Package 'FamilyBasedPGMs'

December 17, 2019

Type Package

Title Methods for Learning Genetic and Environmental Graphical Models from Family Data

Version 2.0

Author Adèle Helena Ribeiro

Maintainer Adèle Helena Ribeiro <adele@ime.usp.br>

Description This package provides methods for learning, from observational Gaussian family data (i.e., Gaussian data clusterized in families), Gaussian undirected and directed acyclic PGMs describing linear relationships among multiple phenotypes and a decomposition of the learned PGM into unconfounded genetic and environmental PGMs.
The structure learning is based on zero partial correlation tests, derived in the work by Ribeiro and Soler, entitled ``Learning Genetic and Environmental Graphical Models from Family Data" (submitted for publication). These tests are based on univariate polygenic linear mixed, with two components of variance: the polygenic or family-specific random effect, which models the phenotypic variability across the families, and the environmental or subject-specific error, which models phenotypic variability after removing the familial aggregation effect.
Particularly, for causal structure learning (learning of the structure of directed acyclic PGMs), these partial correlation tests are used as d-separation oracles in the IC/PC algorithm.

License GPL (>= 2)

Encoding UTF-8

LazyData true

Imports curl, pcalg, parallel, doMC, foreach, MASS, coxme, kinship2, R.utils, Matrix, methods, igraph, Rdpack

Suggests stargazer,
gpuR,
knitr,
rmarkdown

RdMacros Rdpack

RoxygenNote 6.1.1

VignetteBuilder knitr

Depends R (>= 3.5.0)

R topics documented:

`calculateBlockDiagonal2PhiMatrix`
Kinship matrix of a known pedigree structure

Description

Computes the block diagonal kinship matrix with the degree of relatedness between individuals all individuals of F families, multiplied by 2. It is usually denoted as $2\mathbf{\Phi}$, in which $\mathbf{\Phi} = diag\{\Phi^{(f)}, f = 1, \ldots, F\}$ and each entry $\Phi_{ij}^{(f)}$ is the probability that two alleles sampled at random from individuals i and j of family f are identical by descent. Thus, when the pedigree structure is known, $\Phi_{ii}^{(f)} = 1/2$, $\Phi_{ij}^{(f)} = 1/4$ if i and j are siblings or if one of them is a parent of the other, $\Phi_{ij}^{(f)} = 1/8$ if one of them is a grand-parent of the other, and so on (Lange 2003).

Usage

```
calculateBlockDiagonal2PhiMatrix(ped, squaredRoot = FALSE,
  sampled = NULL)
```

Arguments

`ped`	A data.frame with with the pedigrees of the families, with columuns `famid`, `id`, `dadid`, `momid`, and `sex` for all sampled and non-sampled subjects.
`squaredRoot`	a logical value indicating if the square root of the kinship matrix $2\mathbf{\Phi}$ (i.e., the Z matrix) must be computed.
`sampled`	A logical vector in which element i indicates whether individual i was sampled or not.

Value

The kinship matrix $2\mathbf{\Phi}$ or its squared root, if squaredRoot is TRUE.

References

Lange K (2003). *Mathematical and statistical methods for genetic analysis*. Springer Science \& Business Media.

`familyBasedCITest` *Conditional Independence Test for Family Data*

Description

Computes the p-value of the test of the null hypothesis of zero genetic, environmental, or total partial correlation between X and Y given S, if `dagg`, `dage`, or `dagt` is set as TRUE, respectively.

The unconfounded estimation and significance test for the genetic and environmental partial correlation coefficients are described in detail in Ribeiro and Soler (2019). They are based on univariate polygenic linear mixed models (Almasy and Blangero 1998), with two components of variance: the polygenic or family-specific random effect, which models the phenotypic variability across the families, and the environmental or subject-specific error, which models phenotypic variability after removing the familial aggregation effect.

This function can be used as a d-separation oracle in causal structure learning algorithms such as the IC/PC algorithm when the variables are normal distributed and the observations are clusterized in families.

Usage

```
familyBasedCITest(x, y, S, suffStat)
```

Arguments

`x`	An integer value indicating the position of variable X.
`y`	An integer value indicating the position of variable Y.
`S`	An integer vector with the positions of zero or more conditioning variables in S.
`suffStat`	A list with the elements "phen.df", "covs.df", "pedigrees", "minK", "maxFC", "orthogonal", "alpha", "dirToSave", "fileID", "savePlots", and "useGPU", as described in function `learnFamilyBasedDAGs`, "dagg", "dage", "dagt", to specify whether the genetic, environmental, or total conditional independence test should be conducted, and also "Phi2" and "Z", both obtained by the function `calculateBlockDiagonal2PhiMatrix` with the argument "squaredRoot" set, respectively, as FALSE and TRUE.

Value

The p-value of the test.

References

Almasy L, Blangero J (1998). "Multipoint quantitative-trait linkage analysis in general pedigrees." *The American Journal of Human Genetics*, **62**(5), 1198–1211.

Ribeiro AH, Soler JMP (2019). "Learning Genetic and Environmental Graphical Models from Family Data." *Submitted for publication*.

`learnFamilyBasedDAGs` *Learning of Gaussian Directed Acyclic PGMs from Family Data*

Description

The CPDAG representing the Markov equivalence classes to which the Gaussian directed acyclic PGM belong and its decomposition into genetic and environmental components are learned from observational family data by applying the IC/PC algorithm (Pearl 2000; Spirtes et al. 2000) with the zero partial correlation tests derived in the work by Ribeiro and Soler (2019) as d-separation oracles.

These tests are based on univariate polygenic linear mixed models (Almasy and Blangero 1998), with two components of variance: the polygenic or family-specific random effect, which models the phenotypic variability across the families, and the environmental or subject-specific error, which models phenotypic variability after removing the familial aggregation effect.

Usage

```
learnFamilyBasedDAGs(phen.df, covs.df, pedigrees, sampled, fileID,
  dirToSave, alpha = 0.05, max_cores = NULL, minK = 10,
  maxFC = 0.01, orthogonal = TRUE, hidden_vars = FALSE,
  maj.rule = TRUE, useGPU = FALSE, debug = TRUE, savePlots = FALSE,
  logFile = NULL)
```

Arguments

`phen.df`	A data.frame with phenotype variables of only sampled subjects. Column names must be properly set with the names of the phenotypes.
`covs.df`	A data.frame with covariates of only sampled subjects. Column names must be properly set with the names of the covariates.
`pedigrees`	A data.frame with columuns `famid`, `id`, `dadid`, `momid`, and `sex` columns for all sampled and non-sampled subjects.
`sampled`	A logical vector in which element i indicates whether individual i was sampled or not.
`fileID`	A character string to be used as prefix in the filenames of RData objects with the partial correlation results. Note that covariates are not identified in these files.
`dirToSave`	Path to the folder you want to save the output objects.
`alpha`	The significance level to be used in the partial correlation tests.
`max_cores`	An integer indicating the maximum number of CPU cores to be used for parallel execution.
`minK`	A scalar indicating the minimum dimension allowed in the dimensionality reduction for confounding correction.
`maxFC`	A scalar between 0 and 1, indicating the maximum fraction of confounding allowed.
`orthogonal`	A logical value indicating whether the transformation matrix used in the confounding correction is orthogonal or not.
`hidden_vars`	A logical value indicating if the causal structure learning method should account for hidden variables. The `rfci` algorithm is used if hidden_vars is `TRUE` and the `pc` algorithm is used otherwise.

maj.rule	A logical value to be used in the skeleton function, indicating whether the majority rule must be applied or not.
useGPU	A logical value indicating whether GPU cores can be used for parallel execution.
debug	A logical value indicating whether some debug messages can be shown.
savePlots	A logical value indicating whether plots for the confounding correction must be generated.
logFile	Optional file path and name to save progress and error messages. If not provided and debug is True a default file is created in the dirToSave folder.

Value

Returns a list with the partial correlation matrices (pcor), the adjacency matrices (adjM), and with the igraph objects representing the undirected PGM (udg).

References

Almasy L, Blangero J (1998). "Multipoint quantitative-trait linkage analysis in general pedigrees." *The American Journal of Human Genetics*, **62**(5), 1198–1211.

Pearl J (2000). *Causality: models, reasoning, and inference*. Cambridge University Press, Cambridge, UK.

Ribeiro AH, Soler JMP (2019). "Learning Genetic and Environmental Graphical Models from Family Data." *Submitted for publication*.

Spirtes P, Glymour CN, Scheines R (2000). *Causation, prediction, and search*, volume 81. MIT press.

Examples

```
data(scen3) # available simulated datasets are scen1, scen2, scen3, and scen4

scenario = 3 # data was simulated according to scenario 3

fam.nf <- scen3$fam.nf
pedigrees <- scen3$pedigrees
phen.df <- scen3$phen.df[[1]] # accessing the first replicate
covs.df <- NULL # no covariates were used in the simulation process.

N <- sum(fam.nf) # total number of individuals
sampled <- rep(1, N) # in simulated data, all individuals were sampled.

fileID <- paste0("scen", scenario)
dirToSave <- paste0("./objects-PC-", fileID, "/")
dir.create(dirToSave, showWarnings=FALSE)

alpha = 0.05

dags <- learnFamilyBasedDAGs(phen.df, covs.df, pedigrees, sampled,
                             fileID, dirToSave, alpha, max_cores=NULL,
                             minK=10, maxFC = 0.05, orthogonal=TRUE,
                             hidden_vars=FALSE, maj.rule=TRUE,
                             useGPU=FALSE, debug=TRUE, savePlots=FALSE)
```

```
# the adjacency matrix of the learned directed acyclic genetic PGM
as(dags$g, "amat")

# plotting the the learned directed acyclic genetic PGM as an `igraph` object:
plot.igraph(graph.adjacency(adjM_g), vertex.size=30, vertex.color="lightblue")

# the adjacency matrix of the learned directed acyclic environmental PGM
as(dags$e, "amat")

# plotting the the learned directed acyclic environmental PGM as an `igraph` object:
plot.igraph(graph.adjacency(adjM_e), vertex.size=30, vertex.color="lightblue")
```

`learnFamilyBasedUDGs` *Learning of Gaussian Undirected PGMs from Family Data*

Description

A Gaussian undirected PGM and its decomposition into genetic and environmental components are learned from observational family data by assigning an edge between every pair of variables such that the partial correlation between the two variables in question (in the respective component) given all the other variables is significantly different from zero.

The zero partial correlation tests are derived in the work by Ribeiro and Soler (2019). These tests are based on univariate polygenic linear mixed models (Almasy and Blangero 1998), with two components of variance: the polygenic or family-specific random effect, which models the phenotypic variability across the families, and the environmental or subject-specific error, which models phenotypic variability after removing the familial aggregation effect.

Usage

```
learnFamilyBasedUDGs(phen.df, covs.df, pedigrees, sampled, fileID,
  dirToSave, alpha = 0.05, correction = NULL, max_cores = NULL,
  minK = 10, maxFC = 0.01, orthogonal = TRUE, useGPU = FALSE,
  debug = TRUE, logFile = NULL)
```

Arguments

`phen.df`	A data.frame with phenotype variables of only sampled subjects. Column names must be properly set with the names of the phenotypes.
`covs.df`	A data.frame with covariates of only sampled subjects. Column names must be properly set with the names of the covariates.
`pedigrees`	A data.frame with columuns `famid`, `id`, `dadid`, `momid`, and `sex` columns for all sampled and non-sampled subjects.
`sampled`	A logical vector in which element i indicates whether individual i was sampled or not.
`fileID`	A character string to be used as prefix in the filenames of R objects with the partial correlation results. Note that covariates are not identified in these files.
`dirToSave`	Path to the folder you want to save the output objects.

alpha	The significance level to be used in the partial correlation tests.
correction	A character string indicating the correction method to be used in the p.adjust function. The options are: "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", and "none".
max_cores	An integer value indicating the maximum number of CPU cores to be used for parallel execution.
minK	A scalar indicating the minimum dimension allowed in the dimensionality reduction for confounding correction.
maxFC	A scalar between 0 and 1 indicating the maximum fraction of confounding allowed.
orthogonal	A logical value indicating whether the transformation matrix used in the confounding correction is orthogonal or not.
useGPU	A logical value indicating whether GPU cores can be used for parallel execution.
debug	A logical value indicating whether some debug messages can be shown.
logFile	Optional file path and name to save progress and error messages. If not provided and debug is True a default file is created in the dirToSave folder.

Value

Returns a list with the following elements:

pcor A list with the total (pcor_t), genetic (pcor_g), and environmental (pcor_e) partial correlation matrices.

adjM A list with the total (t), genetic (g), and environmental (e) partial adjacency matrices.

udg A list with the total (t), genetic (g), and environmental (e) igraph objects representing the respective undirected graphs.

References

Almasy L, Blangero J (1998). "Multipoint quantitative-trait linkage analysis in general pedigrees." *The American Journal of Human Genetics*, **62**(5), 1198–1211.

Ribeiro AH, Soler JMP (2019). "Learning Genetic and Environmental Graphical Models from Family Data." *Submitted for publication*.

Examples

```
data(scen3) # available simulated datasets are scen1, scen2, scen3, and scen4

scenario = 3 # data was simulated according to scenario 3

fam.nf <- scen3$fam.nf
pedigrees <- scen3$pedigrees
phen.df <- scen3$phen.df[[1]] # accessing the first replicate
covs.df <- NULL # no covariates were used in the simulation process.

N <- sum(fam.nf) # total number of individuals
sampled <- rep(1, N) # in simulated data, all individuals were sampled.

fileID <- paste0("scen", scenario)
dirToSave <- paste0("./objects-UDG-", fileID, "/")
```

```
dir.create(dirToSave, showWarnings=FALSE)
alpha = 0.05

udgs.out <- learnFamilyBasedUDGs(phen.df, covs.df, pedigrees, sampled,
                                 fileID, dirToSave, alpha, correction=NULL,
                                 max_cores=NULL, minK=10, maxFC = 0.05,
                                 orthogonal=TRUE, useGPU=FALSE, debug=TRUE)

# the adjacency matrix of the learned undirected genetic PGM
udgs.out$adjM$g

# the estimates, p-values, and effective sizes of the genetic partial correlations
udgs.out$pCor$pCor_g

# plotting the the learned undirected genetic PGM as an `igraph` object:
plot(udgs.out$udg$g, vertex.size=30, vertex.color="lightblue")

#' # the adjacency matrix of the learned undirected environmental PGM
udgs.out$adjM$e

# the estimates, p-values, and effective sizes of the environmental partial correlations
udgs.out$pCor$pCor_e

# plotting the the learned undirected environmental PGM as an `igraph` object:
plot(udgs.out$udg$e, vertex.size=30, vertex.color="lightblue")
```

`plotFamilyPedigree` *Plotting of Pedigree Chart*

Description

Creates a chart of the pedigree of family `famid`.

Usage

```
plotFamilyPedigree(pedigrees, famid)
```

Arguments

`pedigrees`	A data.frame object with at least the columns: "famid", "id", "momid", "dadid", and "sex"
`famid`	A integer value identifying the family for which the pedigree chart should be plotted.

Examples

```
data(scen1)
pedigrees <- scen1$pedigrees

# plot the pedigree chart of the fifth simulated family
plotFamilyPedigree(pedigrees, 5)
```

scen1 *Simulated Data - Scenario 1*

Description

It contains 100 replicates of data simulated according to scenario 1, as described in (Ribeiro and Soler 2019).

Usage

```
scen1
```

Format

A list with the following elements:

pedigrees A data.frame representing the pedigrees of all simulated families. It contains the columns `famid`, `id`, `dadid`, `momid`, and `sex`.

fam.nf A integer vector where the entry i indicates the number of individuals at family i.

phen.nf A list of size 100 in which each element represents a data.frame with the values of the phenotypes X, Y, and Z for all 900 individuals.

Details

In this scenario, the true DAG in the total, genetic, and environmental components is the unshielded collider $X \rightarrow Z \leftarrow Y$ and its moral graph is complete.

It was simulated pedigrees for 30 families, each with 30 individuals (N = 900).

The same pedigrees were used for simulating 100 replicates of family data for three Gaussian phenotypes, namely X, Y, and Z. No covariates were used in the simulation.

References

Ribeiro AH, Soler JMP (2019). "Learning Genetic and Environmental Graphical Models from Family Data." *Submitted for publication.*

Examples

```
data(scen1)
```

scen2 *Simulated Data - Scenario 2*

Description

It contains 100 replicates of data simulated according to scenario 2, as described in (Ribeiro and Soler 2019).

Usage

```
scen2
```

Format

A list with the following elements:

pedigrees A data.frame representing the pedigrees of all simulated families. It contains the columns `famid`, `id`, `dadid`, `momid`, and `sex`.

fam.nf A integer vector where the entry i indicates the number of individuals at family i.

phen.nf A list of size 100 in which each element represents a data.frame with the values of the phenotypes X, Y, and Z for all 900 individuals.

Details

In this scenario, the true DAG in the total, genetic, and environmental components is the fork $Y \leftarrow X \rightarrow Z$ and its moral graph is exactly its undirected version.

It was simulated pedigrees for 30 families, each with 30 individuals (N = 900).

The same pedigrees were used for simulating 100 replicates of family data for three Gaussian phenotypes, namely X, Y, and Z. No covariates were used in the simulation.

References

Ribeiro AH, Soler JMP (2019). "Learning Genetic and Environmental Graphical Models from Family Data." *Submitted for publication.*

Examples

```
data(scen2)
```

scen3 *Simulated Data - Scenario 3*

Description

It contains 100 replicates of data simulated according to scenario 3, as described in (Ribeiro and Soler 2019).

Usage

```
scen3
```

Format

A list with the following elements:

pedigrees A data.frame representing the pedigrees of all simulated families. It contains the columns `famid`, `id`, `dadid`, `momid`, and `sex`.

fam.nf A integer vector where the entry i indicates the number of individuals at family i.

phen.nf A list of size 100 in which each element represents a data.frame with the values of the phenotypes X, Y, and Z for all 900 individuals.

Details

In this scenario, the true DAG in the total, genetic, and environmental components is $Z \leftarrow X \rightarrow Y \rightarrow Z$ and its moral graph is complete. However, data for the total DAG may be unfaithful, since $\rho_{X,Y|Z}$ is almost zero.

It was simulated pedigrees for 30 families, each with 30 individuals (N = 900).

The same pedigrees were used for simulating 100 replicates of family data for three Gaussian phenotypes, namely X, Y, and Z. No covariates were used in the simulation.

References

Ribeiro AH, Soler JMP (2019). "Learning Genetic and Environmental Graphical Models from Family Data." *Submitted for publication*.

Examples

```
data(scen3)
```

scen4 *Simulated Data - Scenario 4*

Description

It contains 100 replicates of data simulated according to scenario 4, as described in (Ribeiro and Soler 2019).

Usage

```
scen4
```

Format

A list with the following elements:

pedigrees A data.frame representing the pedigrees of all simulated families. It contains the columns `famid`, `id`, `dadid`, `momid`, and `sex`.

fam.nf A integer vector where the entry i indicates the number of individuals at family i.

phen.nf A list of size 100 in which each element represents a data.frame with the values of the phenotypes X, Y, and Z for all 900 individuals.

Details

In this scenario, the DAG of the total and environmental components is the unshielded collider $X \rightarrow Z \leftarrow Y$. However, the edge $X \rightarrow Z$ is absent in the genetic PGM, i.e., the genetic PGM has only the edge $Z \leftarrow Y$.

It was simulated pedigrees for 30 families, each with 30 individuals (N = 900).

The same pedigrees were used for simulating 100 replicates of family data for three Gaussian phenotypes, namely X, Y, and Z. No covariates were used in the simulation.

References

Ribeiro AH, Soler JMP (2019). "Learning Genetic and Environmental Graphical Models from Family Data." *Submitted for publication*.

Examples

```
data(scen4)
```

scen5 *Simulated Data - Scenario 5*

Description

It contains 100 replicates of data simulated according to scenario 5, as described in (Ribeiro and Soler 2019).

Usage

```
scen5
```

Format

A list with the following elements:

pedigrees A data.frame representing the pedigrees of all simulated families. It contains the columns `famid`, `id`, `dadid`, `momid`, and `sex`.

fam.nf A integer vector where the entry i indicates the number of individuals at family i.

phen.nf A list of size 100 in which each element represents a data.frame with the values of the phenotypes X, Y, and Z for all 900 individuals.

Details

In this scenario, the true DAG in the total, genetic, and environmental components is $X \rightarrow Z \leftarrow U \rightarrow W \leftarrow Y$. Since U is hidden, the true MAG is $X \rightarrow Z \leftrightarrow W \leftarrow Y$.

It was simulated pedigrees for 30 families, each with 30 individuals (N = 900).

The same pedigrees were used for simulating 100 replicates of family data for five Gaussian phenotypes, namely X, Y, Z, W, and U. No covariates were used in the simulation.

References

Ribeiro AH, Soler JMP (2019). "Learning Genetic and Environmental Graphical Models from Family Data." *Submitted for publication.*

Examples

```
data(scen5)
```

`simulatePedigrees` *Pedigree Simulation*

Description

Pedigree Simulation

Usage

```
simulatePedigrees(fam.nf, nInitFounderCouples, probToMarry, probChildren)
```

Arguments

`fam.nf`	A integer vector where the entry i indicates the number of individuals at family i.
`nInitFounderCouples`	A integer value representing the number of founders in the first generation.
`probToMarry`	A scalar between 0 and 1 indicating the probability of a individual getting married. Marriages only occur between individuals of the same generation.
`probChildren`	A vector with values between 0 and 1 in which entry i indicates the probability of a couple has i children.

Examples

```
fam.nf <- rep(10, 20)  # for 20 families with 10 individuals
nInitFounderCouples <- 1
probChildren <- c(0, 0, 0.3, 0.3, 0.3, 0.1)
probToMarry = 0.85
pedigrees <- simulatePedigrees(fam.nf, nInitFounderCouples, probToMarry, probChildren)
```

Index