Package ‘geneNetBP’

March 17, 2016

Type Package

Title Belief Propagation in Genotype-Phenotype Networks

Version 1.0.0

Date 2016-02-29

Author Janhavi Moharil <janhavim@buffalo.edu>

Maintainer Janhavi Moharil <janhavim@buffalo.edu>

Depends R (>= 3.1.0),scales (>= 0.2.4),Rgraphviz (>= 2.8.1), graph (>= 1.42.0)

Suggests RHugin

Additional_repositories http://rhugin.r-forge.r-project.org/

Description Belief propagation methods in genotype-phenotype networks (Conditional Gaussian Bayesian Networks) to propagate phenotypic evidence through the network.

License GPL-2

Repository R-Forge

Repository/R-Forge/Project genenetbp

Repository/R-Forge/Revision 73

Repository/R-Forge/DateTimeStamp 2016-03-17 19:34:52

Date/Publication 2016-03-17 19:34:52

NeedsCompilation no

R topics documented:

geneNetBP-package *Belief Propagation in Genotype-Phenotype Networks*

Description

The geneNetBP package provides functions to fit Conditional Gaussian Bayesian network to genotype-phenotype or Quantitative Trait Loci (QTL) data, absorb evidence in the genotype-phenotype network and quantify and visualize the changes in network beliefs. The package makes extensive use of `RHugin` package that provides an R interface for the Hugin Decision Engine. The package RHugin can be obtained from R-Forge by accessing the link given below. All the functions in the geneNetBP package are documented. For a complete list of functions, use the command ls(pos = "package:geneNetBP").

Details

Package:	geneNetBP
Type:	Package
Version:	1.0.0
Date:	2016-02-29
License:	GPL-2

This package implements structure learning and conditional probability learning methods from RHugin package to Quantative Trait Loci (QTL) data and provides functions to query and visualize the network beliefs. The function `fit.gnbp` can be used to learn conditional gaussian bayesian networks from genotype-phenotype data. The main function for absorbing evidence and quantifying the changes in beliefs is `absorb.gnbp`. A `plot` method is available for visualizing network beliefs.

All functions in the package work on RHugin domains. Since Hugin domains are external pointers and cannot be saved in R workspace, the RHugin package provides functions `read.rhd` and `write.rhd` for loading and saving the Hugin domains. See RHugin documentation for more information.

A vignette illustrating the use of geneNetBP is available.

Author(s)

Janhavi Moharil <janhavim@buffalo.edu> Rachael Hageman Blair <hageman@buffalo.edu>

Maintainer: Janhavi Moharil <janhavim@buffalo.edu>

References

Janhavi Moharil, Paul May, Daniel P. Gaile, Rachael Hageman Blair (2016). "Belief Propagation in Genotype-Phenotype Networks", *Stat Appl Genet Mol Biol*, 15(1):39-53.

The RHugin Project website: http://rhugin.r-forge.r-project.org

HUGIN EXPERT website: http://www.hugin.com

HUGIN API Reference Manual http://download.hugin.com/webdocs/manuals/api-manual.pdf

absorb.gnbp *Absorb evidence and infer a genotype-phenotype network*

Description

Absorb a single piece or a spectrum of evidence for one or more continuous nodes in a compiled RHugin domain, obtain the updated beliefs and the Jeffrey's signed information.

Usage

```
absorb.gnbp(gpfit, node, evidence)
```

Arguments

`gpfit` an object of class "gpfit". Output from `fit.gnbp`.

`node` a character vector specifying the names of the nodes for which the evidence is to be absorbed.

`evidence` a matrix or a numeric vector of evidence. number of rows of the matrix or the length of the vector should be equal to the length of node.

Value

absorb.gnbp returns an object of class "gnbp". The functions summary and print can be used for objects of class "gnbp". An object of class "gnbp" is a list containing the following components

`gp` an RHugin domain that is triangulated, compiled and with the latest absorbed evidence propagated

`gp_flag` type of network

`node` a character vector specifying the nodes for which evidence has been absorbed

`marginal` a list of marginal probabilities for phenotypes (pheno) and genotypes (geno)

`belief` a list of updated beliefs for phenotypes (pheno) and genotypes (geno)

`JSI` a matrix of Jeffrey's signed information if network is `Conditional Gaussian`, otherwise `NULL` if network is `Discrete Bayesian`

`FC` a list of two. a matrix `FC` of fold changes and a matrix `pheno_state` of phenotype node beliefs - state with maxium probability. If network is `Conditional Gaussian`, a `NULL` value is returned.

The marginals, beliefs and JSI or FC are calculated for only *d*-connected nodes.

If a sequence of evidence is absorbed for a single node in a Conditional Gaussian network, a plot of `JSI` vs `evidence` is drawn.

Author(s)

Janhavi Moharil<janhavim@buffalo.edu>

Rachael Hageman-Blair<hageman@buffalo.edu>

See Also

`gen.evidence`

Examples

```
data(mouse)

## Not run:
## Simple example : Fit a bayesian network to genotype-phenotype data
network<-fit.gnbp(mousegeno,mousepheno,alpha=0.1)

## Absorb a single evidence for a single node
absorb.gnbp(network,node="Tlr12",evidence=matrix(2.5))

## Absorb a sequence of evidence for a single node generated using \code{\link{gen.evidence}
network<-fit.gnbp(mousegeno,mousepheno,alpha=0.1)
evidence<-gen.evidence(network,node="Tlr12")
absorb.gnbp(network,node="Tlr12",evidence=evidence)

##Absorb sequence of evidence for multiple nodes}
network<-fit.gnbp(mousegeno,mousepheno,alpha=0.1)
evidence<-gen.evidence(network,node=c("Ak2","Ptp4a2","Hmgcl"),std=2,std.equal=TRUE)
absorb.gnbp(network,node=rownames(evidence),evidence=evidence)

## End(Not run)
```

---

fit.gnbp *Fit a Conditional Gaussian Bayesian Network to QTL data*

Description

Learn the structure of a genotype-phenotype network from quantitative trait loci (QTL) data and the conditional probability table for each node in the network using the PC algorithm and the EM algorithm implemented in the RHugin package.

Usage

```
fit.gnbp(geno,pheno,constraints,learn="TRUE",edgelist,type ="cg",
                  alpha=0.001,tol=1e-04,maxit=0)
```

Arguments

geno	a data frame of column vectors of class factor (or one that can be coerced to that class) and non-empty column names.
pheno	a data frame of column vectors of class numeric if `type = "cg"` or class factor if `type = "db"` and non-empty column names.
constraints	an optional list of constraints on the edges for specifying required and forbidden edges. See details.
learn	a boolean value. If TRUE (default), the network structure will be learnt using the PC algorithm in RHugin package. If FALSE, only conditional probabilities will be learnt.
edgelist	a list of edges to be provided if learn == FALSE.

`type`	specify the type of network. `"cg"` for `Conditional Gaussian` (default) and `"db"` for `Discrete Bayesian`.
`alpha`	a single numeric value specifying the significance level (for use with RHugin). Default is 0.001.
`tol`	a positive numeric value (optional) specifying the tolerance for EM algorithm to learn conditional probability tables (for use with RHugin). Default value is 1e-04. See `learn.cpt` for details.
`maxit`	a positive integer value (optional) specifying the maximum number of iterations of EM algorithm to learn conditional probability tables (for use with RHugin). See `learn.cpt` for details.

Details

The function fit.gnbp fits a conditional gaussian bayesian network at the specified level of significance alpha, to genotype-phenotype (QTL) data by the PC algorithm implemented in the RHugin package. The conditional probability tables are learnt for each node in the domain by the EM algorithm implemented in the RHugin package.

Edges between the genotypes at SNP markers are not allowed and the genotypes are constrained to precede the phenotypes. The phenotypes should be either all numeric or all discrete. The function does not currently support mixture of discrete and continuous phenotypes. Additional domain knowledge in terms of edges should be provided as a list of constraints, the structure of which is described in detail in `learn.structure`. Briefly, the constraints argument is a list of two elements: directed and undirected. Each of these elements in turn should be a list with two elements: required and forbidden. The elements of required and forbidden must be a character vector of length two specifying the names of the nodes. See `learn.cpt` for details.

Value

Returns an object of class "gpfit".

`gp`	a pointer to a compiled RHugin domain. There is a cpt table associated with each node in the network.
`gp_nodes`	a data frame containing information about nodes for internal use with other functions.
`gp_flag`	a character string specifying the type of network (`Conditional Gaussian` or `Discrete Bayesian`)

Author(s)

Janhavi Moharil <janhavim@buffalo.edu>

Rachael Hageman-Blair<hageman@buffalo.edu>

See Also

`absorb.gnbp`

Examples

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

## Simple example : Fit a bayesian network to genotype-phenotype data using the default values
```

```
fit.gnbp(mousegeno,mousepheno)

## Fit a bayesian network to genotype-phenotype data at a specified significance level
fit.gnbp(mousegeno,mousepheno,alpha = 0.1)

## End(Not run)
```

gen.evidence *Generate a sequence of evidence for a continuous node in a conditional gaussian bayesian network.*

Description

The evidence for a node in an RHugin domain is generated as a linear sequence within the specified standard deviation from the marginal mean of the node. The evidence can be given as an input to absorb.gnbp

Usage

```
gen.evidence(gpfit, node, std = 2, length.out = 10, std.equal = TRUE)
```

Arguments

`gpfit`	an object of class "gpfit" obtained by using fit.gnbp
`node`	a character string specifying the name of a continuous node in the domain
`std`	a numeric value specifying the number of standard deviations of marginal distribution within which the evidence is generated. A numeric vector of length = number of nodes, must be specified when std.equal=FALSE.
`length.out`	a positive integer giving the desired length of the sequence.
`std.equal`	a logical value indicating whether same number of standard deviations should be used to generate evidence for all nodes. Default is TRUE.

Details

The evidence for a node in an RHugin domain is generated as a linear sequence within the specified standard deviation from the marginal mean of the node. The evidence can be given as an input to absorb.gnbp

Value

A matrix of evidence for each specified node

Author(s)

Janhavi Moharil<janhavim@buffalo.edu>

Rachael Hageman-Blair<hageman@buffalo.edu>

See Also

`absorb.gnbp`,`fit.gnbp`

Examples

```
##Fit a network
data(mouse)
## Not run:
network<-fit.gnbp(mousegeno,mousepheno,alpha=0.1)

##Generate a sequence of evidence for a single node
evidence<-gen.evidence(network,node="Tlr12",std=2,length.out=20)

## End(Not run)
```

mouse *Mus Musculus Kidney eQTL data from Hageman et. al. (2011)*

Description

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

Usage

```
data(mouse)
```

Format

The data set `mouse` contains the following variables:

- mousegeno a data frame of 173 observations of 5 variables. All column vectors are of class factor
- mousepheno a data frame of 173 observations of 14 variables. All column vectors are of class numeric

Details

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

Source

Hageman,R.S., Leduc,M.S., Caputo,C.R., Tsaih,S.-W., Churchill, G.A., and Korstanje,R.(2011). "Uncovering genes and regulatory pathways related to urinary albumin excretion." *Journal of the American Society of Nephrology22*, 1, 73-81.

Examples

```
## Not run:
# load the data and fit a genotype-phenotype network
data(mouse)
fit.gnbp(mousegeno,mousepheno)

## End(Not run)
```

`plot.gnbp` *Plot a Genotype-Phenotype Network*

Description

Plot method for gnbp objects. Plots genotype-phenotype network in which evidence has been absorbed and propagated and maps the beliefs or Jeffrey's signed information onto the network.

Usage

```
## S3 method for class 'gnbp'
plot(x, y="JSI",col.palette,col.length = 100, ncol = 1,
                        fontsize=10, fontcolor="black",...)
```

Arguments

`x` An object of class gnbp

`y` A character string. Valid options are `"JSI"` (default) or `"belief"` for Conditional Gaussian network. For Discrete Bayesian networks, the argument will be ignored and the phenotype states with maximum probability will be plotted.

`col.palette` A list of character strings. For Conditional Gaussian networks, a list of 6 elements specifying colors for colormap.All 6 elements should be character strings specifying the colour for `pos_high`= high end of gradient of positive values (default = "red") `pos_lowi`=low end of gradient of positive values (default = "wheat1") `neg_high`=high end of gradient of positive values (default = "cyan") `neg_low`=low end of gradient of positive values (default = "blue") `dsep_col`= *d*-separated nodes (default = "white") `qtl_col`= discrete nodes (QTLs) (default = "grey") `node_abs_col`= nodes for which evidence has been absorbed (default = "palegreen2")

For Discrete Bayesian networks, a list of 4 elements specifying colors for colormap should be specified. All 4 elements should be character strings specifying the colour for `col_nodes`- a vector of colors for phenotype states should be specified. The length of the vector should be equal to the maximum number of phenotype states possible. `dsep_col`= *d*-separated nodes (default = "white") `qtl_col`= discrete nodes (QTLs) (default = "grey") `node_abs_col`= nodes for which evidence has been absorbed (default = "palegreen2")

`col.length` a positive integer (default = 100) specifying the resolution of the colormap (number of colors).

`ncol` a positive integer specifying the column number of JSI / belief / FC to plot. By default, the first column will be plot

`fontsize` fontsize for node labels

`fontcolor` fontcolor for node labels

`...` further arguments to the function `plot`. These will be ignored

Details

This function plots the genotype-phenotype network in which evidence has been absorbed and propagated. The `plot` method will convert the RHugin domain into an object of class "graphNEL" by using Rgraphviz package. The argument `nodeAttrs` to `plot` method for `graph` objects in

Rgraphviz package is then used to customize the plot. Users can specify the colormap options such as end colors for the positive and negative gradients and the resolution of the colormap. The default node shapes are "ellipse" for the phenotype nodes and "box" for genotype nodes.The *d*-separated nodes are white while the colored nodes are *d*-connected, with the color indicating the strength and direction of change. The node for which evidence is absorbed is colored green (default color).

Value

x is invisibly returned

Author(s)

Janhavi Moharil<janhavim@buffalo.edu>

Rachael Hageman-Blair<hageman@buffalo.edu>

See Also

absorb.gnbp

Examples

```
## Fit, absorb and plot a genotype-phenotype network
data(mouse)
## Not run:
network<-fit.gnbp(mousegeno,mousepheno,alpha=0.1)
network<-absorb.gnbp(network,node="Tlr12",evidence=matrix(-0.99))
plot(x=network)

## End(Not run)
```

toy *Toy example dataset*

Description

Toy example, Simulated data set

Usage

```
data(toy)
```

Format

The data set toy contains the following variables:

- toygeno a data frame of 500 observations of 3 variables. All column vectors are of class factor
- toypheno a data frame of 500 observations of 6 variables. All column vectors are of class numeric

Details

The simulated dataset consists of 3 genotypes each with 2 states and 6 phenotypes.

Examples

```
## Not run:
# load the data and fit a genotype-phenotype network
data(toy)
fit.gnbp(toygeno,toypheno)
## End(Not run)
```

yeast *Saccharomyces Cerevisiae eQTL data from Kruglak et. al. (2005)*

Description

eQTL data from 112 F1 segregants from a cross between BY4716 and RM11-1a strains of *Saccharomyces Cerevisiae*.

Usage

```
data(yeast)
```

Format

The data set `yeast` contains the following variables:

- yeastgeno a data frame of 112 observations of 9 variables. All column vectors are of class factor
- yeastpheno a data frame of 112 observations of 16 variables. All column vectors are of class numeric

Details

The `yeast` dataset is a subset of the widely studied yeast expression dataset comprising of 112 F1 segregants from a cross between BY4716 and RM11-1a strains of *Saccharomyces Cerevisiae*. The original dataset consists of expression values reported as log2(sample/ BY reference) for 6216 genes. The data can be accessed in Gene Expression Omnibus (GEO) by accession number (GSE1990). After linkage analysis and filtering based on location and significance of QTL, a final set of 16 genes and their corresponding 9 SNP markers were identified and included in the yeast dataset.\

Thus the final dataset has 2 variables `yeastgeno`, data frame of 112 observations (genotype) of 9 variables (SNP markers) and `yeastpheno`, data frame of 112 observations (normalized gene expression) of 16 variables (genes).\

Source

Brem RB, Kruglyak L. The landscape of genetic complexity across 5,700 gene expression traits in yeast. Proc Natl Acad Sci U S A 2005 Feb 1;102(5):1572-7.\

Brem RB, Storey JD, Whittle J, Kruglyak L. Genetic interactions between polymorphisms that affect gene expression in yeast. Nature 2005 Aug 4;436(7051):701-3.\

Examples

```
## Not run:
# load the data and fit a genotype-phenotype network
data(yeast)
fit.gnbp(yeastgeno,yeastpheno)

## End(Not run)
```

Index