-Neto (peres-neto.pedro@uqam.ca)  
Paul Galpern (pgalpern@ucalgary.ca)

### References

Galpern, P., Peres-Neto, P., Polfus, J., and Manseau, M. 2014. MEMGENE: Spatial pattern detection in genetic distance data. Submitted.

### Examples

```
## Not run:
## Compare data generated using the radial data against three landscape models
##
## Prepare two resistance surfaces to test (the true radial, and the false river)
## These are produced as a RasterStack object
if (require(raster)) {
  resistanceMaps <- stack(
    raster(system.file("extdata/radial.asc", package="memgene")),
```

```

        raster(system.file("extdata/river.asc", package="memgene")))
} else {
  stop("raster package required for mgLandscape.")
}

## Prepare the radial data for analysis
radialData <- read.csv(system.file("extdata/radial.csv", package="memgene"))
radialGen <- radialData[, -c(1,2)]
radialXY <- radialData[, 1:2]
if (require(adeigenet)) {
  radialDM <- codomToPropShared(radialGen)
} else {
  stop("adeigenet package required to produce genetic distance matrix in example.")
}

## Analyse the two resistance surfaces and a Euclidean model
## and produce a table comparing the three
## Set permutations at low values for a faster (though less accurate) run
compareThree <- mgLandscape(resistanceMaps, radialDM, radialXY, euclid=TRUE,
  forwardPerm=100, finalPerm=100)

print(compareThree)
## Results can vary between runs because selected MEM eigenvectors may vary.
## Setting forwardPerm higher will increase consistency in this regard.
##
## We see that the true radial surface has the highest [a] fraction and
## the lowest [c] fraction indicating that it does well at capturing
## the spatial genetic variation that we expect in this simulated genetic data

## End(Not run)

```

---

mgMap

*Visualization of MEMGENE variables*


---

## Description

A high-level plotting interface for the bubble plot visualization of MEMGENE variables. If there are exactly two columns in memgene and therefore two MEMGENE variables to be plotted, then a single plotting window is created with the two plots side by side. Otherwise each MEMGENE variable is plotted in its own window unless.

## Usage

```
mgMap(coords, memgene, wid = NULL, hei = NULL, dev.open = FALSE,
  add.plot = FALSE, legend = FALSE, ...)
```

**Arguments**

coords	A two column matrix or data.frame of x and y coordinates of sampling locations of individual genotypes
memgene	A matrix giving as columns the MEMGENE variables to be plotted (e.g. can be subsetted from the \$memgene element produced by <a href="#">mgQuick</a> )
wid	The width of the plotting device to be created. If NULL the decision is made by the function.
hei	The height of the plotting device to be created. If NULL the decision is made by the function.
dev.open	If TRUE do not open a new plotting device.
add.plot	If TRUE superimposing bubbles on an existing plot or map.
legend	If TRUE add a legend to the plot
...	Additional parameters passed to the sr.value function modified from Borcard et al (2011).

**Details**

This function embeds slightly modified versions of `sr.value`, `scatterutil.legend.bw.circle`, and `scatterutil.legend.circle.grey` distributed with Borcard et al. (2012) which are themselves modified from similar functions distributed with the [ade4](#) package under a GPL-2 license.

**Value**

Side effect. A plot is produced.

**Author(s)**

Pedro Peres-Neto ([peres-neto.pedro@uqam.ca](mailto:peres-neto.pedro@uqam.ca))  
 Paul Galpern ([pgalpern@ucalgary.ca](mailto:pgalpern@ucalgary.ca))

**References**

Borcard, D., Gillet, F., and Legendre. P. 2011. Numerical Ecology with R. Springer, New York.

**Examples**

```
## Not run:
## Prepare the radial data for analysis
radialData <- read.csv(system.file("extdata/radial.csv", package="memgene"))
radialGen <- radialData[, -c(1,2)]
radialXY <- radialData[, 1:2]
if (require(adeigenet)) {
  radialDM <- codomToPropShared(radialGen)
} else {
  stop("adeigenet package required to produce genetic distance matrix in example.")
}

## Run the MEMGENE analysis
```

```

radialAnalysis <- mgQuick(radialDM, radialXY)

## Visualize the first two MEMGENE variables side-by-side
mgMap(radialXY, radialAnalysis$memgene[, 1:2])

## Visualize the first MEMGENE variable superimposed over a raster map
## with the same coordinate system, AND include a legend
if (require(raster)) {
  resistanceMap <- raster(system.file("extdata/radial.asc", package="memgene"))
  plot(resistanceMap, legend=FALSE)
  mgMap(radialXY, radialAnalysis$memgene[, 1], add.plot=TRUE, legend=TRUE)
} else {
  mgMap(radialXY, radialAnalysis$memgene[, 1], legend=TRUE)
}

## End(Not run)

```

---

mgMEM

*Extraction of MEM eigenvectors given distances among sampling locations*

---

## Description

Extract MEM eigenvectors given a distance matrix among sampling locations of genetic material. This matrix could be Euclidean or otherwise. If truncation and/or transformation parameters are provided these operations occur. Truncation implies that distances that exceed a threshold amount are assigned to 4 \* threshold. Minimum spanning tree truncation is the recommended default. Transformation performs an exponential or gaussian transformation of the distance matrix after truncation.

## Usage

```
mgMEM(locD, truncation = NULL, transformation = NULL)
```

## Arguments

locD	A symmetric distance matrix giving the distances (typically Euclidean) among the sampling locations of genetic material (e.g. of genotyped individuals or populations).
truncation	See details (EXPERIMENTAL)
transformation	Can be character "exponential" or "gaussian" or NULL for no transformation (EXPERIMENTAL)

## Details

If sampling locations are in longitude/latitude and are far apart, be sure to supply the geodesic distance as `locD`. (Note that `mgQuick` implements geodesic distances using the `longlat=TRUE` parameter when provided with sampling coordinates)

`truncation`

1. Can be numeric from 0 to 1 specifying the proportion of the maximum distance in `locD` to truncate following this a spanning tree is used to further truncate as in PCNM (aka `dbMEM` or classical MEM)
2. Can be NULL (default) indicating only the minimum spanning tree (MST) truncation where links that exceed the longest link in the MST (`dMST`) are replaced with  $4 * dMST$
3. Can be FALSE indicating that nothing is done to the distance matrix which is only suitable when `locD` is non-euclidean (i.e. will have negative eigenvectors)

## Value

A list

`$valuesMEM` gives the eigenvalues all MEM eigenvectors

`$vectorsMEM` gives the MEM eigenvectors in columns

## Author(s)

Pedro Peres-Neto ([peres-neto.pedro@uqam.ca](mailto:peres-neto.pedro@uqam.ca))

Paul Galpern ([pgalpern@ucalgary.ca](mailto:pgalpern@ucalgary.ca))

## References

Legendre, P., and Legendre L. 2012. Numerical Ecology, 3rd. ed. Elsevier, Amsterdam.

## Examples

```
## Prepare the radial data for analysis
radialData <- read.csv(system.file("extdata/radial.csv", package="memgene"))
radialXY <- radialData[, 1:2]

## Find MEM eigenvectors given sampling locations
## by first finding the Euclidean distance matrix
radialEuclid <- dist(radialXY)
radialMEM <- mgMEM(radialEuclid)
```

mgQuick

*Memgene analysis of genetic distance data (main interface for package)***Description**

Performs multiple–typical–steps in a memgene analysis of genetic distance data. Gracefully handles potential errors. Steps are as follows:

1. Find MEM eigenvectors given coordinates (coords)
2. Perform separate forward selections of positive and negative MEM eigenvectors against genetic distance (genD), to identify a significant subset, using parameters forwardPerm as the number of permutations and forwardAlpha as the alpha level for a significant eigenvector. NOTE: The number of permutations forwardPerm is set at 100 by default to reduce analysis time for exploratory analyses. This number should be increased for final analyses.
3. Find the fit of the selected eigenvectors to the genetic distance data (using RDA).
4. Optionally run a permutation test (finalPerm) for the fit of the selected eigenvectors to the genetic distance data.
5. Produce MEMGENE variables using the fitted values from the RDA analysis. MEMGENE variables are the eigenvectors from a PCA of the fitted values. These are the product of memgene and can be used for visualization and subsequent analyses.
6. Optionally produce plots of the scores for the first n MEMGENE variables if doPlot = n.

**Usage**

```
mgQuick(genD, coords, longlat = FALSE, truncation = NULL,
        transformation = NULL, forwardPerm = 100, forwardAlpha = 0.05,
        finalPerm = NULL, doPlot = NULL, verbose = TRUE)
```

**Arguments**

genD	A symmetrical distance matrix giving the genetic distances among individual genotypes or populations
coords	A two column matrix or data.frame of x and y coordinates of sampling locations of individual genotypes
longlat	If TRUE then coords are longitude and latitude, so find the geodesic distances among sampling locations using the geosphere package
truncation	NULL under typical usage. See <a href="#">mgMEM</a> for experimental options.
transformation	NULL under typical usage. See <a href="#">mgMEM</a> for experimental options.

forwardPerm	The number of permutations in the randomization test for the forward selection of MEM eigenvectors. The default forwardPerm=100 is sufficient for exploratory purposes, however this should be increased for final analyses.
forwardAlpha	The 1-alpha level for the forward selection process
finalPerm	The number of permutations for the final randomization test of the reduced model. NULL by default does not perform a final randomization test.
doPlot	Plot doPlot = n MEMGENE variables
verbose	If TRUE then report progress to the console

### Value

A list

- \$P gives the probability of the null hypothesis for the RDA on the final model
- \$RSqAdj is the adjusted R2 for the RDA, understood as the proportion of all genetic variation that is explicable by spatial pattern (i.e. spatial genetic signal)
- \$memgene contains a matrix with the MEMGENE variables in columns
- \$memSelected gives a matrix containing the selected MEM eigenvectors in columns
- \$whichSelectPos gives the indices of the selected MEM eigenvectors with positive eigenvalues (i.e. from \$mem)
- \$whichSelectNeg gives the indices of the selected MEM eigenvectors with negative eigenvalues (i.e. from \$mem)
- \$mem the output of [mgMEM](#) given coords

### Author(s)

Pedro Peres-Neto (peres-neto.pedro@uqam.ca)  
Paul Galpern (pgalpern@ucalgary.ca)

### References

Galpern, P., Peres-Neto, P., Polfus, J., and Manseau, M. 2014. MEMGENE: Spatial pattern detection in genetic distance data. Submitted.

### Examples

```
## Not run:
## Prepare the radial data for analysis
radialData <- read.csv(system.file("extdata/radial.csv", package="memgene"))
radialGen <- radialData[, -c(1,2)]
radialXY <- radialData[, 1:2]
if (require(adeigenet)) {
  radialDM <- codomToPropShared(radialGen)
} else {
  stop("adeigenet package required to produce genetic distance matrix in example.")
}

## Run the MEMGENE analysis
radialAnalysis <- mgQuick(radialDM, radialXY)
```

```
## Extract the scores on the first 3 MEMGENE variables
## for subsequent analysis
radialMEMGENE1 <- radialAnalysis$memgene[, 1]
radialMEMGENE2 <- radialAnalysis$memgene[, 2]
radialMEMGENE3 <- radialAnalysis$memgene[, 3]

## Find the proportion of variation explained by all MEMGENE variables
propVariation <- radialAnalysis$sdev/sum(radialAnalysis$sdev)

## End(Not run)
```

---

mgRDA

---

*Extraction of MEMGENE variables using redundancy analysis (RDA)*


---

## Description

Performs a redundancy analysis (RDA) given MEM eigenvectors and a genetic distance matrix. Optionally performs a permutation test for the RDA. Returns the MEMGENE variables, which are the product of a PCA conducted on the fitted values of this RDA.

## Usage

```
mgRDA(genD, vectorsMEM, perm = NULL, full = TRUE)
```

## Arguments

genD	A symmetrical distance matrix giving the genetic distances among individual genotypes or populations
vectorsMEM	A matrix giving a set of any number of MEM eigenvectors
perm	The number of permutations in a randomization test
full	If TRUE returns the MEMGENE variables. FALSE is used primarily by <a href="#">mgForward</a> which calls this function.

## Details

Any type of genetic distance matrix `genD` giving pairwise distances among individual genotypes could be used. Population genetic distances (e.g. pairwise  $F_{st}$  among populations) could also be used in principle, in which case the sampling centroids of populations should be used to develop the MEM eigenvectors.

## Value

A list:  
`$RsqAdj` is the adjusted  $R^2$  for the RDA, understood as the proportion of all genetic variation that is explicable by spatial pattern (i.e. spatial genetic signal)  
`$memgene` gives the MEMGENE variables ordered according to the eigenvalues which are given in `$sdev`

**Author(s)**

Pedro Peres-Neto (peres-neto.pedro@uqam.ca)  
 Paul Galpern (pgalpern@ucalgary.ca)

**Examples**

```
## Not run:
## Prepare the radial data for analysis
radialData <- read.csv(system.file("extdata/radial.csv", package="memgene"))
radialGen <- radialData[, -c(1,2)]
radialXY <- radialData[, 1:2]

if (require(adeigenet)) {
  radialDM <- codomToPropShared(radialGen)
} else {
  stop("adeigenet package required to produce genetic distance matrix in example.")
}

## Find MEM eigenvectors given sampling locations
## by first finding the Euclidean distance matrix
radialEuclid <- dist(radialXY)
radialMEM <- mgMEM(radialEuclid)

## Forward select significant MEM eigenvectors using RDA
## Positive MEM eigenvectors (positive spatial autocorrelation) first
radialPositive <- mgForward(radialDM,
  radialMEM$vectorMEM[, radialMEM$valueMEM > 0])
## Negative MEM eigenvectors (negative spatial autocorrelation) second
radialNegative <- mgForward(radialDM,
  radialMEM$vectorMEM[, radialMEM$valueMEM < 0])

## Summarize the selected MEM eigenvectors
allSelected <- cbind(radialMEM$vectorMEM[, radialMEM$valueMEM > 0][
  , na.omit(radialPositive$selectedMEM)],
  radialMEM$vectorMEM[, radialMEM$valueMEM < 0][
  , na.omit(radialNegative$selectedMEM)])

## Use the selected MEM eigenvectors in a final model
radialAnalysis <- mgRDA(radialDM, allSelected, full=TRUE)

## End(Not run)
```

**Description**

This function performs a variation partitioning of the genetic distance matrix using the supplied MEM eigenvectors and spatial coordinates. Randomization tests are conducted to determine the significance of the [a] fraction representing the MEM eigenvectors, the [c] fraction representing the spatial coordinates and the [abc] fraction representing the spatial genetic variation. It is called by [mgLandscape](#).

**Usage**

```
mgVarPart(genD, vectorsMEM, coords, perm=1000)
```

**Arguments**

genD	A symmetrical distance matrix giving the genetic distances among individual genotypes or populations
vectorsMEM	A matrix giving a set of any number of MEM eigenvectors
coords	A two column matrix or data.frame of x and y coordinates of sampling locations of individual genotypes or populations.
perm	The number of permutations to use when testing the significance of the [a], [c] and [abc] fractions.

**Details**

See [mgLandscape](#) for explanation of the fractions.

**Author(s)**

Pedro Peres-Neto (peres-neto.pedro@uqam.ca)  
Paul Galpern (pgalpern@ucalgary.ca)

**Examples**

```
## Not run:
## Prepare the radial data for analysis
radialData <- read.csv(system.file("extdata/radial.csv", package="memgene"))
radialGen <- radialData[, -c(1,2)]
radialXY <- radialData[, 1:2]

if (require(adeget)) {
  radialDM <- codomToPropShared(radialGen)
} else {
  stop("adeget package required to produce genetic distance matrix in example.")
}

## Find MEM eigenvectors given sampling locations
## by first finding the Euclidean distance matrix
radialEuclid <- dist(radialXY)
radialMEM <- mgMEM(radialEuclid)
```

```
## Forward select significant MEM eigenvectors using RDA
## Positive MEM eigenvectors (positive spatial autocorrelation) first
radialPositive <- mgForward(radialDM,
  radialMEM$vectorMEM[ , radialMEM$valueMEM > 0])
## Negative MEM eigenvectors (negative spatial autocorrelation) second
radialNegative <- mgForward(radialDM,
  radialMEM$vectorMEM[ , radialMEM$valueMEM < 0])

## Summarize the selected MEM eigenvectors
allSelected <- cbind(radialMEM$vectorMEM[, radialMEM$valueMEM > 0][
  , na.omit(radialPositive$selectedMEM)],
  radialMEM$vectorMEM[, radialMEM$valueMEM < 0][
  , na.omit(radialNegative$selectedMEM)])

## Use the selected MEM eigenvectors and coordinates in
## variation partitioning
radialVarPart <- mgVarPart(radialDM, allSelected, radialXY)

## End(Not run)
```

# Index

## \* **package**

memgene-package, [2](#)

ade4, [9](#)

codomToPropShared, [3](#)

memgene (memgene-package), [2](#)

memgene-package, [2](#)

mgForward, [4](#), [14](#)

mgLandscape, [5](#), [16](#)

mgMap, [8](#)

mgMEM, [6](#), [10](#), [12](#), [13](#)

mgQuick, [5](#), [9](#), [11](#), [12](#)

mgRDA, [4](#), [14](#)

mgVarPart, [15](#)

print.mgLandscape (mgLandscape), [5](#)

raster, [6](#)