Package ‘evolvability’

June 2, 2015

Type Package
Title Calculation of Evolvability Parameters
Version 1.1.0
Description An implementation of the evolvability parameters defined in Hansen and Houle (2008).
License GPL (>= 2)
Imports coda
NeedsCompilation no
Author Geir H. Bolstad [aut, cre]
Maintainer Geir H. Bolstad <geir.bolstad@nina.no>
Repository CRAN
Date/Publication 2015-06-02 16:51:34

R topics documented:

- evolvability-package .................................................. 2
- evolvabilityBeta ......................................................... 2
- evolvabilityBetaMCMC ................................................. 3
- evolvabilityBetaMCMC2 ............................................... 5
- evolvabilityMeans ..................................................... 6
- evolvabilityMeansMCMC ............................................... 8
- meanStdG ................................................................. 9
- meanStdGMCMC ........................................................ 10
- randomBeta .............................................................. 11
- summary.evolvabilityBeta .......................................... 12
- summary.evolvabilityBetaMCMC ................................... 12

Index 14
evolvability-package  Calculate parameters of evolvability

Description

This package calculates evolvability parameters from estimated G-matrices. It can use both point estimates and posterior/bootstrap distributions of variance matrices.

Details

Package: evolvability
Type: Package
Version: 1.1.0
Date: 2015-04-13
License: GPL (>= 2)

Author(s)

Geir H. Bolstad <geir.bolstad@nina.no>

References


evolvabilityBeta  Calculate evolvability parameters along a set of selection gradients

Description

evolvabilityBeta calculates (unconditional) evolvability (e), respondability (r), conditional evolvability (c), autonomy (a) and integration (i) along selection gradients given an additive-genetic variance matrix as described in Hansen and Houle (2008).

Usage

evolvabilityBeta(G, Beta, means = 1)
Arguments

\( G \) a variance matrix.
\( \text{Beta} \) either a vector or a matrix of unit length selection gradients stacked column wise.
\( \text{means} \) optional vector of trait means.

Value

\( \text{Beta} \) matrix of selection gradients
\( e \) evolvability of each selection gradient
\( r \) respondability of each selection gradient
\( c \) conditional evolvability of each selection gradient
\( a \) autonomy of each selection gradient
\( i \) integration of each selection gradient

Author(s)

Geir H. Bolstad <geir.h.bolstad@ntnu.no>

References


Examples

\[
G = \text{matrix}(c(1, 1, 0, 1, 2, 1, 0, 1, 2), \text{ncol} = 3)/10
\]
\[
\text{Beta} = \text{randomBeta}(5, 3)
\]
\[
X = \text{evolvabilityBeta}(G, \text{Beta})
\]
\[
\text{summary}(X)
\]

---

\textit{evolvabilityBetaMCMC} \text{Calculate posterior distribution of evolvability parameters from a set of selection gradients}

Description

\textit{evolvabilityBetaMCMC} calculates (unconditional) evolvability (e), respondability (r), conditional evolvability (c), autonomy (a) and integration (i) from selection gradients given the posterior distribution of an additive-genetic variance matrix. These measures and their meaning are described in Hansen and Houle (2008).

Usage

\texttt{evolvabilityBetaMCMC(G_mcmc, Beta, post.dist = FALSE)}
evolvabilityBetaMCMC

Arguments

G_mcmc  posterior distribution of a variance matrix in the form of a table. Each row in the table must be one iteration of the posterior distribution (or bootstrap distribution). Each iteration of the matrix must be on the form as given by \( \text{c}(x) \), where \( x \) is a matrix. A posterior distribution of a matrix in the slot VCV of an object of class MCMCg1mm is by default on this form.

Beta  either a vector or a matrix of unit length selection gradients stacked column wise.

post.dist  logical: should the posterior distribution of the evolvability parameters be saved.

Value

eB  posterior median and highest posterior density interval of evolvability for each selection gradient

rB  posterior median and highest posterior density interval of respondability for each selection gradient

cB  posterior median and highest posterior density interval of conditional evolvability for each selection gradient

aB  posterior median and highest posterior density interval of autonomy for each selection gradient

iB  posterior median and highest posterior density interval of integration for each selection gradient

Beta  matrix of selection gradients

summary  means of evolvability parameters across all selection gradients

post.dist  full posterior distributions

Author(s)

Geir H. Bolstad <geir.h.bolstad@ntnu.no>

References


Examples

# Simulating a posterior distribution (or bootstrap distribution) of a G-matrix:
G = matrix(c(1, 1, 0, 1, 4, 1, 0, 1, 2), ncol = 3)
G_mcmc = sapply(c(G), function(x) rnorm(10, x, 0.01))
G_mcmc = t(apply(G_mcmc, 1, function(x){
  G = matrix(x, ncol=sqrt(length(x)))
  G[lower.tri(G)] = t(G)[lower.tri(G)]
  c(G)
})}
evolvabilityBetaMCMC2

# Simulating a posterior distribution
# (or bootstrap distribution) of trait means:
means = c(1, 1.4, 2.1)
means_mcmc = sapply(means, function(x) rnorm(10, x, 0.01))

# Mean standardizing the G-matrix:
G_mcmc = meanStdGMCMC(G_mcmc, means_mcmc)

# Generating selection gradients in five random directions:
Beta = randomBeta(5, 3)

# Calculating evolvability parameters:
x = evolvabilityBetaMCMC2(G_mcmc, Beta, post.dist=TRUE)
summary(x)

---

evolvabilityBetaMCMC2  *Calculate posterior distribution of evolvability parameters from a selection gradient with uncertainty*

**Description**

evolvabilityBetaMCMC2 calculates (unconditional) evolvability (e), respondability (r), conditional evolvability (c), autonomy (a) and integration (i) along a selection gradient with uncertainty.

**Usage**

evolvabilityBetaMCMC2(G_mcmc, Beta_mcmc, post.dist = FALSE)

**Arguments**

- **G_mcmc**  posterior distribution of a variance matrix in the form of a table. Each row in the table must be one iteration of the posterior distribution (or bootstrap distribution). Each iteration of the matrix must be on the form as given by c(x), where x is a matrix. A posterior distribution of a matrix in the slot VCV of a object of class MCMCglm is by default on this form.
- **Beta_mcmc**  posterior distribution of a unit length selection gradient where iterations are given row wise.
- **post.dist**  logical: should the posterior distribution of the evolvability parameters be saved.

**Value**

- **Beta.median**  posterior median and highest posterior density interval of the selection gradient
- **summary**  posterior median and highest posterior density interval of evolvability parameters
- **post.dist**  full posterior distributions for the evolvability parameters
evolvabilityMeans

Calculate average evolvability parameters of a G-matrix

description

EvolvabilityMeans calculates the average (unconditional) evolvability (e), respondability (r), conditional evolvability (c), autonomy (a) and integration (i) of a additive-genetic variance matrix using the approximation formulas described in Hansen and Houle (2008, 2009).

usage

evolvabilityMeans(G, means = 1)
Arguments

- `G`: a variance matrix.
- `means`: optional vector of trait means, for mean standardization.

Details

The formulas for calculating the evolvability parameters are approximations, except for the formula for unconditional evolvability which is exact. The bias of the approximations depends on the dimensionality of the G-matrix, with higher bias for few dimensions (see Hansen and Houle 2008). In most instances we recommend to estimate the averages of the evolvability parameters using `evolvabilityBeta` on a large number of random selection gradients. The maximum and minimum evolvability, which are also the maximum and minimum respondability and conditional evolvability, equals the largest and smallest eigenvalue of the G-matrix, respectively.

Value

- `e_mean`: average (unconditional) evolvability
- `e_min`: minimum evolvability
- `e_max`: maximum evolvability
- `r_mean`: average respondability
- `c_mean`: average conditional evolvability
- `a_mean`: average autonomy
- `i_mean`: average integration

Author(s)

Geir H. Bolstad <geir.h.bolstad@ntnu.no>

References


Examples

```r
G = matrix(c(1, 1, 0, 1, 2, 1, 0, 1, 2), ncol = 3)
evolvabilityMeans(G)
```
evolvabilityMeansMCMC  Calculate posterior distribution of average evolvability parameters of a G-matrix

Description

evolvabilityMeans calculates the average (unconditional) evolvability (e), respondability (r), conditional evolvability (c), autonomy (a) and integration (i) given the posterior distribution of an additive-genetic variance matrix using the approximation formulas described in Hansen and Houle (2008, 2009).

Usage

evolvabilityMeansMCMC(G_mcmc)

Arguments

G_mcmc  posterior distribution of a variance matrix in the form of a table. Each row in the table must be one iteration of the posterior distribution (or bootstrap distribution). Each iteration of the matrix must be on the form as given by \( c(x) \), where \( x \) is a matrix. A posterior distribution of a matrix in the slot VCV of a object of class MCMCglmm is by default on this form.

Details

The formulas for calculating the evolvability parameters are approximations, except for the formula for unconditional evolvability which is exact. The bias of the approximations depends on the dimensionality of the G-matrix, with higher bias for few dimensions (see Hansen and Houle 2008). In most instances we recommend to estimate the averages of the evolvability parameters using evolvabilityBetamCMC on a large number of random selection gradients. The maximum and minimum evolvability, which are also the maximum and minimum respondability and conditional evolvability, equals the largest and smallest eigenvalue of the G-matrix, respectively.

Value

post_dist  posterior distribution of the average evolvability parameters
post_medians  posterior medians and HPD interval of the average evolvability parameters

Author(s)

Geir H. Bolstad <geir.h.bolstad@ntnu.no>

References

Examples

# Simulating a posterior distribution
# (or bootstrap distribution) of a G-matrix:
G = matrix(c(1, 1, 0, 1, 0, 1, 2), ncol = 3)
G_mcmc = sapply(c(G), function(x) rnorm(10, x, 0.01))
G_mcmc = t(apply(G_mcmc, 1, function(x){
    G = matrix(x, ncol=sqrt(length(x)))
    G[lower.tri(G)] = t(G)[lower.tri(G)]
    c(G)
}))

# Simulating a posterior distribution
# (or bootstrap distribution) of trait means:
means = c(1, 1.4, 2.1)
means_mcmc = sapply(means, function(x) rnorm(10, x, 0.01))

# Mean standardizing the G-matrix:
G_mcmc = meanstdGMC(G_mcmc, means_mcmc)

# Estimating average evolvability parameters:
evolvabilityMeansMCMC(G_mcmc)

meanstdG

Mean standardize a G-matrix

Description

Mean standardization of a G-matrix.

Usage

meanstdG(G, means)

Arguments

G a variance matrix.
means a vector of trait means.

Value

meanstdG returns a mean standardized variance matrix.

Author(s)

Geir H. Bolstad
Examples

```r
G = matrix(c(1, 1, 0, 1, 4, 1, 0, 1, 2), ncol = 3)
means = c(1, 1.4, 2.1)
meanStdG(G, means)
```

Description

Mean standardization of the posterior distribution of a G-matrix

Usage

```r
meanStdGMCMC(G_mcmc, means_mcmc)
```

Arguments

- `G_mcmc`: posterior distribution of a variance matrix in the form of a table. Each row in the table must be one iteration of the posterior distribution (or bootstrap distribution). Each iteration of the matrix must be on the form as given by `c(x)`, where `x` is a matrix. A posterior distribution of a matrix in the slot `VCV` of a object of class `MCMCglmm` is by default on this form.

- `means_mcmc`: posterior distribution of a vector of means in the form of a table. Each row in the table must be one iteration of the posterior distribution (or bootstrap distribution). A posterior distribution of a mean vector in the slot `Sol` of a object of class `MCMCglmm` is by default on this form.

Value

`meanStdGMCMC` returns the posterior distribution of a mean standardized variance matrix.

Author(s)

Geir H. Bolstad <geir.h.bolstad@ntnu.no>

Examples

```r
# Simulating a posterior distribution
# (or bootstrap distribution) of a G-matrix:
G = matrix(c(1, 1, 0, 1, 4, 1, 0, 1, 2), ncol = 3)
G_mcmc = sapply(c(G), function(x) rnorm(10, x, 0.01))
G_mcmc = t(apply(G_mcmc, 1, function(x){
  G = matrix(x, ncol=sqrt(length(x))))
  G[lower.tri(G)] = t(G)[lower.tri(G)]
  c(G)
})
```

randomBeta

# Simulating a posterior distribution
# (or bootstrap distribution) of trait means:
means = c(1, 1.4, 2.1)
means_mcmc = sapply(means, function(x) rnorm(10, x, 0.01))

# Mean standardizing the G-matrix:
meanStdGmcmc(G_mcmc, means_mcmc)

randomBeta  Generating selection gradients/vectors in random directions.

Description

randomBeta generates unit length vectors (selection gradients) uniformly distributed in a k-dimensional hypersphere.

Usage

randomBeta(n = 1, k = 2)

Arguments

n  Number of selection gradients/vectors.
k  Number of dimensions.

Details

randomBeta exploits the spherical symmetry of a multidimensional Gaussian density function. Each element of each vector is randomly sampled from a univariate Gaussian distribution with zero mean and unit variance. The vector is then divided by its norm to standardize it to unit length.

Value

randomBeta returns a matrix where the vectors are stacked column wise.

Author(s)

Geir H. Bolstad

Examples

## Two vectors of dimension 3:
randomBeta(n = 2, k = 3)
summary.evolvabilityBeta

*Summarizing evolvability parameters over a set of selection gradients*

**Description**

summary method for class "evolvabilityBeta". The returned object is suitable for printing with the `print.summary.evolvabilityBeta` method.

**Usage**

```r
## S3 method for class 'evolvabilityBeta'
summary(object, ...)
```

**Arguments**

- `object`: an object of class "evolvabilityBeta"
- `...`: additional arguments affecting the summary produced.

**Value**

- **Averages**: the averages of the evolvability parameters over all selection gradients
- **Minimum**: the minimum of the evolvability parameters over all selection gradients
- **Maximum**: the maximum of the evolvability parameters over all selection gradients

**Author(s)**

Geir H. Bolstad <geir.h.bolstad@ntnu.no>

**See Also**

evolvabilityBeta

summary.evolvabilityBetaMCMC

*Summarizing posterior distribution of evolvability parameters over a set of selection gradients*

**Description**

summary method for class "evolvabilityBetaMCMC". The returned object is suitable for printing with the `print.summary.evolvabilityBetaMCMC` method.
Usage

```r
## S3 method for class 'evolvabilityBetaMCMC'
summary(object, ...)
```

Arguments

- `object`: an object of class "evolvabilityBetaMCMC"
- `...`: additional arguments affecting the summary produced.

Value

- **Averages**: the averages of the evolvability parameters over all selection gradients
- **Minimum**: the minimum (given by the posterior median) of the evolvability parameters over all selection gradients
- **Maximum**: the maximum (given by the posterior median) of the evolvability parameters over all selection gradients

Author(s)

Geir H. Bolstad <geir.h.bolstad@ntnu.no>

See Also

evolvabilityBetaMCMC
Index

*Topic algebra
  evolvabilityBeta, 2
  evolvabilityBetaMCMC, 3
  evolvabilityBetaMCMC2, 5
  evolvabilityMeans, 6
  evolvabilityMeansMCMC, 8
  meanStdG, 9
  meanStdGMCMC, 10
  summary.evolvabilityBeta, 12
  summary.evolvabilityBetaMCMC, 12

*Topic array
  evolvabilityBeta, 2
  evolvabilityBetaMCMC, 3
  evolvabilityBetaMCMC2, 5
  evolvabilityMeans, 6
  evolvabilityMeansMCMC, 8
  meanStdG, 9
  meanStdGMCMC, 10
  summary.evolvabilityBeta, 12
  summary.evolvabilityBetaMCMC, 12

*Topic multivariate
  evolvabilityBetaMCMC, 3
  evolvabilityBetaMCMC2, 5
  meanStdGMCMC, 10

*Topic package
  evolvability-package, 2
  evolvability (evolvability-package), 2
  evolvability-package, 2
  evolvabilityBeta, 2, 12
  evolvabilityBetaMCMC, 3, 13
  evolvabilityBetaMCMC2, 5
  evolvabilityMeans, 6
  evolvabilityMeansMCMC, 8
  meanStdG, 9
  meanStdGMCMC, 10

  print.evolvabilityBetaMCMC
    (summary.evolvabilityBetaMCMC), 12

  print.evolvabilityBetaMCMC2
    (evolvabilityBetaMCMC2), 5

  print.evolvabilityMeansMCMC
    (evolvabilityMeansMCMC), 8

  print.summary.evolvabilityBeta
    (summary.evolvabilityBeta), 12

  print.summary.evolvabilityBetaMCMC
    (summary.evolvabilityBetaMCMC), 12

  randomBeta, 11

  summary.evolvabilityBeta, 12
  summary.evolvabilityBetaMCMC, 12