Package: noia (via r-universe)

October 31, 2024

Type Package

Title Implementation of the Natural and Orthogonal InterAction (NOIA) Model

Version 0.97.3

Date 2023-03-07

Author Arnaud Le Rouzic (2007-2015), Arne B. Gjuvsland (2010), Olivier Ariste (2010)

Maintainer Arnaud Le Rouzic

<arnaud.le-rouzic@universite-paris-saclay.fr>

Depends stats

Suggests parallel, numDeriv

LazyData yes

URL https://github.com/lerouzic/noia

Description The NOIA model, as described extensively in Alvarez-Castro & Carlborg (2007), is a framework facilitating the estimation of genetic effects and genotype-to-phenotype maps. This package provides the basic tools to perform linear and multilinear regressions from real populations (provided the phenotype and the genotype of every individuals), estimating the genetic effects from different reference points, the genotypic values, and the decomposition of genetic variances in a multi-locus, 2 alleles system. This package is presented in Le Rouzic & Alvarez-Castro (2008).

License GPL-2

Repository https://lerouzic.r-universe.dev

RemoteUrl https://github.com/lerouzic/noia

RemoteRef HEAD

RemoteSha b46988691b211982d60d57ffa67733d7b5b1089e

Contents

`Genetic effects` *Genetic Effects*

Description

`geneticEffects` displays the genetic effects (and their standard errors) from the result of `linearRegression`. If a new reference point is provided, a "change of reference" operation is performed (Alvarez-Castro and Carlborg 2007).

Usage

```
geneticEffects(obj, reference="P1", ref.genotype = NULL)
```

Arguments

`obj`	An object of class `"noia.linear"` provided by `linearRegression`.
`reference`	The new reference point. Can be `"F2"`, `"F1"`, `"Finf"`, `"P1"`, `"P2"` (see `linearRegression` for details.
`ref.genotype`	The same as `reference`, provided for compatibility with older versions.

Details

Variance decomposition and change of reference operation are not possible from the result of a multilinear regression.

Author(s)

Arnaud Le Rouzic

References

Alvarez-Castro JM, Carlborg O. (2007). A unified model for functional and statistical epistasis and its application in quantitative trait loci analysis. Genetics 176(2):1151-1167.

Le Rouzic A, Alvarez-Castro JM. (2008). Estimation of genetic effects and genotype-phenotype maps. Evolutionary Bioinformatics, 4.

See Also

`linearRegression`, `multilinearRegression`.

Examples

```
map <- c(0.25, -0.75, -0.75, -0.75, 2.25, 2.25, -0.75, 2.25, 2.25)
pop <- simulatePop(map, N=500, sigmaE=0.2, type="F2")

# Regressions

linear <- linearRegression(phen=pop$phen, gen=cbind(pop$Loc1, pop$Loc2))

geneticEffects(linear, "P1")
```

`Genetic regression` *Linear and Multilinear Genetic Regressions*

Description

The regression aims at estimating genetic effects from a population in which the genotypes and phenotypes are known.

Usage

```
linearRegression(phen, gen=NULL, genZ=NULL,
    reference="noia", max.level=NULL, max.dom=NULL, fast=FALSE)
multilinearRegression(phen, gen=NULL, genZ=NULL,
    reference="noia", max.level=NULL, max.dom=NULL, fast=FALSE,
    e.unique=FALSE, start.algo = "linear", start.values=NULL,
    robust=FALSE, bilinear.steps=1, ...)
```

Arguments

`phen`	The vector of individual phenotypes measured in the population.
`gen`	The matrix of individual genotypes in the population, one column per locus. See `genNames` for the genotype encoding. Not necessary if `genZ` is provided.
`genZ`	The matrix of individual genotypic probabilities in the population, 3 columns per locus, corresponding of the probability of each of the 3 genotypes (the sum must be 1). Not necessary if `gen` is provided.

reference	The reference point from which the regression is performed. By default, the "noia" reference point is used, since it provides a fairly good orthogonality. Other possibilities are "G2A", "F2", "F1", "Finf", "UWR", "P1" and "P2".
max.level	Maximum level of interactions.
max.dom	Maximum level for dominance effects. Does not have any effect if >= max.level. In the multilinear regression, the maximum level for dominance effects cannot be > 1.
fast	This "fast" algorithm should be used when (i) the number of loci is high (> 8) and (ii) there are uncertainties in the dataset (missing values or Haley-Knott regression). This algorithm computes the regression matrix directly function, i.e. without computing Z nor S matrices.
e.unique	Whether the multilinear term is the same for all pairs.
start.algo	Algorithm used to compute the starting values. Can be "linear", "multilinear", "subset" or "bilinear". Ignored if start.values are provided.
start.values	Vector of starting values.
robust	Tries sequentially all starting values algorithms.
bilinear.steps	Number of steps. Ignored if start.algo is not "bilinear". If NULL, the bilinear algorithm is run until (almost) convergence.
...	Extra parameters to the non-linear regression function nls, including nls.control.

Details

If a gen data set is provided, it will be turned into a genZ. Missing data (unknown genotypes) are considered as loci for which genotypic probabilities are identical to the genotypic frequencies in the population.

The algebraic framework is described extensively in Alvarez-Castro & Carlborg 2007. The default reference point ("noia") provides an orthogonal decomposition of genetic effects in the 1-locus case, whatever the genotypic frequencies. It remains a good approximation of orthogonality in the multi-locus case if linkage disequilibrium is small. Other optional reference points are those of the "G2A" model (Zeng et al. 2005), and the unweighted regression model "UWR" (Cheverud & Routman, 1995). Several key populations can be taken as reference as well: "F2", "F1", "Finf" (F infinity), and the two "parental" homozygous populations "P1" and "P2".

The multilinear model for genetic interactions is an alternative way to model epistatic interactions between at least two loci (see Hansen & Wagner 2001). The computation of multilinear estimates requires a non-linear regression step that relies on the nls function. Providing good starting values for the non-linear regression is a key to ensure convergence, and different algorithms are provided, that can be specified by the "start.algo" option. "linear" performs a linear regression and approximates the genetic effects from it, while "multilinear" performs a simpler multilinear regression (without dominance) to initialize the genetic effects. "subset" estimate all genetic effects from a random subset (50%) of the population, and "bilinear" estimate alternatively marginal and epistatic effects.

Value

linearRegression and multilinearRegression return an object of class "noia.linear" or "noia.multilinear", both having their own print methods: print.noia.linear and print.noia.multilinear.

Author(s)

Arnaud Le Rouzic

References

Alvarez-Castro JM, Carlborg O. (2007). A unified model for functional and statistical epistasis and its application in quantitative trait loci analysis. Genetics 176(2):1151-1167.

Alvarez-Castro JM, Le Rouzic A, Carlborg O. (2008). How to perform meaningful estimates of genetic effects. PLoS Genetics 4(5):e1000062.

Cheverud JM, Routman, EJ. (1995). Epistasis and its contribution to genetic variance components. Genetics 139:1455-1461.

Hansen TF, Wagner G. (2001) Modeling genetic architecture: A multilinear theory of gene interactions. Theoretical Population Biology 59:61-86.

Le Rouzic A, Alvarez-Castro JM. (2008). Estimation of genetic effects and genotype-phenotype maps. Evolutionary Bioinformatics 4.

Zeng ZB, Wang T, Zou W. (2005). Modelling quantitative trait loci and interpretation of models. Genetics 169: 1711-1725.

See Also

`geneticEffects`, `GPmap`, `varianceDecomposition`.

Examples

```
set.seed(123456789)

map <- c(0.25, -0.75, -0.75, -0.75, 2.25, 2.25, -0.75, 2.25, 2.25)
pop <- simulatePop(map, N=500, sigmaE=0.2, type="F2")

# Regressions

linear <- linearRegression(phen=pop$phen, gen=cbind(pop$Loc1, pop$Loc2))

multilinear <- multilinearRegression(phen=pop$phen,
    gen=cbind(pop$Loc1, pop$Loc2))

# Linear effects, associated variances and stderr
linear

# Multilinear effects
multilinear
```

`Genotype-to-Phenotype map` *Genotype-to-Phenotype Mapping*

Description

The Genotype-to-Phenotype map is a vector providing the estimate of the genotypic value for any multi-locus genotype. The estimates may be computed from `linearRegression` or `multilinearRegression`.

Usage

```
GPmap(obj)
```

Arguments

`obj`	An object of class `"noia.linear"` or `"noia.multilinear"`.

Value

Returns a matrix with two columns: the first one is the estimate of genotypic effects, the second one the standard error of this estimate.

Author(s)

Arnaud Le Rouzic

References

Le Rouzic A, Alvarez-Castro JM. (2008). Estimation of genetic effects and genotype-phenotype maps. Evolutionary Bioinformatics, 4.

See Also

`linearRegression`, `multilinearRegression`, `genNames`.

Examples

```
set.seed(123456789)

map <- c(0.25, -0.75, -0.75, -0.75, 2.25, 2.25, -0.75, 2.25, 2.25)
pop <- simulatePop(map, N=500, sigmaE=0.2, type="F2")

# Regression
linear <- linearRegression(phen=pop$phen, gen=cbind(pop$Loc1, pop$Loc2))

# GP map
GPmap(linear)
```

`GP map analysis` *Noia analysis of genotype-to-phenotype (GP) maps in ideal populations*

Description

Functions for doing a NOIA analysis of a GP map for L loci in a population where the loci are in complete linkage equilibrium.

Usage

```
linearGPmapanalysis(gmap, reference="F2", freqmat=NULL,
                    max.level=NULL , S_full=NULL)
```

Arguments

`gmap`	Vector of length 3^L with genotypic values for all possible genotypes in the order defined by `genNames`.
`reference`	The reference population in which the analysis is done. By default, the `"F2"` population is used. Other possibilities are `"noia"`, `"G2A"`, `"UWR"`.
`freqmat`	For `reference="G2A"`: A vector of length L containing allele frequencies such that `freqmat[i]=frequency(allele 1)` for locus `i`. For `reference="noia"`: A $(L \times 3)$ matrix of genotype frequencies such that `freqmat[i,]=[frequency(1) frequency(2) frequency(3)]` for locus `i`.
`max.level`	Maximum level of interactions.
`S_full`	Boolean argument indicating whether to keep full `S` matrix $(3^L \times 3^L)$ in memory or alternatively to keep L single locus `S` matrices (3×3) and compute single row and columns of the full matrix.

Details

The algebraic framework is described extensively in Alvarez-Castro & Carlborg 2007. When analysing GP maps in ideal populations we can work directly with the `S` matrix and do not have to consider the `X` and `Z` matrices used in `linearRegression`. When it comes to the `S_full` argument keeping the multilocus `S` matrix in memory is generally fastest for computing all 3^L genetic effects. However it does not allow for computing only a subset of the effects and also runs out of memory for $L > 8$ on a typical desktop machine. For S_full=NULL in `linearGPmapanalysis` a full `S` matrix is used if $L <= 8$ and max.level=NULL, while L single locus `S` matrices are used otherwise.

Value

`linearGPmapanalysis` returns an object of class `"noia.linear.gpmap"` , with its own `print` method: `print.noia.linear.gpmap`.

Author(s)

Arne B. Gjuvsland

References

Alvarez-Castro JM, Carlborg O. (2007). A unified model for functional and statistical epistasis and its application in quantitative trait loci analysis. Genetics 176(2):1151-1167.

Cheverud JM, Routman, EJ. (1995). Epistasis and its contribution to genetic variance components. Genetics 139:1455-1461.

Le Rouzic A, Alvarez-Castro JM. (2008). Estimation of genetic effects and genotype-phenotype maps. Evolutionary Bioinformatics 4.

Zeng ZB, Wang T, Zou W. (2005). Modelling quantitative trait loci and interpretation of models. Genetics 169: 1711-1725.

See Also

varianceDecomposition

Examples

```
map <- c(0.25, -0.75, -0.75, -0.75, 2.25, 2.25, -0.75, 2.25, 2.25)

# Genotype-to-phenotype map analysis
linearGP <- linearGPmapanalysis(map, reference="F2")

# Linear effects in ideal F2 population
linearGP
```

Marginal locus calculation

Estimation of parameters for specific allele frequencies

Description

This function computes some parameters of interest (mean phenotype, genetic variance, additive variance, and evolutionary change in additive variance) for a combination of allele frequencies, based on a genotype-phenotype map.

Usage

```
marginallocus(gmap, freq=NULL, what="mean", definition=11, mc.cores=1, ...)
## S3 method for class 'noia.marloc'
plot(x, xlab=NULL, ylim=NULL, ylab=attr(x, "what"), ...)
## S3 method for class 'noia.marloc'
image(x, xlab=NULL, ylab=NULL, zlim=NULL,
      main=attr(x, "what"), col.max="red", col.min="blue", col.zero="white",
     n.cols=1000, zeropart=0.01, contour.levels=10, contour.options=list(), ...)
```

Arguments

gmap	Either an object of class noia.gpmap, or a vector of phenotypic values in the order defined as in genotypesNames
.	
freq	A vector indicating the loci that should be analysed. See Details.
what	A character string among "mean", "varA", "varG", or "dvarA.dt".
definition	The number of allele frequencies to try for each locus.
mc.cores	If more than 1, the calculation is run on mc.cores cores via the library parallel.
x	An object of class noia.marloc obtained after running marginallocus.
col.max, col.min, col.zero	Colors standing for the maximal, minimal, and nil values, respectively. Setting col.zero to NULL generates a color gradient between col.min and col.max.
n.cols	Number of colors in the gradient.
zeropart	Width (relative to the full amplitude) of the region around zero which will be colored as col.zero.
contour.levels	Number of contour lines. Setting this to 0 leads to no contour lines.
contour.options	List of additional options to the contour function.
xlab, ylab, ylim, zlim, main	Classical parameters passed to plot and image.
...	Additional parameters to internal functions.

Details

marginallocus computes a population parameter for a series of allele frequencies. The loci under investigation are provided through the freq vector, which need to have as many elements as loci in the system. Values of the freq vector indicate fixed allele frequencies, while NA indicate loci under investigation. For instance, freq=c(NA, 1, NA, 0.5), will investigate the effect of varying loci 1 and 3, while keeping loci 2 and 4 at constant allele frequencies. The population is assumed to be at Hardy-Weinberg frequencies. If freq is not provided, all loci will be investigated.

Value

marginallocus returns an array with as many dimensions as loci under investigation. This array is an object of class "noia.marloc" which can be graphically illustrated through the provided plot (for 1-dimensional data) and image (for 2-dimensional data). Arrays of higher dimensionality cannot be represented graphically.

Author(s)

Arnaud Le Rouzic

See Also

linearGPmapanalysis

Examples

```
map <- c(0.25, -0.75, -0.75, -0.75, 2.25, 2.25, -0.75, 2.25, 2.25)

mrg2D <- marginallocus(map)
mrg1D <- marginallocus(map, freq=c(NA, 0)) # the second locus is fixed for allele 1

image(mrg2D)
plot(mrg1D)
```

NOIA package — *Implementation of the Natural and Orthogonal InterAction (NOIA) model*

Description

The NOIA model, as described extensively in Alvarez-Castro & Carlborg (2007), is a framework facilitating the estimation of geneticEffects and genotype-to-phenotype maps. This package provides the basic tools to perform linear and multilinear regressions from real populations, analyse pure genotype-to-phenotype (GP) maps in ideal populations, estimating the genetic effects from different reference points, the genotypic values, and the decomposition of genetic variances in a multi-locus, 2 alleles system. This package is extensively described in Le Rouzic & Alvarez-Castro (2008).

Details

Package:	noia
Type:	Package
Version:	0.94.1
Date:	2010-04-20
License:	GPL-2

Regression data set: The user must provide (i) The vector of phenotypes of all individuals measured in the population, and (ii) The matrix of the genotypes. There are two input formats for the genotype, see `linearRegression`.

Regression functions: `linearRegression` and `multilinearRegression`.

GP map data set: The user must provide (i) The 3^L (where L is the number of loci) vector of genotypic values (**G** in Alvarez-Castro & Carlborg (2007)) (ii) Allele or genotype frequencies in the reference population.

GP map analysis function: `linearGPmapanalysis`.

Change of reference: `geneticEffects`.

Genotype-to-phenotype map: `GPmap`.

Decomposition of genetic variance: `varianceDecomposition`.

Author(s)

Arnaud Le Rouzic, Arne B. Gjuvsland

Maintainer: Arnaud Le Rouzic

References

Alvarez-Castro JM, Carlborg O. (2007). A unified model for functional and statistical epistasis and its application in quantitative trait loci analysis. Genetics 176(2):1151-1167.

Alvarez-Castro JM, Le Rouzic A, Carlborg O. (2008). How to perform meaningful estimates of genetic effects. PLoS Genetics 4(5):e1000062.

Le Rouzic A, Alvarez-Castro JM. (2008). Estimation of genetic effects and genotype-phenotype maps. Evolutionary Bioinformatics 4.

Examples

```
set.seed(123456789)

map <- c(0.25, -0.75, -0.75, -0.75, 2.25, 2.25, -0.75, 2.25, 2.25)
names(map) <- genNames(2)
pop <- simulatePop(map, N=500, sigmaE=0.2, type="F2")

# Regressions

linear <- linearRegression(phen=pop$phen, gen=pop[2:3])

multilinear <- multilinearRegression(phen=pop$phen, gen=cbind(pop$Loc1,
 pop$Loc2))

# Linear effects, associated variances and stderr
linear

# Multilinear effects
multilinear

# Genotype-to-phenotype map analysis
linearGP <- linearGPmapanalysis(map, reference="F2")

# Linear effects in ideal F2 population
linearGP

# Change of reference: geneticEffects in the "11" genotype (parental 1)
geneticEffects(linear, ref.genotype="P1")

# Variance decomposition
varianceDecomposition(linear)
varianceDecomposition(linearGP)

# GP maps
maps <- cbind(map, GPmap(linear)[,1], GPmap(multilinear)[,1])
colnames(maps) <- c("Actual", "Linear", "Multilinear")
maps
```

plot.noia *Graphical display of genetic regressions and genotype-phenotype maps*

Description

These functions allow a graphic representation of the result of genetic regressions from linearRegression and GPmap.

Usage

```
## S3 method for class 'noia.linear'
plot(x, loc = 1:x$nloc, effect=TRUE, epistasis = TRUE,
ylim=range(GPmap(x)[,1]) + c(-1,1)*max(GPmap(x)[,2]), ...)
## S3 method for class 'noia.gpmap'
barplot(height, GPcol = c("indianred", "palegreen", "royalblue"),
arrowscol = "purple", stderr = TRUE , main=NA, ylab=NA, ...)
```

Arguments

x	An object of class "noia.linear" for the plot function, or of class "noia.gpmap" for the barplot function.
loc	The vector loci to plot (by default, all of them are displayed).
effect	Whether genetic effects have to be plotted for each locus.
epistasis	Whether pairwise effects have to be plotted.
height	An object of class "noia.gpmap".
GPcol	Colors for each of the three genotypes.
arrowscol	Color of the error bars.
stderr	If TRUE, error bars stand for starndard errors. Otherwise, error bars are 95% condidence intervals.
main	The same as in plot.
ylab	The same as in plot.
ylim	The same as in plot.
...	Additional options for the plot and barplot routines.

Author(s)

Olivier Ariste, Arnaud Le Rouzic

print.noia *Printing Genetic Regressions and GP map analyses*

Description

Display the output of functions linearRegression, multilinearRegression and linearGPmapanalysis

Usage

```
## S3 method for class 'noia.linear'
print(x, ...)
## S3 method for class 'noia.multilinear'
print(x, ...)
## S3 method for class 'noia.common'
print(x, ...)
## S3 method for class 'noia.linear.gpmap'
print(x, ...)
```

Arguments

x	An object of class "noia.linear", class "noia.linear.gpmap" or class "noia.multilinear".
...	No effect for the moment.

Details

The print method being actually very similar for the linear and multilinear regressions, both call the common method print.noia.common.

Author(s)

Arnaud Le Rouzic, Arne B. Gjuvsland

References

Le Rouzic A, Alvarez-Castro JM. (2008). Estimation of genetic effects and genotype-phenotype maps. Evolutionary Bioinformatics, 4.

`Simulate population` *Simulates a Population from a Genotype-Phenotype Map*

Description

The `simulatePop` function takes a Genotype-to-Phenotype map (i.e. a vector defining the genotypic value of all possible genotypes) and returns a data frame containing the simulated population.

Usage

```
simulatePop(gmap, N = 100, sigmaE = 1, type = "F2", freqmat=NULL)
```

Arguments

`gmap`	The Genotype-to-phenotype map: a vector of size 3^L, where L is the number of loci. The vector should be named with the code of each genotype (see `genNames`.
`N`	Number of individuals.
`sigmaE`	Standard deviation of the environmental noise (normally distributed).
`type`	Type of population. `"F2"`, `"Finf"`, `"F1"`, `"UWR"`, `"G2A"`, and `"noia"` are possible.
`freqmat`	For `type="G2A"`: A vector of length `nloc` containing allele frequencies such that `freqmat[i]=frequency(allele 1)` for locus `i`. For `type="noia"`: A (`nloc` x `3`) matrix of genotype frequencies such that `freqmat[i,]=[frequency(1) frequency(2) frequency(3)]` for locus `i`.

Details

The type of population refers to the expected allelic and genotypic frequences:

- "F1"First generation of an intercross between two parental populations fixed for alleles A and B respectively; expected genotypic frequencies are: AA: 0, AB: 1, BB: 0.
- "F2"Second generation of an intercross between two parental populations fixed for alleles A and B respectively; expected genotypic frequencies are AA: 0.25, AB: 0.5, BB: 0.25.
- "Finf"Theoretical population from an infinite number of generations after an intercross between two parental populations fixed for alleles A and B respectively; expected genotypic frequencies are AA: 0.5, AB: 0, BB: 0.5.
- "UWR"Theoretical population corresponding to ideal (but experimentally unrealistic) equal genotypic frequencies; expected genotypic frequencies are AA: 0.333, AB: 0.333, BB: 0.333. In such a population, the "UnWeighted Regression model" (UWR) by Cheverud and Routman 1995 provides orthogonal estimates.
- "G2A"Population at Hardy-Weinberg frequencies; expected genotypic frequencies are: AA: p*p, AB: 2p(1-p), BB: (1-p)(1-p), the frequency of allele A (p) at locus i being provided by the i-th element of vector `freqmat`. "G2A" is the name of the statistical model by Zeng et al. (2005) in which genetic effects estimated from such a population are orthogonal.

- "noia"Population in which genotypic frequencies are arbitrary; expected genotypic frequencies are: AA: pAA, AB: pAB, BB: pBB, frequences pAA, pAB, and pBB at locus i being provided by the i-th line of matrix `freqmat`. "noia" is the name of the statistical model by Alvarez-Castro and Carlborg (2007) in which genetic effects estimated from such a population are orthogonal. In all populations, loci are considered as independent and are at linkage equilibrium.

Value

Returns a data frame, in which the first column (`$phen`) contains the phenotypes, and the following ones (`$Loc1`, `$loc2`, etc) the genotypes of all individuals.

Author(s)

Arnaud Le Rouzic, Arne B. Gjuvsland

References

Alvarez-Castro JM, Carlborg O. (2007). A unified model for functional and statistical epistasis and its application in quantitative trait loci analysis. Genetics 176(2):1151-1167.

Cheverud JM, Routman, EJ. (1995). Epistasis and its contribution to genetic variance components. Genetics 139:1455-1461.

Le Rouzic A, Alvarez-Castro JM. (2008). Estimation of genetic effects and genotype-phenotype maps. Evolutionary Bioinformatics, 4.

Zeng ZB, Wang T, Zou W. (2005). Modelling quantitative trait loci and interpretation of models. Genetics 169: 1711-1725.

See Also

`GPmap`, `genNames`

Examples

```
set.seed(123456789)

map <- c(0.25, -0.75, -0.75, -0.75, 2.25, 2.25, -0.75, 2.25, 2.25)
pop <- simulatePop(map, N=500, sigmaE=0.2, type="F2")
str(pop)

## Create a "noia" population with genotype frequencies 1/3,1/3,1/3 for locus 1
## and 0.2,0.6,0.2 for locus 2
pop = simulatePop(map, N=1000, sigma=1, type='noia',
  freqmat=matrix(c(1/3,1/3,1/3,0.2,0.6,0.2),nrow=2, byrow=TRUE))
```

`Variance decomposition` *Decomposition of Genetic Variance*

Description

Variance decomposition in a classical operation in quantitative genetics (e.g. Fisher 1918, Lynch and Walsh 1998). The genetic variance, i.e. the part of phenotypic variance that can be identify as due to genetic factors, can be decomposed into several orthogonal components (generally, the part due to additive factors Var(A), to dominance factors Var(D), and to genetic interactions Var(I)).

Usage

```
varianceDecomposition(obj)
## S3 method for class 'noia.vardec'
print(x, ...)
```

Arguments

`obj`	An object of class `"noia.linear"`, the output of `linearRegression` or of class `"noia.linear.gpmap"`, the output of `linearGPmapanalysis`.
`x`	An object of class `"noia.vardec"`, the output of `varianceDecomposition`.
`...`	No effect for the moment.

Details

The details of the variance decomposition are provided for all levels of interaction: Var(A) and Var(D) for marginal effects, Var(AA), Var(AD) and Var(DD) for 2nd order interactions, etc.

Value

`varianceDecomposition` returns a list of vectors. Each element of the list corresponds to an order of interactions, and the vectors detail the variance decomposition within each level. `print.noia.vardec` prints the previous list in a nice way, and computed the percentage of genetic variance explained by each variance component.

Author(s)

Arnaud Le Rouzic

References

Alvarez-Castro JM, Carlborg O. (2007). A unified model for functional and statistical epistasis and its application in quantitative trait loci analysis. Genetics 176(2):1151-1167.

Fisher RA. (1918). The correlation between relatives on the supposition of Mendelian inheritance. Thans. Roy. Soc. Edinburgh 52:339-433.

Le Rouzic A, Alvarez-Castro JM. (2008). Estimation of genetic effects and genotype-phenotype maps. Evolutionary Bioinformatics, 4.

Lynch M, Walsh B (1998) Genetics and Analysis of Quantitative Traits. Sunderland, MA; Sinauer Associates.

See Also

linearRegression

Examples

```
map <- c(0.25, -0.75, -0.75, -0.75, 2.25, 2.25, -0.75, 2.25, 2.25)
pop <- simulatePop(map, N=500, sigmaE=0.2, type="F2")

# Regression

linear <- linearRegression(phen=pop$phen, gen=cbind(pop$Loc1, pop$Loc2))

# Variance decomposition
varianceDecomposition(linear)
```

Index