Package 'netgwas'

April 12, 2019

Type Package

Title Network-Based Genome Wide Association Studies

Version 1.10

Author Pariya Behrouzi <https://orcid.org/0000-0001-6762-5433> and Ernst C. Wit

Maintainer Pariya Behrouzi <pariya.behrouzi@gmail.com>

Depends R (>= 3.1.0)

Imports Matrix, igraph, qtl, parallel, methods, glasso, MASS, RBGL, huge,tmvtnorm

Suggests testthat

Description A multi-core R package that contains a set of tools based on undirected graphical models for accomplishing three important and interrelated goals in genetics: (1) linkage map construction, (2) reconstructing intra- and inter-chromosomal conditional interactions (linkage disequilibrium) networks, and (3) exploring high-dimensional genotype-phenotype network and genotype-phenotype-environment interactions network. We use conditional independence relationships between variables.
The netgwas package can deal with biparental inbreeding and outbreeding species with any ploidy level, namely diploid (2 sets of chromosomes), triploid (3 sets of chromosomes), tetraploid (4 sets of chromosomes) and so on. We target on high-dimensional data where number of variables p is larger than number of sample sizes (p >> n). The computations is memory-optimized using the sparse matrix output. The package is implemented the recent developments in Behrouzi and Wit (2017) <doi:10.1111/rssc.12287> and Behrouzi and Wit (2017) <arXiv:1710.01063>.
NOTICE proper functionality of 'netgwas' requires that the 'RBGL' package is installed from 'bioconductor'; for installation instruction please refer to the 'RBGL' web page given below.

License GPL-3

Date 2019-04-12

NeedsCompilation no

Repository CRAN

Date/Publication 2019-04-12 14:02:43 UTC

R topics documented:

`netgwas-package` *Network Based Genome Wide Association Studies*

Description

The R package **netgwas** provides a set of tools based on undirected graphical models for accomplishing three important and interrelated goals in genetics: (1) linkage map construction, (2) reconstructing intra- and inter-chromosomal conditional interactions (linkage disequilibrium) networks, and (3) exploring high-dimensional genotype-phenotype network and genotype-phenotype- environment interactions network. The netgwas can deal with biparental species with any ploidy level. The package implemented the recent improvements both for construction of linkage maps in diploid and polyploid species in Behrouzi and Wit(2017b), and in reconstructing networks for non-Gaussian data, ordinal data, and mixed continuous and discrete data in Behrouzi and Wit (2017a). One application is to uncover epistatic interactions network, where the network captures the conditionally

dependent short- and long-range linkage disequilibrium structure of a genomes and reveals aberrant marker-marker associations. In addition, Behrouzi and Wit(2017c) implemented their proposed method to explore genotype-phenotype networks where nodes are either phenotypes or genotypes, and each phenotype is connected by an edge to a genotype or a group of genotypes if there is a direct association between them, given the rest of the variables. Different phenotypes may also interconnect. The conditionally dependent relationships between markers on a genome and phenotypes is determined through Gaussian copula graphical model. We remark that environmental variables can also be included along with genotype-phenotype input data to reconstruct networks between genotypes, phenotypes, and environment variables. Beside, the package contains functions for simulation and visualization, as well as three multivariate datasets taken from literature.

Author(s)

Pariya Behrouzi and Ernst C. Wit
Maintainers: Pariya Behrouzi <pariya.behrouzi@gmail.com>

References

1. Behrouzi, P., and Wit, E. C. (2019). Detecting epistatic selection with partially observed genotype data by using copula graphical models. Journal of the Royal Statistical Society: Series C (Applied Statistics), 68(1), 141-160.
2. Behrouzi, P., and Wit, E. C. (2018). De novo construction of polyploid linkage maps using discrete graphical models. Bioinformatics.
3. Behrouzi, P., and Wit, E. C. (2017c). netgwas: An R Package for Network-Based Genome-Wide Association Studies. arXiv preprint, arXiv:1710.01236.

Examples

```
## Not run:
# Notice: first install 'RBGL' from Bioconductor for the proper functionality of 'netgwas'
source("https://bioconductor.org/biocLite.R")
biocLite("RBGL")


install.packages("netgwas")
library(netgwas)

## End(Not run)
```

---

bp *genotype-phenotype data in mice*

Description

Data from an intercross between BALB/cJ and CBA/CaJ mouse strains

Usage

```
data(bp)
```

Format

The format is a matrix containing 93 SNP markers across the genome, and 4 phenotypes: blood pressure (bp), heart rate (hr), body weight (bw), and heart weight (heart-wt), as measured for 163 individuals.

Details

This genotype data can be used to reconstruct genotype-phenotype networks in mice (see below example) to identify genomic regions that regulate blood pressure, heart rate, and heart weight.

Source

Sugiyama, F., Churchill, G.A., Li, R., Libby, L.J., Carver, T., Yagami, K.I., John, S.W. and Paigen, B., 2002. QTL associated with blood pressure, heart rate, and heart weight in CBA/CaJ and BALB/cJ mice. Physiological genomics, 10(1), pp.5-12.

Examples

```
data(bp)
#Constructing genotype-phenotype networks in mice
out <- netphenogeno(bp)
sel <- selectnet(out)
plot(sel, vis= "interactive", vertex.color = c(rep("red",4), rep( "white", 93) ))
```

---

buildMap *linkage group detection and ordering markers for class "netgwasmap"*

Description

Implements different algorithms for detecting linkage groups and ordering markers in each linkage group.

Usage

```
buildMap( res, opt.index, min.m = NULL, use.comu = FALSE)
```

Arguments

res	An object with S3 class "netgwasmap"
opt.index	An index of a desired regularization parameter.
min.m	Expected minimum number of markers in a chromosome. Optional
use.comu	Using community detection algorithm to detect linkage groups. Default is FALSE.

Details

This function determines linkage groups and order markers within each linkage group for class "netgwasmap".

Value

An object with S3 class `"netgwasmap"` is returned:

`map`	Constructed linkage map associated with `opt.index`.
`opt.index`	The index of a desired 3-D map to construct linkage map.
`cross`	The specified cross type by user.
`allres`	A list containing results for different regularization parameter. Belongs to class "netgwas". To visualize a path of different 3D maps consider function `plot.netgwas`. Note that the input data is reordered based on the estimated linkage map and is saved as data in this argument.

Author(s)

Pariya Behrouzi and Ernst C.Wit
Maintainer: Pariya Behrouzi <pariya.behrouzi@gmail.com>

References

1. Behrouzi, P., and Wit, E. C. (2018). De novo construction of polyploid linkage maps using discrete graphical models. Bioinformatics.
2. Behrouzi, P., and Wit, E. C. (2017c). netgwas: An R Package for Network-Based Genome-Wide Association Studies. arXiv preprint, arXiv:1710.01236.

See Also

`netmap`

Examples

```
## Not run:
data(CviCol)
#Randomly change the order of markers across the genome
cvicol <- CviCol[ ,sample(1:ncol(CviCol), ncol(CviCol), replace=FALSE)]

#Constructing linkage map for Cvi x Col genotype data
out <- netmap(cvicol, cross= "inbred", ncores=1); out
plot(out)
map <- out$map; map

#Visualization of other networks
plot(out$allres)
#Constructing a linkage map for 5th network
bm <- buildMap(out, opt.index=5); bm
plot(bm, vis= "summary")
```

```
#or
plot(bm, vis= "interactive", label.vertex="all")

## End(Not run)
```

`cal.pos` *Estimate genetic map distances*

Description

Calculation of genetic map distances for an estimated markers order from either `net.map` or `buildMap` functions. This function is only for diploid populations. We note that the output of `net.map` and `buildMap` functions include estimated linkage groups and estimated markers order within each linkage group.

Usage

```
cal.pos (netgwasmap, pop.type= NULL , map.func = "haldane", chr )
```

Arguments

`netgwasmap`	A `netgwasmap` object. The output of `netmap` or `buildMap` functions.
`pop.type`	Character string specifying the population type of the genotype data. Accepted values are "DH" (doubled haploid), "BC" (backcross), "RILn" (non-advanced RIL population with n generations of selfing) and "ARIL" (advanced RIL) (see Details).
`map.func`	Character string defining the distance function used for calculation of genetic distances. Options are "kosambi", "haldane", and "morgan". Default is "haldane".
`chr`	A character string of linkage group names that require calculating of their genetic map distances.

Details

In **qtl** package, the genotype data for a backcross is coded as NA = missing, 1 = AA, 2 = AB. For an F2 intercross, the coding is NA = missing, 1 = AA, 2 = AB, 3 = BB, 4 = not BB (i.e. AA or AB), 5 = not AA (i.e. AB or BB).

If `pop.typ = "RILn"` the number of generations of selfing is limited to 20 to ensure sensible input. The constructed object is returned as a R/qtl `cross` object with the appropriate class structure. For `"RILn"` populations the constructed object is given the class "bcsft" by using the **qtl** package conversion function `convert2bcsft` with arguments `F.gen = n` and `BC.gen = 0`. For `"ARIL"` populations the constructed object is given the class `"riself"`.

This function uses the Viterbi algorithm implemented in `argmax.geno` of the **qtl** package to estimate genetic distances. Initial conservative estimates of the map distances are calculated from inverting recombination fractions outputted from `est.rf`. These are then passed to `argmax.geno` and imputation of missing allele scores is performed along with re-estimation of map distances. This is an adapted version of `quickEst` function from **ASMap** package.

Value

The netgwas constructed linkage map is returned as a R/qtl cross object. The object is a list with usual components "pheno" and "geno".

geno	The "geno" element contains data and map for separated linkage groups which have been constructed using net.map function.
pheno	Character string containing the genotype names.

Author(s)

Pariya Behrouzi
Maintainer: Pariya Behrouzi <pariya.behrouzi@gmail.com>

Examples

```
## Not run:
sim <- simRIL(d=25, n=200, g=5, cM=100, selfing= 2)
 # to use the same genotyping coding as qtl package (See details)
sim$data <- (sim$data) + 1

 #Estimate linkage groups and order markers within each LG
out <- netmap(sim$data, cross = "inbred")
map <- out$map; map

plot(out)

 #Calculate map positions and convert the map to cross object from qtl package
pos.map <- cal.pos(netgwasmap = out, pop.type= "RIL2", map.func = "haldane" )
plotMap(pos.map)

## End(Not run)
```

---

cross2netgwas — cross *object to* netgwas *data frame*

Description

Converts cross object from R/**qtl** package to netgwas dataframe

Usage

```
cross2netgwas (cross.obj)
```

Arguments

cross.obj	An object of class cross.

Value

An $(n \times p)$ matrix corresponds to a genotype data matrix (n is the sample size and p is the number of variables). This matrix can be as an input data for `netmap`, and `netsnp` functions.

Author(s)

Pariya Behrouzi
Maintainer: Pariya Behrouzi <pariya.behrouzi@gmail.com>

`cutoffs` *Cut-points*

Description

Calculates cut-points of ordinal variables with respect to the Gaussian copula.

Usage

```
cutoffs(y)
```

Arguments

`y`	An $(n \times p)$ matrix or a `data.frame` corresponding to the data matrix (n is the sample size and p is the number of variables). It also could be an object of class `"simgeno"`.

Details

The relationship between jth variable and jth latent variable is expressed through this set of cut-points.

Value

`cutoffs`	A p by $(k+1)$ matrix representing the cut-point values under the Gaussian copula, where k defines the number of categories in the dataset.

Author(s)

Pariya Behrouzi and Ernst C. Wit
Maintainer: Pariya Behrouzi <pariya.behrouzi@gmail.com>

References

1. Behrouzi, P., and Wit, E. C. (2019). Detecting epistatic selection with partially observed genotype data by using copula graphical models. Journal of the Royal Statistical Society: Series C (Applied Statistics), 68(1), 141-160.
2. Behrouzi, P., and Wit, E. C. (2018). De novo construction of polyploid linkage maps using discrete graphical models. Bioinformatics.
3. Behrouzi, P., and Wit, E. C. (2017c). netgwas: An R Package for Network-Based Genome-Wide Association Studies. arXiv preprint, arXiv:1710.01236.

See Also

`lower.upper`, `simgeno` and `netgwas-package`.

Examples

```
D <- simgeno(p = 100, n = 50, k = 3)
cutoffs(D$data)
```

`CviCol` *Arabidopsis thaliana genotype data*

Description

The genotype data of the Cvi-0 $\times$ Col-0 Recombinant Inbred Line (RIL) population.

Usage

```
data(CviCol)
```

Format

The format is a matrix containing 90 single-nucleotide polymorphism (SNP) markers for 367 individuals.

Details

The *Arabidopsis thaliana* genotype data is derived from a RIL cross between Columbia-0 (Col-0) and the Cape Verde Island (Cvi-0), where 367 individuals were genotyped for 90 genetic markers. This is a diploid population with three possible genotpe states (`k = 3`), where the genotypes coded as 0, 1, 2, where 0 and 2 represent the homozygous genotypes and 1 defines the heterozygous genotype.
This data set can be used to detect epistatic selection, short- and long- range linkage disequilibrium between 90 SNP markers.

Author(s)

Pariya Behrouzi and Ernst C. Wit
Maintainer: Pariya Behrouzi <pariya.behrouzi@gmail.com>

Source

Simon, M., et al. "QTL mapping in five new large RIL populations of Arabidopsis thaliana genotyped with consensus SNP markers." Genetics 178 (2008): 2253-2264. It is publicly available at http://publiclines.versailles.inra.fr/page/8

Examples

```
data(CviCol)
dim(CviCol)
head(CviCol, n=3)
```

`detect.err` *Identiying likely genotyping error*

Description

Calculates a LOD score for each genotype, measuring the evidence for genotyping errors. This uses `calc.errorlod` function from R/**qtl** package.

Usage

```
detect.err(netgwas.map,  err.prob= 0.01, cutoff= 4,
          pop.type= NULL, map.func= "haldane")
```

Arguments

`netgwas.map`	An object of class `netgwasmap` object (The output of `netmap` or `netmap` functions).
`err.prob`	Assumed genotyping error rate used in the calculation of the penetrance Pr(observed genotype | true genotype).
`cutoff`	Only those genotypes with error LOD scores above this cutoff will be listed.
`pop.type`	Character string specifying the population type of the genotype data. Accepted values are "DH" (doubled haploid), "BC" (backcross), "RILn" (non-advanced RIL population with n generations of selfing) and "ARIL" (advanced RIL) (see Details).
`map.func`	Character string defining the distance function used for calculation of genetic distances. Options are "kosambi", "haldane", and "morgan". Default is "haldane".

Value

A data.frame with 4 columns, whose rows correspond to the genotypes that are possibly in error. The four columns give the chromosome number, individual number, marker name, and error LOD score.

Examples

```
## Not run:
sim <- simRIL(d=25, n=200, g=5, cM=100, selfing= 2)
 # to use the same genotyping coding as R/qtl package (See details)
sim$data <- (sim$data) + 1

 #Estimate linkage groups and order markers within each LG
out <- netmap(sim$data, cross = "inbred")
map <- out$map; map
plot(out)

# A list of genotyoing error
detect.err(out, pop.type = "RIL2")

## End(Not run)
```

---

hdl *Mus Musculus HDL data in mice*

Description

HDL QTL data was obtained from a F2 inner-cross between inbred MRL/MpJ and SM/J strains of mice.

Usage

```
data(hdl)
```

Format

The format is a matrix consists of 280 observations for 15 variables: genotype data (genotype states at 5 SNP markers) and phenotype data (HDL levels and normalized expression values of 10 genes). Three possible genotype states MM (homozygous) are denoted by 1, H (heterozygous) by 2, and SS (homozygous) by 3 and phenotypes are of class numeric.

Details

The *Mus Musculus* HDL data were obtained from an F2 inner-cross between inbred MRL/MpJ and SM/J strains of mice.

Source

Leduc MS, Blair RH, Verdugo RA, Tsaih SW, Walsh K, Churchill GA, Paigen B.(2012). "Using bioinformatics and systems genetics to dissect HDL-cholesterol genetics in an MRL/MpJ x SM/J intercross." J Lipid Res., 6, 1163-75.

Examples

```
data(hdl)
#Constructing genotype-phenotype networks in mice
out <- netphenogeno(hdl)
sel <- selectnet(out)
plot(sel, vis= "CI", vertex.label= TRUE, xlab= "Genotype-Phenotype", ylab= "Genotype-Phenotype")
cl <- c(rep("white", 5), rep("red", 10))
plot(sel, vis="interactive", n.mem=c(5, 10), vertex.color= cl, w.btw= 10, w.within = 8  )
```

---

lower.upper *Calculates lower band and upper band*

Description

Calculates lower and upper bands for each data point, using a set of cut-points which is obtained from the Gaussian copula.

Usage

```
lower.upper(y)
```

Arguments

y	An $(n \times p)$ matrix or a `data.frame` corresponding to the data matrix (n is the sample size and p is the number of variables). It also could be an object of class `"episim"`.

Value

lower	A n by p matrix representing the lower band for each data point.
upper	A n by p matrix representing the upper band for each data point.

Author(s)

Pariya Behrouzi and Ernst C. Wit
Maintainer: Pariya Behrouzi <pariya.behrouzi@gmail.com>

References

Behrouzi, P., and Wit, E. C. (2019). Detecting epistatic selection with partially observed genotype data by using copula graphical models. Journal of the Royal Statistical Society: Series C (Applied Statistics), 68(1), 141-160.

See Also

cutoffs and netgwas-package.

Examples

```
D <- simgeno(p = 100, n = 50, k = 3)
lower.upper(D$data)
```

---

netgwas2cross *netgwasmap object to cross object*

Description

Convertes netgwasmap object from net.map or buildMap functions to cross object from R/**qtl** package.

Usage

```
netgwas2cross(netgwasmap, pop.type= NULL, map.func = "haldane")
```

Arguments

netgwasmap	A netgwasmap object. The output of netmap or buildMap functions.
pop.type	Character string specifying the population type of the genotype data. Accepted values are "DH" (doubled haploid), "BC" (backcross), "RILn" (non-advanced RIL population with n generations of selfing) and "ARIL" (advanced RIL).
map.func	Character string defining the distance function used for calculation of genetic distances. Options are "kosambi", "haldane", and "morgan". Default is "haldane".

Details

If pop.typ = "RILn" the number of generations of selfing is limited to 20 to ensure sensible input. The constructed object is returned as a R/qtl cross object with the appropriate class structure. For "RILn" populations the constructed object is given the class "bcsft" by using the **qtl** package conversion function convert2bcsft with arguments F.gen = n and BC.gen = 0. For "ARIL" populations the constructed object is given the class "riself".

In R/**qtl** package, the genotype data for a backcross is coded as NA = missing, 1 = AA, 2 = AB. For an F2 intercross, the coding is NA = missing, 1 = AA, 2 = AB, 3 = BB, 4 = not BB (i.e. AA or AB), 5 = not AA (i.e. AB or BB).

Value

The netgwas.map object is returned as a cross object form R/**qtl**. The object is a list with usual components "pheno" and "geno".

geno	The "geno" element contains data and map for separated linkage groups which have been constructed using net.map function.
pheno	Character string containing the genotype names.

Author(s)

Pariya Behrouzi
Maintainer: Pariya Behrouzi <pariya.behrouzi@gmail.com>

Examples

```
## Not run:
  sim <- simRIL(d=25, n=200, g=5, cM=100, selfing= 2)
  # to use the same genotyping coding as R/qtl package (See details)
  sim$data <- (sim$data) + 1

  #Estimate linkage groups and order markers within each LG
  out <- netmap(sim$data, cross = "inbred")
  map <- out$map; map

  plot(out)

  #Calculate map positions and convert the map to cross object from qtl package
  map <- netgwas2cross(netgwasmap = out, pop.type= "RIL2", map.func = "haldane" )
  plotMap(map)

## End(Not run)
```

---

netmap *Constructing linkage map for diploids and polyploids*

Description

This is one of the main functions of **netgwas** package. This function reconstructs linkage maps for biparental diploid and polyploid organisms using three methods.

Usage

```
netmap(data, method = "npn", cross= NULL, rho = NULL, n.rho = NULL,
       rho.ratio = NULL, min.m= NULL, use.comu= FALSE, ncores = "all",
  em.iter = 5, verbose = TRUE)
```

Arguments

data	An ($n \times p$) matrix or a data.frame corresponding to a genotype data matrix (n is the sample size and p is the number of variables). Input data can contain missing values.
method	Three available methods to construct linkage map: "gibbs", "approx", and "npn". Default is "npn"
rho	A decreasing sequence of non-negative numbers that control the sparsity level. Leaving the input as rho = NULL, the program automatically computes a sequence of rho based on n.rho and rho.ratio. Users can also supply a decreasing sequence values to override this.
n.rho	The number of regularization parameters. The default value is 6.
rho.ratio	Determines distance between the elements of rho sequence. A small value of rho.ratio results in a large distance between the elements of rho sequence. And a large value of rho.ratio results into a small distance between elements of rho. If keep it as NULL the program internally chooses a value.
cross	To be specified either "inbred" or "outbred".
min.m	Expected minimum number of markers in a chromosome. Optional
use.comu	Use community detection algorithm to detect linkage groups. Default is FALSE.
ncores	The number of cores to use for the calculations. Using ncores = "all" automatically detects number of available cores and runs the computations in parallel on (available cores - 1).
em.iter	The number of EM iterations. The default value is 5.
verbose	Providing a detail message for tracing output. The default value is TRUE.

Details

Constructing linkage maps for diploid and polyploid organisms. Diploid organisms contain two sets of chromosomes, one from each parent, whereas polyploids contain more than two sets of chromosomes. Inbreeding is mating between two parental lines where they have recent common biological ancestors. If they have no common ancestors up to roughly e.g. 4-6 generations, this is called outcrossing. In both cases the genomes of the derived progenies are random mosaics of the genome of the parents. However, in the case of inbreeding parental alleles are distinguishable in the genome of the progeny; in outcrossing this does not hold.

Value

An object with S3 class "netgwasmap" is returned:

map	Constructed linkage map.
opt.index	The index of selected graph using model selection.
cross	The pre-specified cross type.
allres	A list containing results for different regularization parameter. Belongs to class "netgwas". To visualize a path of different 3D maps consider function plot.netgwas. Note that the input data is reordered based on the estimated linkage map and is saved as data in this argument.

Author(s)

Pariya Behrouzi and Ernst C. Wit
Maintainers: Pariya Behrouzi <pariya.behrouzi@gmail.com>

References

1. Behrouzi, P., and Wit, E. C. (2018). De novo construction of polyploid linkage maps using discrete graphical models. Bioinformatics.
2. Behrouzi, Pariya, and Ernst C. Wit. "netgwas: An R Package for Network-Based Genome-Wide Association Studies." arXiv preprint arXiv:1710.01236 (2017).
3. Guo, Jian, Elizaveta Levina, George Michailidis, and Ji Zhu. "Graphical models for ordinal data." Journal of Computational and Graphical Statistics 24, no. 1 (2015): 183-204.
4. Liu, Han, Fang Han, Ming Yuan, John Lafferty, and Larry Wasserman. "High-dimensional semiparametric Gaussian copula graphical models." The Annals of Statistics 40, no. 4 (2012): 2293-2326.
5. Witten, Daniela M., Jerome H. Friedman, and Noah Simon. "New insights and faster computations for the graphical lasso." Journal of Computational and Graphical Statistics 20, no. 4 (2011): 892-900.

Examples

```
## Not run:
data(CviCol)
#Randomly change the order of markers across the genome
cvicol <- CviCol[ ,sample(1:ncol(CviCol), ncol(CviCol), replace=FALSE)]

#Constructing linkage map using gibbs method
out <- netmap(cvicol, cross= "inbred", ncores=1); out
#Estimated linkage map
map <- out$map; map
#Plot the associated network
plot(out)
#Visualizing the path networks
plot(out$allres)
#Build a linkage map for 5th networks
bm <- buildMap(out, opt.index=5); bm
####################

#Constructing linkage map using approx method
out2 <- netmap(cvicol, method="approx", cross= "inbred", ncores=1); out2
#Estimated linkage map
map2 <- out2$map; map2
#Plot the related network
plot(out2)
#Visualize other networks
plot(out2$allres)
#Build a linkage map for 5th network
bm2 <- buildMap(out2, opt.index=5); bm2

#Constructing linkage map using npn method
```

```
out3 <- netmap(cvicol, method="npn", cross= "inbred", ncores=1); out3
#Estimated linkage map
map3 <- out3$map; map3
#Plot the related network
plot(out3)

## End(Not run)
```

`netphenogeno` *Reconstructs conditional dependence network among genetic loci and phenotypes*

Description

This is one of the main functions of the **netgwas** package. This function reconstructs a conditional independence network between genotypes and phenotypes for diploids and polyploids. Three methods are available to reconstruct networks, namely (i) Gibbs sampling, (ii) approximation method, and (iii) nonparanormal approach within the Gaussian copula graphical model. The first two methods are able to deal with missing genotypes. The last one is computationally faster.

Usage

```
netphenogeno(data, method = "gibbs", rho = NULL, n.rho = NULL, rho.ratio = NULL,
ncores = 1, em.iter = 5, em.tol=.001, verbose = TRUE)
```

Arguments

`data`	An ($n \times p$) matrix or a `data.frame` corresponding to the data matrix (n is the sample size and p is the number of variables). The p columns include either a marker or trait(s) information. Input data can contain missing values.
`method`	Reconstructing both genotype-phenotype interactions network and genotype-phenotype-environment interactions network with three methods: "gibbs", "approx", and "npn". For a medium (~500) and a large number of variables we recommend to choose "gibbs" and "approx", respectively. Choosing "npn" for a very large number of variables (> 2000) is computationally efficient. The default method is "gibbs".
`rho`	A decreasing sequence of non-negative numbers that control the sparsity level. Leaving the input as `rho = NULL`, the program automatically computes a sequence of `rho` based on `n.rho` and `rho.ratio`. Users can also supply a decreasing sequence values to override this.
`n.rho`	The number of regularization parameters. The default value is `10`.
`rho.ratio`	Determines distance between the elements of `rho` sequence. A small value of `rho.ratio` results in a large distance between the elements of `rho` sequence. And a large value of `rho.ratio` results into a small distance between elements of `rho`. The default value is `0.3`.

ncores	The number of cores to use for the calculations. Using ncores = "all" automatically detects number of available cores and runs the computations in parallel on (available cores - 1).
em.iter	The number of EM iterations. The default value is 5.
em.tol	A criteria to stop the EM iterations. The default value is .001.
verbose	Providing a detail message for tracing output. The default value is TRUE.

Details

This function reconstructs both genotype-phenotype network and genotype-phenotype-environment interactions network. In genotype-phenotype networks nodes are either markers or phenotypes; each phenotype is connected by an edge to a marker if there is a direct association between them given the rest of the variables. Different phenotypes may also interconnect. In addition to markers and phenotypes information, the input data can include environmental variables. Then, the interactions network shows the conditional dependence relationships between markers, phenotypes and environmental factors.

Value

An object with S3 class "netgwas" is returned:

Theta	A list of estimated p by p precision matrices that show the conditional independence relationships patterns among measured items.
path	A list of estimated p by p adjacency matrices. This is the graph path corresponding to Theta.
ES	A list of estimated p by p conditional expectation corresponding to rho.
Z	A list of n by p transformed data based on Gaussian copula.
rho	A n.rho dimensional vector containing the penalty terms.
loglik	A n.rho dimensional vector containing the maximized log-likelihood values along the graph path.
data	The n by p input data matrix. The n by p transformed data in case of using "npn".

Note

This function estimates a graph path . To select an optimal graph please refer to selectnet.

Author(s)

Pariya Behrouzi and Ernst C. Wit
Maintainers: Pariya Behrouzi <pariya.behrouzi@gmail.com>

References

1. Behrouzi, P., and Wit, E. C. (2019). Detecting epistatic selection with partially observed genotype data by using copula graphical models. Journal of the Royal Statistical Society: Series C (Applied Statistics), 68(1), 141-160.
2. Behrouzi, P., and Wit, E. C. (2017c). netgwas: An R Package for Network-Based Genome-Wide Association Studies. arXiv preprint, arXiv:1710.01236.
3. D. Witten and J. Friedman. New insights and faster computations for the graphical lasso. *Journal of Computational and Graphical Statistics*, to appear, 2011.
4. Guo, Jian, et al. "Graphical models for ordinal data." Journal of Computational and Graphical Statistics 24.1 (2015): 183-204.

See Also

selectnet

Examples

```
data(thaliana)
head(thaliana, n=3)
#Construct a path for genotype-phenotype interactions network in thaliana data
res <-  netphenogeno(data = thaliana); res
plot(res)
#Select an optimal network
sel <- selectnet(res)
#Plot selected network and the conditional correlation (CI) relationships
plot(sel, vis="CI")
plot(sel, vis="CI", n.mem = c(8, 56, 31, 33, 31, 30), w.btw =50, w.within= 1)

#Visualize interactive plot for the selected network
#Color "red" for 8 phenotypes, and different colors for each chromosome.
cl <- c(rep("red", 8), rep("white",56), rep("tan1",31),
      rep("gray",33), rep("lightblue2",31), rep("salmon2",30))

#The IDs of phenotypes and SNPs to be shown in the network
    id <- c("DTF_LD","CLN_LD","RLN_LD","TLN_LD","DTF_SD","CLN_SD","RLN_SD",
        "TLN_SD","snp15","snp16","snp17","snp49","snp50","snp60","snp75",
        "snp76","snp81","snp83","snp84","snp86","snp82", "snp113","snp150",
        "snp155","snp159","snp156","snp161","snp158","snp160","snp162","snp181")

plot(sel, vis="interactive", n.mem = c(8, 56, 31, 33, 31, 30), vertex.color= cl,
    label.vertex= "some", sel.nod.label= id, edge.color= "gray", w.btw= 50,
    w.within= 1)

#Partial correlations between genotypes and phenotypes in the thaliana dataset.
library(Matrix)
image(sel$par.cor, xlab="geno-pheno", ylab="geno-pheno", sub="")
```

netsnp *Reconstructs intra- and inter- chromosomal conditional interactions among genetic loci*

Description

This is one of the main functions of the **netgwas** package. This function can be used to reconstruct the intra- and inter-chromosomal interactions among genetic loci in diploids and polyploids. The input data can be belong to any biparental genotype data which contains at least two genotype states. Two methods are available to reconstruct the network, namely (1) approximation method, and (2) gibbs sampling within the Gaussian copula graphical model. Both methods are able to deal with missing genotypes.

Usage

```
netsnp(data, method = "gibbs", rho = NULL, n.rho = NULL, rho.ratio = NULL,
ncores = 1, em.iter = 5, em.tol = .001, verbose = TRUE)
```

Arguments

data	An $(n \times p)$ matrix or a data.frame corresponding to a genotype data matrix (n is the sample size and p is the number of variables). It also could be an object of class "simgeno". Input data can contain missing values.
method	Reconstructs intra- and inter- chromosomal conditional interactions (epistatic selection) network with three methods: "gibbs", "approx", and "npn". For a medium (~500) and a large number of variables we would recommend to choose "gibbs" and "approx", respectively. For a very large number of variables (> 2000) choose "npn". The default method is "gibbs".
rho	A decreasing sequence of non-negative numbers that control the sparsity level. Leaving the input as rho = NULL, the program automatically computes a sequence of rho based on n.rho and rho.ratio. Users can also supply a decreasing sequence values to override this.
n.rho	The number of regularization parameters. The default value is 10.
rho.ratio	Determines the distance between the elements of rho sequence. A small value of rho.ratio results in a large distance between the elements of rho sequence. And a large value of rho.ratio results into a small distance between elements of rho. If keep it as NULL the program internally chooses a value.
ncores	The number of cores to use for the calculations. Using ncores = "all" automatically detects number of available cores and runs the computations in parallel on (available cores - 1).
em.iter	The number of EM iterations. The default value is 5.
em.tol	A criteria to stop the EM iterations. The default value is .001.
verbose	Providing a detail message for tracing output. The default value is TRUE.

Details

Viability is a phenotype that can be considered. This function detects the conditional dependent short- and long-range linkage disequilibrium structure of genomes and thus reveals aberrant marker-marker associations that are due to epistatic selection. This function can be used to estimate conditional independence relationships between partially observed data that not follow Gaussianity assumption (e.g. continuous non-Gaussian, discrete, or mixed dataset).

Value

An object with S3 class `"netgwas"` is returned:

`Theta`	A list of estimated p by p precision matrices that show the conditional independence relationships patterns among genetic loci.
`path`	A list of estimated p by p adjacency matrices. This is the graph path corresponding to `Theta`.
`ES`	A list of estimated p by p conditional expectation corresponding to `rho`.
`Z`	A list of n by p transformed data based on Gaussian copula.
`rho`	A `n.rho` dimensional vector containing the penalty terms.
`loglik`	A `n.rho` dimensional vector containing the maximized log-likelihood values along the graph path.
`data`	The n by p input data matrix.

Note

This function estimates a graph path . To select an optimal graph please refer to `selectnet`.

Author(s)

Pariya Behrouzi and Ernst C. Wit
Maintainers: Pariya Behrouzi <pariya.behrouzi@gmail.com>

References

1. Behrouzi, P., and Wit, E. C. (2019). Detecting epistatic selection with partially observed genotype data by using copula graphical models. Journal of the Royal Statistical Society: Series C (Applied Statistics), 68(1), 141-160.
2. Behrouzi, P., and Wit, E. C. (2017c). netgwas: An R Package for Network-Based Genome-Wide Association Studies. arXiv preprint, arXiv:1710.01236. 3. D. Witten and J. Friedman. New insights and faster computations for the graphical lasso. *Journal of Computational and Graphical Statistics*, to appear, 2011.
4. Guo, Jian, et al. "Graphical models for ordinal data." Journal of Computational and Graphical Statistics 24.1 (2015): 183-204.

See Also

`selectnet`

Examples

```
data(CviCol)
out <- netsnp(CviCol); out
plot(out)

#select optimal graph
epi <- selectnet(out)
plot(epi, vis="CI", xlab="markers", ylab="markers",
    n.mem = c(24,14,17,16,19), vertex.size=4)

#Visualize interactive plot of the selected network
#Different colors for each chromosome
cl <- c(rep("red", 24), rep("white",14), rep("tan1",17),
      rep("gray",16), rep("lightblue2",19))
plot(epi, vis="interactive", vertex.color= cl)

#Partial correlations between markers on genome
image(as.matrix(epi$par.cor), xlab="markers", ylab="markers", sub="")
```

---

plot.netgwas *plot for S3 class "netgwas"*

Description

Plot the graph path which is the output of two functions netsnp, netphenogeno.

Usage

```
## S3 method for class 'netgwas'
plot( x, n.markers=NULL , ... )
```

Arguments

x	An object from "netgwas" class.
n.markers	A vector containing number of variables/markers in each group/chromosome. For example, the CviCol dataset that is provided in the package contains 5 chromosomes/ groups which the total number of markers is $p = 90$, where the first 24 markers belong into chromosome 1, the next 14 markers into chromosome 2, ..., and chromosome 5 contains 19 markers. Thus, n.mrkr = c(24,14,17,16,19). If n.mrkr = NULL, in the graph visualization all markers are represented same colour.
...	System reserved (No specific usage)

Author(s)

Pariya Behrouzi and Ernst C. Wit
Maintainer: Pariya Behrouzi <pariya.behrouzi@gmail.com>

References

Behrouzi, P., and Wit, E. C. (2017c). netgwas: An R Package for Network-Based Genome-Wide Association Studies. arXiv preprint, arXiv:1710.01236.

See Also

netmap, netsnp, netphenogeno.

---

plot.netgwasmap *plot for S3 class "netgwasmap"*

Description

Plot the graph associated with constructed linkage map via function netmap.

Usage

```
## S3 method for class 'netgwasmap'
plot(x, vis= NULL, layout= NULL, vertex.size= NULL, label.vertex =
"none", label.size= NULL, vertex.color= NULL, edge.color = "gray29",
sel.ID = NULL, ... )
```

Arguments

x	An object from "netgwasmap" class.
vis	Visualizing in four options: (i) "summary" plots the related network, conditional dependence relationships between markers before and after ordering markers; (ii) "interactive" plots the associated network, where it opens a new windows with interactive graph drawing facility; (iii) "unordered markers" plots the conditional dependence relationships between markers before ordering markers; (iv) "ordered markers" plots conditional dependence relationships between markers after ordering markers. Default is "summary".
layout	The vertex placement algorithm which is according to **igraph** package. The default layout is Fruchterman-Reingold layout. Other possible layouts are, for example, layout_with_kk, circle, and Reingold-Tilford graph in **igraph** package.
vertex.size	Optional integer to adjust vertex size in graph G. Default is 5.

label.vertex	Assign names to the vertices. There are three options: "none", "some", "all". (i) Specifying "none" omits vertex labels in the graph, (ii) using label.vertex = "some" you need to provide a vector of vertex IDs or a single vertex ID to the sel.ID argument, which you would like to be shown in the graph. label.vertex = "some" is only applicable for vis = "interactive", (iii) Specifying "all" includes all vertex labels in the graph. Default is "none".
label.size	Optional integer to adjust the size of node's label in graph G. Applicable when vertex.label is TRUE. Default is 0.8.
vertex.color	Optional integer vectors giving colors to the vertices.
edge.color	Optional integer vectors giving colors to edges.
sel.ID	ONLY applicable when vis= "interactive". A vector of vertex IDs or a single vertex ID, which you would like to be shown in the graph. ONLY applicable when label.vertex="some".
...	ONLY applicable when vis= "CI". System reserved (No specific usage)

Author(s)

Pariya Behrouzi and Ernst C. Wit
Maintainer: Pariya Behrouzi <pariya.behrouzi@gmail.com>

References

1. Behrouzi, P., and Wit, E. C. (2018). De novo construction of polyploid linkage maps using discrete graphical models. Bioinformatics.
2. Behrouzi, P., and Wit, E. C. (2017c). netgwas: An R Package for Network-Based Genome-Wide Association Studies. arXiv preprint, arXiv:1710.01236.

See Also

netmap, buildMap.

---

plot.select *Plot function for S3 class "select"*

Description

Plot the optimal graph by model selection

Usage

```
## S3 method for class 'select'
plot(x, vis= NULL, xlab= NULL, ylab= NULL, n.mem= NULL, vertex.label= FALSE
, ..., layout= NULL, label.vertex= "all", vertex.size= NULL, vertex.color= NULL,
edge.color= "gray29", sel.nod.label= NULL, label.size = NULL, w.btw= 800,
 w.within = 10, sign.edg= TRUE, edge.width= NULL, edge.label= NULL,
max.degree= NULL, layout.tree= NULL, root.node= NULL, degree.node= NULL,
curve= FALSE, pos.legend= "bottomleft", cex.legend= 0.8, iterl = NULL, temp = NULL,
tk.width = NULL, tk.height= NULL)
```

Arguments

x	An object with S3 class "select"
vis	Visualizing the results as a graph (network) or as a matrix. There are 4 options to visulize the selected graph: (i) "CI": plotting conditional independence (CI) relationships between variables, (ii) "interactive": plotting the conditional independence network, where opens a new windows with interactive graph drawing facility, and (iii) "parcor.network": plots the estimated graph based on partial correlation values. (iv) "parcor.interactive": plots the estimated graph based on partial correlation matrix with an interactive graph drawing facility. Default is "CI". Also, there are 3 options to visulaze the selected graph as a matrix: (i) vis= "image.parcorMatrix" plots the image of partial correlation matrix, (ii) vis = "image.adj" draws the adjacency matrix (only presence and absence of links), (iii) vis = "image.precision" plots the selected precision matrix.
xlab	ONLY applicable when vis = "CI", "image.parcorMatrix", "image.adj", or "image.precision".
ylab	ONLY applicable when vis = "CI", "image.parcorMatrix", "image.adj", or "image.precision".
n.mem	A vector of memberships. For example, the CviCol dataset, which is provided in the package, contain 5 chromosomes which the total number of markers is $p = 90$, where the first 24 markers belong into chromosome 1, the next 14 markers into chromosome 2, ..., and chromosome 5 contains 19 markers. Thus, n.mem = c(24,14,17,16,19). If n.mem = NULL and vis = "CI" all vertices are coloured the same.
vertex.label	ONLY applicable when vis= "CI". Assign names to the vertices. Default is FALSE.
...	ONLY applicable when vis= "CI". System reserved (No specific usage)
layout	ONLY applicable when vis= "interactive" or "parcor.network". The layout specification. Some graph layouts examples: layout_with_fr, layout_in_circle, layout_as_tree, and layout.fruchterman.reingold. The default layout is layout_with_fr.
label.vertex	ONLY applicable when vis= "interactive". Assign names to the vertices. There are three options: "none", "some", "all". Specify "none" to omit vertex labels in the graph; using label.vertex = "some" you must provide a vector of vertex IDs or a single vertex ID to the sel.label argument, which you would like to be shown in the graph. Specify "all" to include all vertex labels in the graph. Default is "all".
vertex.size	Optional. The size of vertices in the graph visualization. The default value is 7.
vertex.color	ONLY applicable when vis= "interactive" or "parcor.network". Optional vector (or a color name) giving the colors of the vertices. The default is "red"
edge.color	ONLY applicable when vis= "interactive". Optional. The default is "gray".
sel.nod.label	ONLY applicable when vis= "interactive" or "parcor.network". A vector of vertex IDs or a single vertex ID, which you would like to be shown in the graph. ONLY applicable when label.vertex="some".
label.size	ONLY applicable for vis= "interactive" or vis= "parcor.network". The font size of the vertex labels.

w.btw	Distance between nodes from different memberships of n.mem in layout.
w.within	Distance of nodes within one membership of n.mem in layout.
sign.edg	Optional. ONLY applicable when vis= "parcor.network". If TRUE then edges are colored as red and blue, where red stands for positive and blue negative partial correlation values. If FASLE all edeges are colored as gray. Default is TRUE.
edge.width	Optional. ONLY applicable when vis= "parcor.network". Based on the strength of partial correlation values, edges will shown with different line type. Default is FALSE.
edge.label	Optional. ONLY applicable when vis= "parcor.network". If TRUE then the partial correlation values will be shown on top of each edge. Default is FALSE.
max.degree	Optional. ONLY applicable when vis= "parcor.network". A number showing degree of a node. This can be used to print those vertex labels that the correspondence vertex have at least e.g. 1 degree.
layout.tree	Optional. ONLY applicable when vis= "parcor.network". If TRUE then it uses layout_as_tree from igraph package. Default is FALSE.
root.node	Optional. ONLY applicable when vis= "parcor.network". The index of the root vertex or root vertices. If this is a non-empty vector then the supplied vertex ids are used as the roots of the trees . If it is an empty vector, then the root vertices are automatically calculated based on topological sorting, performed with the opposite mode than the mode argument. After the vertices have been sorted, one is selected from each component.
degree.node	Optional. ONLY applicable when vis= "parcor.network". It is related to the vertex label degree. It controls the position of the labels with respect to the vertices. Value are for example -pi/2, 0, pi/2, pi sets above, to the right, below, to the left of a node, respectively.
curve	Optional. ONLY applicable when vis= "parcor.network". Edge curvature, range between 0 and 1 (FALSE sets it to 0, TRUE to 0.5). Default is FALSE.
pos.legend	Applicable when vis= "parcor.network" or vis= "CI". The x and y co-ordinates to be used to position the legend. They can be specified by keywords like "topright", "topleft", and etc. Default is "bottomleft".
cex.legend	Applicable when vis= "parcor.network" or vis= "CI".
iterl	Optional. ONLY applicable when vis= "parcor.interactive". integer scalar, the number of iterations to perform for layout_with_fr layout.
temp	Optional. ONLY applicable when vis= "parcor.interactive". Real scalar, the start temperature for layout_with_fr layout.
tk.width	Optional. The size of the drawing area of interactive plot.
tk.height	Optional. The size of the drawing area of interactive plot.

Value

An object with S3 class "select" is returned:

network	Plot of a selected graph, when vis= "CI".
adjacency	Conditional independence (CI) relationships between variables, when vis= "CI"
network	Interactive plot of a selected graph with .eps format, when vis= "interactive"

Author(s)

Pariya Behrouzi
Maintainer: Pariya Behrouzi <pariya.behrouzi@gmail.com>

References

Behrouzi, P., and Wit, E. C. (2017c). netgwas: An R Package for Network-Based Genome-Wide Association Studies. arXiv preprint, arXiv:1710.01236.

See Also

select

Examples

```
#simulate data
data(CviCol)
out <- netsnp(CviCol)
sel <- selectnet(out)

cl <- c(rep("palegoldenrod", 24), rep("white",14), rep("tan1",17),
            rep("gray",16), rep("lightblue2",19))
    plot(sel, vis= "parcor.network", sign.edg = TRUE, layout = NULL, vertex.color = cl)
```

---

plot.simgeno *Plot function for* S3 *class* "simgeno"

Description

Visualizes the pattern of the true graph, the adjacency matrix, precison matrix and the covariance matrix of the simulated data.

Usage

```
## S3 method for class 'simgeno'
plot(x, layout = layout.fruchterman.reingold, ...)
```

Arguments

x	An object of S3 class "simgeno", from function simgeno.
layout	The default is "layout.fruchterman.reingold".
...	System reserved (No specific usage)

Author(s)

Pariya Behrouzi and Ernst C. Wit
Maintainer: Pariya Behrouzi <pariya.behrouzi@gmail.com>

References

Behrouzi, P., and Wit, E. C. (2017c). netgwas: An R Package for Network-Based Genome-Wide Association Studies. arXiv preprint, arXiv:1710.01236.

See Also

simgeno

Examples

```
## Not run:
# Generating discrete ordinal data with "genome-like" graph structure
data.sim <- simgeno(alpha = 0.01, beta = 0.02)
plot( data.sim )

## End(Not run)
```

---

print.netgwas *Print function for S3 class "netgwas"*

Description

Print a summary of results from function netsnp, netphenogeno.

Usage

```
## S3 method for class 'netgwas'
print(x, ...)
```

Arguments

x	An object with S3 class "netgwas"
...	System reserved (No specific usage)

Author(s)

Pariya Behrouzi and Ernst C. Wit
Maintainer: Pariya Behrouzi <pariya.behrouzi@gmail.com>

References

Behrouzi, P., and Wit, E. C. (2017c). netgwas: An R Package for Network-Based Genome-Wide Association Studies. arXiv preprint, arXiv:1710.01236.

See Also

netmap, netsnp, netphenogeno

`print.netgwasmap` *Print function for S3 class "netgwasmap"*

Description

Print a summary of results from function netmap.

Usage

```
## S3 method for class 'netgwasmap'
print(x, ...)
```

Arguments

x	An object with S3 class "netgwasmap"
...	System reserved (No specific usage)

Author(s)

Pariya Behrouzi and Ernst C. Wit
Maintainer: Pariya Behrouzi <pariya.behrouzi@gmail.com>

References

Behrouzi, P., and Wit, E. C. (2017c). netgwas: An R Package for Network-Based Genome-Wide Association Studies. arXiv preprint, arXiv:1710.01236.

See Also

netmap

print.select *Print function for S3 class "select"*

Description

Print function for selectnet.

Usage

```
## S3 method for class 'select'
print(x, ...)
```

Arguments

x	An object with S3 class "select"
...	System reserved (No specific usage)

Author(s)

Pariya Behrouzi and Ernst C. Wit
Maintainer: Pariya Behrouzi <pariya.behrouzi@gmail.com>

References

Behrouzi, P., and Wit, E. C. (2017c). netgwas: An R Package for Network-Based Genome-Wide Association Studies. arXiv preprint, arXiv:1710.01236.

See Also

selectnet

print.simgeno *Print function for S3 class "simgeno"*

Description

Print a summary of simulated data from function simgeno.

Usage

```
## S3 method for class 'simgeno'
print(x, ...)
```

Arguments

x	An object with S3 class "simgeno"
...	System reserved (No specific usage)

Author(s)

Pariya Behrouzi and Ernst C. Wit
Maintainer: Pariya Behrouzi <pariya.behrouzi@gmail.com>

References

Behrouzi, P., and Wit, E. C. (2017c). netgwas: An R Package for Network-Based Genome-Wide Association Studies. arXiv preprint, arXiv:1710.01236.

See Also

simgeno

R.approx *The expectation of covariance using approximation method*

Description

This function implements the approximation method within the Gaussian copula graphical model to estimate the conditional expectation for the data that not follow Gaussianity assumption (e.g. ordinal, discrete, continuous non-Gaussian, or mixed dataset).

Usage

```
R.approx(y, Z = NULL, Sigma=NULL, rho = NULL, ncores = NULL )
```

Arguments

y	An $(n \times p)$ matrix or a data.frame corresponding to the data matrix (n is the sample size and p is the number of variables). It also could be an object of class "simgeno".
Z	A $(n \times p)$ matrix which is a transformation of the data via the Gaussian copula. If Z = NULL internally calculates an initial value for Z.
Sigma	The covariance matrix of the latent variable given the data. If Sigma = NULL the Sigma matrix is calculated internally with a desired penalty term, rho.
rho	A (non-negative) regularization parameter to calculate Sigma. rho=0 means no regularization.
ncores	If ncores = NULL, the algorithm internally detects number of available cores and run the calculations in parallel on (available cores - 1). Typical usage is to fix ncores = 1 when p is small ($p < 500$), and ncores = NULL when p is large.

Value

ES	Expectation of covariance matrix(diagonal scaled to 1) of the Gaussian copula graphical model.
Z	New transformation of the data based on given or default Sigma.

Author(s)

Pariya Behrouzi and Ernst C. Wit
Maintainer: Pariya Behrouzi <pariya.behrouzi@gmail.com>

References

1. Behrouzi, P., and Wit, E. C. (2017c). netgwas: An R Package for Network-Based Genome-Wide Association Studies. arXiv preprint, arXiv:1710.01236.
2. Behrouzi, P., and Wit, E. C. (2019). Detecting epistatic selection with partially observed genotype data by using copula graphical models. Journal of the Royal Statistical Society: Series C (Applied Statistics), 68(1), 141-160.

Examples

```
## Not run:
D <- simgeno(p = 90, n = 50, k = 3)
R.approx(D$data)

## End(Not run)
```

R.gibbs *The expectation of covariance matrix using Gibbs sampling*

Description

This function implements the Gibbs sampling method within Gaussian copula graphical model to estimate the conditional expectation for the data that not follow Gaussianity assumption (e.g. ordinal, discrete, continuous non-Gaussian, or mixed dataset).

Usage

```
R.gibbs(y, theta, gibbs.iter = 1000, mc.iter = 500,
                  ncores = NULL, verbose = TRUE)
```

Arguments

y	An $(n \times p)$ matrix or a data.frame corresponding to the data matrix (n is the sample size and p is the number of variables). It also could be an object of class "simgeno".
theta	A $p \times p$ precision matrix. Default is a diagonal matrix.

gibbs.iter	The number of burn-in for the Gibbs sampling. The default value is 1000.
mc.iter	The number of Monte Carlo samples to calculate the conditional expectation. The default value is 500.
ncores	If ncores = NULL, the algorithm internally detects number of available cores and run the calculations in parallel on (available cores - 1). Typical usage is to fix ncores = 1 when p is small $(p < 500)$, and ncores = NULL when p is very large.
verbose	If verbose = FALSE, printing information is disabled. The default value is TRUE.

Details

This function calculates $\bar{R}$ using Gibbs sampling method within the E-step of EM algorithm, where

$$\bar{R} = n^{-1} \sum_{i=1}^{n} E(Z^{(i)} Z^{(i)t} | y^{(i)}, \hat{\Theta}^{(m)})$$

which n is the number of sample size and Z is the latent variable which is obtained from Gaussian copula graphical model.

Value

ES	Expectation of covariance matrix (diagonal scaled to 1) of the Gaussian copula graphical model

Author(s)

Pariya Behrouzi, Danny Arends and Ernst C. Wit
Maintainers: Pariya Behrouzi <pariya.behrouzi@gmail.com>

References

1. Behrouzi, P., and Wit, E. C. (2017c). netgwas: An R Package for Network-Based Genome-Wide Association Studies. arXiv preprint, arXiv:1710.01236.
2. Behrouzi, P., and Wit, E. C. (2019). Detecting epistatic selection with partially observed genotype data by using copula graphical models. Journal of the Royal Statistical Society: Series C (Applied Statistics), 68(1), 141-160.

Examples

```
D <- simgeno(p = 100, n = 50, k = 3)
R.gibbs(D$data, ncores=1)
```

`selectnet` *Model selection*

Description

Estimate the optimal regularization parameter at EM convergence based on different information criteria .

Usage

```
selectnet(netgwas.obj, opt.index= NULL, criteria= NULL, ebic.gamma=0.5,
   ncores= NULL, verbose= TRUE)
```

Arguments

`netgwas.obj`	An object with S3 class "netgwas"
`opt.index`	The program internally determines an optimal graph using `opt.index= NULL`. Otherwise, to manually choose an optimal graph from the graph path.
`criteria`	Model selection criteria. "ebic" and "aic" are available. BIC model selection can be calculated by fixing `ebic.gamma = 0`. Applicable only if `opt.index= NULL`.
`ebic.gamma`	The tuning parameter for ebic. The`ebic.gamma = 0` results in bic model selection. The default value is 0.5. Applicable only `opt.index= NULL`.
`ncores`	The number of cores to use for the calculations. Using `ncores = "all"` automatically detects number of available cores and runs the computations in parallel.
`verbose`	If `verbose = FALSE`, printing information is disabled. The default value is `TRUE`. Applicable only `opt.index= NULL`.

Details

This function computes extended Bayesian information criteria (ebic), Bayesian information criteria, Akaike information criterion (aic) at EM convergence based on observed or joint log-likelihood. The observed log-likelihood can be obtained through

$$\ell_Y(\widehat{\Theta}_\lambda) = Q(\widehat{\Theta}_\lambda|\widehat{\Theta}^{(m)}) - H(\widehat{\Theta}_\lambda|\widehat{\Theta}^{(m)}),$$

Where Q can be calculated from `netmap`, `netsnp`, `netphenogeno` function and H function is

$$H(\widehat{\Theta}_\lambda|\widehat{\Theta}^{(m)}_\lambda) = E_z[\ell_{Z|Y}(\widehat{\Theta}_\lambda)|Y;\widehat{\Theta}_\lambda] = E_z[\log f(z)|Y;\widehat{\Theta}_\lambda] - \log p(y).$$

The "ebic" and "aic" model selection criteria can be obtained as follow

$$ebic(\lambda) = -2\ell(\widehat{\Theta}_\lambda) + (\log n + 4\gamma \log p)df(\lambda)$$

$$aic(\lambda) = -2\ell(\widehat{\Theta}_\lambda) + 2df(\lambda)$$

where df refers to the number of non-zeros offdiagonal elements of $\hat{\Theta}_\lambda$, and $\gamma \in [0, 1]$. Typical value for for `ebic.gamma` is 1/2, but it can also be tuned by experience. Fixing `ebic.gamma = 0` results in bic model selection.

Value

An obj with S3 class "selectnet" is returned:

`opt.adj`	The optimal graph selected from the graph path
`opt.theta`	The optimal precision matrix from the graph path
`opt.sigma`	The optimal covariance matrix from the graph path
`ebic.scores`	Extended BIC scores for regularization parameter selection at the EM convergence. Applicable if `opt.index = NULL`.
`opt.index`	The index of optimal regularization parameter.
`opt.rho`	The selected regularization parameter.
`par.cor`	A partial correlation matrix.

and anything else that is included in the input `netgwas.obj`.

Author(s)

Pariya Behrouzi and Ernst C.Wit
Maintainer: Pariya Behrouzi <pariya.behrouzi@gmail.com>

References

1. BBehrouzi, P., and Wit, E. C. (2019). Detecting epistatic selection with partially observed genotype data by using copula graphical models. Journal of the Royal Statistical Society: Series C (Applied Statistics), 68(1), 141-160.
2. Behrouzi, P., and Wit, E. C. (2017c). netgwas: An R Package for Network-Based Genome-Wide Association Studies. arXiv preprint, arXiv:1710.01236.
3. Ibrahim, Joseph G., Hongtu Zhu, and Niansheng Tang. (2012). Model selection criteria for missing-data problems using the EM algorithm. Journal of the American Statistical Association. 4. D. Witten and J. Friedman. (2011). New insights and faster computations for the graphical lasso. *Journal of Computational and Graphical Statistics*, to appear.
5. J. Friedman, T. Hastie and R. Tibshirani. (2007). Sparse inverse covariance estimation with the lasso, *Biostatistics*.
6. Foygel, R. and M. Drton. (2010). Extended bayesian information criteria for Gaussian graphical models. In Advances in Neural Information Processing Systems, pp. 604-612.

See Also

`netmap`, `netsnp`, `netphenogeno`

Examples

```
#simulate data
D <- simgeno(p=50, n=100, k= 3, adjacent = 3, alpha = 0.06 , beta = 0.06)
plot(D)
```

```
#explore intra- and inter-chromosomal interactions
out  <-  netsnp(D$data, n.rho= 5, ncores= 1)
plot(out)

#different graph selection methods
sel.ebic1 <- selectnet(out, criteria = "ebic")
plot(sel.ebic1)

sel.aic <- selectnet(out, criteria = "aic")
plot(sel.aic)

sel.bic <- selectnet(out, criteria = "ebic", ebic.gamma = 0)
plot(sel.bic)
```

---

simgeno *Generate genotype data based on Gaussian copula*

Description

Generating discrete ordinal data based on underlying "genome-like" graph structure. The procedure of simulating data relies on a continues variable, which can be simulated from either multivariate normal distribution, or multivariate t-distribution with d degrees of freedom.

Usage

```
simgeno( p = 90, n = 200, k = NULL, g = NULL, adjacent = NULL, alpha =
              NULL , beta = NULL, con.dist = "Mnorm", d = NULL, vis = TRUE)
```

Arguments

p	The number of variables. The default value is 90.
n	The number of sample size (observations). The default value is 200.
k	The number of states (categories). The default value is 3.
g	The number of groups (chromosomes) in the graph. The default value is about $p/20$ if $p >= 40$ and 2 if $p < 40$.
adjacent	The number of adjacent variable(s) to be linked to a variable. For example, if `adjacent = 1` indicates a variable is linked via an edge with its adjacent variable on the left hand side, and its adjacent variable on the right hand side. The `adjacent = 2` defines a variable is linked via an edge with its 2 adjacent variables on its left hand side, and 2 adjacent variables on its right hand side. The default value is 1.
alpha	A probability that a pair of non-adjacent variables in the same group is given an edge. The default value is 0.01.
beta	A probability that variables in different groups are linked with an edge. The default value is 0.02.

con.dist	The distribution of underlying continuous variable. If con.dist = "Mnorm", a multivariate normal distribution with mean 0 is applied. If con.dist = "Mt", the t-distribution with a degrees of freedom is applied. The default distribution is con.dist = "Mnorm".
d	The degrees of freedom of the continuous variable, only applicable when code-con.dist = "Mt". The default value is 3.
vis	Visualize the graph pattern and the adjacency matrix of the true graph structure. The default value is TRUE.

Details

The graph pattern is generated as below:

genome-like: p variables are evenly partitions variables into g disjoint groups; the adjacent variables within each group are linked via an edge. With a probability alpha a pair of non-adjacent variables in the same group is given an edge. Variables in different groups are linked with an edge with a probability of beta.

Value

An object with S3 class "simgeno" is returned:

data	The generated data as an n by p matrix.
Theta	A p by p matrix corresponding to the inverse of covariance.
adj	A p by p matrix corresponding to the adjacency matrix of the true graph structure.
Sigma	A p by p covariance matrix for the generated data.
n.groups	The number of groups.
groups	A vector that indicates each variable belongs to which group.
sparsity	The sparsity levels of the true graph.

Author(s)

Pariya Behrouzi and Ernst C. Wit
Maintainer: Pariya Behrouzi <pariya.behrouzi@gmail.com>

References

1. Behrouzi, P., and Wit, E. C. (2017c). netgwas: An R Package for Network-Based Genome-Wide Association Studies. arXiv preprint, arXiv:1710.01236.
2. Behrouzi, P., and Wit, E. C. (2019). Detecting epistatic selection with partially observed genotype data by using copula graphical models. Journal of the Royal Statistical Society: Series C (Applied Statistics), 68(1), 141-160.

See Also

netsnp, and netgwas-package

Examples

```
#genome-like graph structure
sim1 <- simgeno(alpha = 0.01, beta = 0.02)
plot(sim1)

#genome-like graph structure with more edges between variables in a same or different groups
sim2 <- simgeno(adjacent = 3, alpha = 0.02 , beta = 0.03)
plot(sim2)

#simulate data
D <- simgeno(p=50, n=100, g=5, k= 3, adjacent = 3, alpha = 0.06 , beta = 0.08)
plot(D)

#Reconstructing intra- and inter-chromosomal conditional interactions (LD) network
out <- netsnp(data = D$data, n.rho= 4, ncores= 1)
plot(out)
#Select an optimal graph
sel <- selectnet(out)
plot(sel, vis= "CI" )
```

---

simRIL *Generate genotype data of RIL*

Description

Generating genotype data from a recombinant inbred line (RIL) population.

Usage

```
simRIL( d = 25, n = 200, g = 5, cM = 100, selfing=2 )
```

Arguments

d	The number of markers per chromosome. The default value is 25.
n	The number of sample size (observations). The default value is 200.
g	The number of linkage groups (chromosomes). The default value is 5.
cM	The length of each chromosome based on centiMorgan.
selfing	The number of selfing in RIL population.

Value

data	The generated RIL genotype data as an n by (d x g) matrix.
map	The genetic map of the data.

Author(s)

Pariya Behrouzi
Maintainer: Pariya Behrouzi <pariya.behrouzi@gmail.com>

See Also

netmap, netsnp, and netgwas-package

Examples

```
#genome-like graph structure
ril <- simRIL(g = 5, d = 25, cM = 100, n = 200, selfing = 2)
geno <- ril$data; image(geno, xlab= "individuals", ylab="markers")
map <- ril$map
```

---

tetraPotato *tetraploid potato genotype data*

Description

Tetraploid potato (*Solanum tuberosum L.*) genotype data.

Usage

```
data(tetraPotato)
```

Format

The format is a matrix containing 1972 single-nucleotide polymorphism (SNP) markers for 156 individuals.

Details

The full-sib mapping population MSL603 consists of 156 F1 plants resulting from a cross between female parent "Jacqueline Lee" and male parent "MSG227-2". The obtained genotype data contain 1972 SNP markers with five allele dosages. This genotype data can be used to construct linkage map for tetraploid potato (see below example).

Source

Massa, Alicia N., Norma C. Manrique-Carpintero, Joseph J. Coombs, Daniel G. Zarka, Anne E. Boone, William W. Kirk, Christine A. Hackett, Glenn J. Bryan, and David S. Douches. "Genetic linkage mapping of economically important traits in cultivated tetraploid potato (Solanum tuberosum L.)." G3: Genes, Genomes, Genetics 5, no. 11 (2015): 2357-2364.

Examples

```
data(tetraPotato)
#Shuffle the order of markers
potato <- tetraPotato[ ,sample(1:ncol(tetraPotato), ncol(tetraPotato), replace=FALSE)]
#Constructing linkage map for tetraploid potato
out <- netmap(potato, cross = "outbred"); out
potato.map <- out$map; potato.map
#plot(out)
```

---

thaliana *Arabidopsis thaliana phenotype and genotype data*

Description

The genotype data of the Kend-L x Col Recombinant Inbred Line (RIL) population along with flowering time and leaf numbers phenotype information.

Usage

```
data(thaliana)
```

Format

The format is a matrix containing 181 single-nucleotide polymorphism (SNP) markers and 8 phenotypes information for 197 individuals.

Details

The accession Kend-L (Kendalville-Lehle; Lehle-WT-16-03) is crossed to the common lab strain Col (Co\-lum\-bi\-a). The resulting lines were taken through six rounds of selfing without any intentional selection. The resulting 282 KendC (Kend-L $\times$ Col) lines were genotyped at 181 markers. The flowering time was measured for 197 lines of this population in both long days, which promote rapid flowering in many A. *thaliana* strains, and in short days. Flowering time was measured using days to flowering (DTF) as well as the total number of leaves (TLN), partitioned into rosette and cauline leaves. In total eight phenotypes have been measured, namely days to flowering (DTF), cauline leaf number (CLN), rosette leaf number (RLN), and total leaf number (TLN) in long days (LD), and DTF, CLN, RLN, and TLN in short days (SD). Thus, the final dataset consist of 197 observations for 189 variables (8 phenotypes and 181 genotypes - SNP markers)
This data set can be used to reconstruct network among SNP markers and the measured phenotypes.

Source

Balasubramanian, Sureshkumar, et al. (2009). "QTL mapping in new Arabidopsis thaliana advanced intercross-recombinant inbred lines." PLoS One 4.2: e4318.

Examples

```
## Not run:
data(thaliana)

# Graph path
out <- netphenogeno(thaliana, ncores=1)
plot(out)

sel <- selectnet(out)
plot(sel, vis= "interactive")

## End(Not run)
```

Index