Package ‘qtlnet.prior’

February 1, 2015

Title Causal network inference incorporating knowledge
Version 0.7
Date 2015-02-01
Author Jee Young Moon and Brian S. Yandell
Maintainer Jee Young Moon <jeeyoung.moon@gmail.com>
Description It infers a causal network from gene expression and genotypes with prior knowledge
License GPL-3
Depends R(>= 2.10), qtlnet(>= 1.3.6)

R topics documented:

qtlnet.prior-package .................................................. 1
all.config ............................................................... 2
mcmc.qtlnet.prior ...................................................... 3
node.nbhd.size ......................................................... 8
partition.eff ............................................................ 9
propose.new.beta .................................................... 10
propose.new.node.structure ........................................ 11
score.beta ............................................................. 12
score.model.qtlnet.prior ............................................ 13
score.prior ........................................................... 14
sim.example1.cross .................................................. 15
sim.knowledge ......................................................... 16

Index 18

qtlnet.prior-package Causal network inference incorporating knowledge

Description
It infers a causal network from gene expression and genotypes with prior knowledge
Details

Package: qtlnet.prior
Type: Package
Version: 0.7
Date: 2015-02-01
License: GPL-3

Author(s)
Jee Young Moon and Brian S. Yandell
Maintainer: Jee Young Moon <jeeyoung.moon@gmail.com>

References

See Also
The main function in the package is mcmc.qtlnet.prior.

all.config

Columnwise calculation of |B-G| for each parent configuration

Description
|B-G| is calculated for each column i: |B[i,i] - G[i,i]|. For each child node/column i, all possible parent configurations up to max.parents.partition are considered and |B[i,i] - G[i,i]| is calculated for each parent configuration k.

Usage
all.config(n.pheno, max.parents.partition, B)

Arguments

n.pheno
Number of phenotypes.

max.parents.partition
Maximum number of parents to consider for partition function calculation.

B
A list of biological knowledge matrices.
mcmc.qtlnet.prior

Value
A list with components

- **e.value**
  It is an array of size \((n.pheno, \text{count.parents.config}, \text{number of types of knowledge})\). \(e.value[i,k,j]\) is \(|B[i,i] - G[i,i]|\) for child node \(i\), parent configuration \(k\), and \(j\)-th knowledge. It is calculated by \(\text{sum}(1 - B[j][k, i]) + \text{sum}(B[j][-k, i])\).

- **count.parents.config**
  Total number of parent configurations from no parent up to \(\text{max.parents.partition}\) parents for a node.

Author(s)
Jee Young Moon and Brian S. Yandell

References

See Also
partition.eff

Examples
B <- runif(25, 0, 1)
B <- matrix(B, ncol=5)
diag(B) <- 0
B <- list(B)
ex.config <- all.config(n.pheno=5, max.parents.partition=3, B)

mcmc.qtlnet.prior

MCMC run for causal network inference with a prior knowledge

Description
It runs an MCMC to infer a causal network of genotypes and phenotypes incorporating knowledge as a prior.

Usage
mcmc.qtlnet.prior(cross, B = NULL, nogenotype=FALSE, pheno.col=NULL, addcov = NULL, intcov = NULL, nSamples = 3000, thinning = 10, burnin = 0.1, random.seed = NULL, max.parents = 3, M = NULL, init.edges = 0, beta0 = NULL, lambda = NULL, threshold=NULL, n.perm=1, alpha=0.05, step=0, method = "hk", saved.scores = NULL, rev.method = c("node.nbhd","node","nbhd", "node.edge", "single"), verbose = FALSE, max.parents.partition = 3, filename = "tmp.RData", ...)
Arguments

cross An object of class cross. It contains genotypes and phenotypes. See read.cross.

B A matrix if only one type of biological knowledge is incorporated or a list of matrices for biological knowledge. If B = NULL, biological knowledge is not used in the causal network inference. Otherwise, each matrix in the list encodes one type of biological knowledge. Each matrix is a square matrix where (i,j)-th element is the probability of edge presence (i -> j) supported by the corresponding type of biological knowledge. Elements should be between 0 and 1.

nogenotype If genotype information is used, nogenotype = FALSE. If genotype information is not used, nogenotype = TRUE.

pheno.col Phenotype identifiers from cross object to be included in the causal network inference. May be numeric, logical or character. If pheno.col=NULL, all the phenotypes in the cross object will be used.

addcov Additive covariates for each phenotype. If there is no additive covariate, addcov = NULL.

intcov Interactive covariates for each phenotype. If there is no interactive covariate, intcov = NULL.

nSamples Number of MCMC samples to record.

thinning Every thinning is recorded after burnin period.

burnin Ratio of initial MCMC iterations to drop.

random.seed Random seed.

max.parents Maximum number of parents allowed in the network structure.

M0 Initial phenotype network structure satisfying init.edges and max.parents in the MCMC run. It is a ‘number of phenotypes’ by ‘number of phenotypes’ matrix.

init.edges Number of edges in the initial network.

beta0 Initial beta value in the MCMC run. beta is a hyperparameter in the prior distribution for phenotype networks which controls the contribution of biological knowledge. The prior distribution for phenotype networks is defined to be \( P(G_Y|B, beta) \propto e^{\beta G_Y - B} \).

lambda The scale parameter in the hyperprior distribution for beta. It sets the hyperprior for beta to be \( P(beta|lambda) \propto lambda e^{-lambda * beta} \).

threshold A scalar or a vector of thresholds for QTL identification. If it is a scalar, it will be replicated as a vector of length(pheno.col). Otherwise, it should be the same length as the number of phenotypes. The same threshold is used for autosomes and X-chromosome for this version. If genotypes are not included in network inference (nogenotype=TRUE), threshold will be a vector of NA.

n.perm If threshold is NULL, a permutation test to get a vector of thresholds for QTL identification is performed. n.perm is the number of permutations to perform to get an alpha-level threshold for QTL detection for each phenotype. If there is addcov or intcov, the permutation test takes into account of it.

alpha Significance cut-off for thresholds for QTL identification.

step Genotype probabilities are calculated at the maximum step (cM) between pseudomarkers. When step = 0, genotype probabilities are calculated only at the marker positions.

method A method used for QTL interval mapping. A default is Haley-Knott regression. See scanone.
saved.scores Pre-computed scores with all possible parent configurations up to max.parents. If it is NULL, it will be calculated inside the function mcmc.qtlnet.prior.

rev.method A proposal method for a new network structure. If rev.method = "single", a proposal is among ‘add’, ‘delete’, and ‘reverse’ of an edge. If rev.method = "nbhd", it is an updated version of rev.method="single" in reversing an edge as described in Grzegorczyk and Husmeier (2008). If rev.method="node.edge", it is an extension of the method in Grzegorczyk and Husmeier (2008). In proportional to the score between a parent set and a node, it randomly selects 1) a node and its new parent set, 2) an existing edge and new parent sets for its two nodes, without making an edge between them, or 3) an existing edge and new parent sets for its two nodes, reversing the edge between them. If rev.method="node", it randomly selects a node and moves to the new structure by adding or reversing [x -> node], or deleting [x -> node] or [node -> x]. If rev.method="node.nbhd", it is a mix of ‘node’ and ‘nbhd’ methods.

verbose If TRUE or 1, print iteration numbers and permutations. If verbose=2 or 4, print MCMC moves. If verbose=3 or 4, plot BIC.

max.parents.partition Maximum number of parents to be used to calculate the partition function (normalizing term in the prior probability of phenotype network structures).

filename The name of a file to save the output.

... Additional arguments.

Details

It runs an MCMC to infer a causal network of genotypes and phenotypes, incorporating biological knowledge B.

Arguments related with using data: nogenotype, B. This function can run with four conditions as described in the example below – with or without genotypes, with or without biological knowledge.

Arguments related with initial points in MCMC: m0, init.edges, beta0.

Arguments related with MCMC running length: nSamples, thinning, burnin.

Arguments related with QTL mapping when using genotypes: threshold, n.perm, alpha, step, method.

Arguments to constrain the network structures due to the super-exponential size of network structure space: max.parents.

Arguments for network structure proposal: rev.method. There are five ways to propose a new network structure.

Value

List of class qtlnet with additional values. See mcmc.qtlnet. Additional values are

mav Model average of M for nSamples of MCMC samples

cont.accept Count of acceptance or rejection for new structure proposals (addition, deletion, reverse)

post.beta Posterior beta

post.partition Partition function calculation (normalizing term in the prior probability of phenotype networks) specified by beta value

beta.freq.accept Acceptance frequency of beta
mcmc.qtlnet.prior

post.prob         Proportional to log of posterior probability

Additional attributes from input

B               A list of matrices for biological knowledge
nogenotype       Indicator whether genotypes are used
lambda          The scale parameter in the hyperprior distribution for beta.

Author(s)

Jee Young Moon and Brian S. Yandell
Maintainer: Jee Young Moon <jeeyoung.moon@gmail.com>

References


See Also

For the class of qtlnet object: mcmc.qtlnet.
For the cross object and QTL mapping: read.cross, scanone.
For the prior probability of a phenotype network: score.prior, partition.eff, all.config.
For the prior probability of beta: score.beta.
For the BIC score of a network: score.model.qtlnet.prior.
For the proposal of new network structures by rev.method=node.nbhd: propose.new.node.structure, node.nbhd.size.
For the generation of simulated knowledge: sim.knowledge.

Examples

```r
## A network of 5 phenotypes with following causal relations:
## 1->2, 1->3, 1->4, 2->5, 3->4, 3->5, 4->5.
## i->j is denoted by A[i,j]=1; otherwise, A[i,j]=0
pheno.col=c(1,2,3,4,5)
A=matrix(0, nrow=5, ncol=5)
A[1,2]=1; A[1,3]=1; A[1,4]=1;
connected.Y <- c(1,2,1,3,1,4,
```
2, 5, 
3, 4, 3, 5, 
4, 5)
c = matrix(c(1, 2, 3, 4, 5), ncol=2, byrow=TRUE)
for (i in seq(nrow(c))) {
  c[i, i] <- 1
}

## Number of biological knowledge types
B.N <- 1
## Accuracy of biological knowledge
delta <- c(0, 2)
## Scanone threshold for QTL detection
threshold = 3.83

## MCMC simulation parameters
# final number of MCMC samples
nSamples = 1000
# MCMC samples will
thinning = 10
# Maximum number of parents
max.parents = 3

## Simulate a cross object
cross <- sim.example1.cross(342)
cross <- calc.genoprob(cross, step=0)

## Simulate a list of knowledge matrices
B <- sim.knowledge(B.N, delta, A)

## Save scanone result condition on the possible parental configurations
saved.scores <- bic.qtlnet(cross, pheno.col=pheno.col, threshold=threshold, max.parents=max.parents)
saved.scores <- bic.join(cross, pheno.col, list(saved.scores), max.parents=max.parents)

## Case 1: Run MCMC to infer a causal network of genotypes and phenotypes
## incorporating biological knowledge
outBG <- mcmc.qtlnet.prior(cross, B=B, nogenotype=FALSE, pheno.col=pheno.col, 
nSamples=nSamples, thinning=thinning, random.seed=243, 
init.edges = 0, threshold=threshold, saved.scores=saved.scores, 
rev.method=nbhd)

## Case 2: Run MCMC to infer a causal network of genotypes and phenotypes
## (no biological knowledge): Chaibub Neto, E. et al. (2010)
outG <- mcmc.qtlnet.prior(cross, B=NULL, nogenotype=FALSE, pheno.col=pheno.col, 
nSamples=nSamples, thinning=thinning, random.seed=472, 
init.edges = 0, threshold=threshold, saved.scores=saved.scores, 
rev.method=nbhd)

## Case 3: Run MCMC to infer a causal network of phenotypes
outB <- mcmc.qtlnet.prior(cross, B=B, nogenotype=TRUE, pheno.col=pheno.col, 
nSamples=nSamples, thinning=thinning, random.seed=6821, 
init.edges = 0, threshold=NULL, saved.scores=NULL, 
rev.method=nbhd)

## Case 4: Run MCMC to infer a causal network of phenotypes
node.nbhd.size

Number of moves around a selected node in a network

Description

Calculates possible number of structures from the current network structure after selecting a node and by performing one of the following moves: addition [x -> node], deletion [x -> node], deleteion [node -> x], or reversion of [x -> node], satisfying max.parents and acyclicity.

Usage
	node.nbhd.size(M, node, max.parents = 3)

Arguments

- **M**: Adjacency matrix of a phenotype network structure.
- **node**: A selected node index.
- **max.parents**: Maximum number of parents.

Value

A list with components

- **nbhd.size**: n.deletions + n.additions + n.reversions
- **n.deletions**: Possible number of structures by a deletion of an edge [x -> node] or [node -> x]
- **n.additions**: Possible number of structures by an addition of an edge [x -> node]
- **n.reversions**: Possible number of structures by a reversion of an edge [x -> node]
- **moves**: A (nbhd.size * 3) data.frame of possible moves. Columns are 'from', 'to', and 'move'. Each row represents the 'move' of an edge ['from' -> 'to'] in the current network.

Note

The nbhd.size component is greater than 0 because 1) if edge addition is not possible, we can always delete an edge [x->node] or [node->x], 2) if there is no edge to delete, in other words, ther is no edge connected with the node, we can always add an edge [x->node] satisfying acyclicity and max.parents.

Author(s)

Jee Young Moon and Brian S. Yandell
partition.eff

See Also

propose.new.node.structure

Examples

```r
mat <- diag(3)
mat[1,2] <- mat[2,3] <- 1
diag(mat) <- 0

node.nbhd.size(mat, 1, 2)
node.nbhd.size(mat, 2, 2)
node.nbhd.size(mat, 3, 2)
```

partition.eff

*Calculates the log of partition function*

Description

It calculates the log of partition function in an approximate way (normalizing term in prior probability of phenotype networks) specified by beta. The partition function is defined to be \( \sum_{\gamma} \exp(-\sum_j \beta_j * |B[j][,i] - G_Y[,i]|) \). Instead, for each child node i, it first calculates the \( \sum_j |B[j][,i] - G_Y[,i]| \) over all parent configurations: \( \sum_{k \text{nodes'parent configuration}} \exp(-\sum_j \beta_j * [(1 - B[j][k,i]) + (B[j][k,i] - k,i)]) \). Then, the product of the values across nodes are used as the partition function. This function partition.eff returns the log of the calculated partition function.

Usage

```r
partition.eff(beta, n.pheno, B, count.parents.config, e.value)
```

Arguments

- `beta`: A vector of beta values. beta is the same length of list B.
- `n.pheno`: Number of phenotypes.
- `B`: A list of biological knowledge matrices.
- `count.parents.config`: Total number of possible parents’ configurations up to max.parents.partition parents for a node. See all.config.
- `e.value`: An array of (n.pheno, count.parents.config, number of types of knowledge). `e.value[i,k,j]` is `|B[j][k,i] - G_Y[,i]|` with `G_Y` having the k-th parent configuration for the child node i, and j-th knowledge. See all.config.

Value

A numeric value, log of the partition function.

Author(s)

Jee Young Moon and Brian S. Yandell
References


See Also

`all.config, score.prior`

Examples

```r
B <- runif(25, 0, 1)
B <- matrix(B, ncol=5)
diag(B) <- 0
B <- list(B)

ex.config <- all.config(n.pheno=5, max.parents.partition=3, B)
partition.eff(beta=1.2, n.pheno=5, B, ex.config[[count.parents.config]], ex.config[[e.value]])
```

```
propose.new.beta Propose a new beta

Description

Propose a new beta. First, it samples `beta.new` from $U[\beta_{old}-1, \beta_{old}+1]$. Second, if `beta.new` is negative, use $-1 \times beta.new$. This proposal satisfies the following ratio: $P(\beta_{new} | \beta_{old}) / P(\beta_{old} | \beta_{new}) = 1$.

Usage

`propose.new.beta(beta.old, lambda)`

Arguments

- `beta.old` Old beta
- `lambda` lambda is not used.

Value

A positive value for a new beta.

Author(s)

Jee Young Moon and Brian S. Yandell

Examples

`propose.new.beta(0.5, 1)`
propose.new.node.structure

Propose a new network with edge addition, deletion or reversion

Description

After selecting a node randomly, node.nbhd.size finds possible single moves by adding or reversing \([x \rightarrow \text{node}]\), or by deleting \([x \rightarrow \text{node}]\) or \([\text{node} \rightarrow x]\). This function (propose.new.node.structure) randomly selects a move from the possible moves.

Usage

propose.new.node.structure(M, max.parents = 3, saved.scores, node, neighborM, rev.method = c("node","node.nbhd"), verbose = FALSE)

Arguments

M
An adjacency matrix of a phenotype network.
max.parents
Maximum number of parents.
saved.scores
Pre-computed scores with all possible parent configuration with max.parents.
node
The index of a selected node.
neighborM
Information of possible moves around the selected node. It is a result from node.nbhd.size.
rev.method
If rev.method="node", the move is among the possible single moves from node.nbhd.size. If rev.method="node.nbhd", the move for a reversion is enhanced with rev.method=nbhd as in Grzegorczyk, M. and Husmeier, D. (2008).
verbose
Print out the move.

Value

A list with components

M
An adjacency matrix of a new network
rev.ratio
When the move is 'reverse-nbhd', it is a ratio of edge reversion probabilities to the old network and to the new network. Otherwise, it is 1.
move
A move among 'add', 'delete', 'reverse', and 'reverse-nbhd'.
ne.new
When the move is 'reverse-nbhd', it is 0 (it is not considered). Otherwise, it is a possible number of structures from the new network by the method rev.method=node.

Note

For rev.method=node, after selecting a node randomly with probability \(1/(\text{number of nodes})\), the move is \(1/(\text{neighborM}\$\text{nbhd.size})\). Depending on the move type, the backward move probability from the new network to the old network is as follows: 1) move was add \([x \rightarrow \text{node}]\): backward moves are selecting the node and deleting \([\text{node} \rightarrow x]\) or selecting x and deleting \([\text{node} \rightarrow x]\), 2) move was delete \([a \rightarrow b]\) where \((a,b)=(\text{node},x)\) or \((x,\text{node})\): selecting node a and adding \([a \rightarrow b]\), 3) move was reverse \([x \rightarrow \text{node}]\): backward move is selecting x and reversing \([\text{node} \rightarrow x]\).
score.beta

Log of the prior probability of beta

Description

The hyperprior probability of beta is defined to be \( P(\beta|\lambda) = \lambda e^\left(-1 \ast \lambda \ast \beta\right) \) where \( \lambda \) is a fixed value. This function score.beta returns \( -\lambda \ast \beta \) without a constant term as a score for the prior of beta.

Usage

score.beta(beta, lambda)

Arguments

beta : Beta.
lambda : lambda.

Value

It returns a numeric value : \( -1 \ast \lambda \ast \beta \).

Author(s)

Jee Young Moon and Brian S. Yandell
Calculates the BIC score of a network

Description

It calculates the BIC score of a network. QTL mapping can be chosen to be done or not.

Usage

score.model.qtlnet.prior(M, saved.scores, cross, addcov, intcov, threshold, nogenotype, verbose = TRUE, ...)

Arguments

M An adjacency matrix of a phenotype network.
saved.scores Pre-computed BIC score for each child and parent set combination.
cross Cross object.
addcov Additive covariates.
intcov Interactive covariates.
threshold Threshold for QTL mapping.
nogenotype Whether QTL mapping is done or not.
verbose Verbose.
...
Additional arguments.

Value

A list with components

model.score BIC score of the network
update.scores Updated scores. A list with components 'code', 'pheno.col', and 'bic'.
model.name Code name for the network

Examples

```r
## M
pheno.col=c(1,2,3,4,5)
M=matrix(0, nrow=5, ncol=5)
connected.Y <- c(1,2, 1,3, 1,4,
                 2,5,
                 3,4, 3,5,
                 4,5)
connected.Y <- matrix(connected.Y, ncol=2, byrow=TRUE)
for (i in seq(nrow(connected.Y))){
  M[connected.Y[i,1], connected.Y[i,2]] <-1
}
## Simulate a cross object
cross <- sim.example1.cross(342)
```
cross <- calc.genoprob(cross, step=0)
threshold <- 3.8
addcov=NULL; intcov=NULL; max.parents=3;

## Pre-computed score with QTL mapping
saved.scores.QTL <- bic.qtlnet(cross, pheno.col=pheno.col, threshold=threshold, addcov=addcov, intcov=intcov, max.parents=max.parents)
saved.scores.QTL <- bic.join(cross,pheno.col, list(saved.scores.QTL), max.parents=max.parents)

## Pre-computed score without QTL mapping
saved.scores.noQTL <- qtlnet.prior:::bic.qtlnet.pheno(cross, pheno.col=pheno.col, threshold=threshold, addcov=addcov, intcov=intcov, max.parents=max.parents)
saved.scores.noQTL <- bic.join(cross,pheno.col, list(saved.scores.noQTL), max.parents=max.parents)

## Score of a network
score.model.qtlnet.prior(M, saved.scores=saved.scores.QTL, cross=cross, addcov=NULL, intcov=NULL,
threshold=3.8, nogenotype=FALSE, verbose=TRUE)
score.model.qtlnet.prior(M, saved.scores=saved.scores.noQTL, cross=cross, addcov=NULL, intcov=NULL,
threshold=3.8, nogenotype=TRUE, verbose=TRUE)

---

score.prior

Log of the numerator in prior probability of phenotype networks

Description

The prior probability of phenotype networks is defined to be $P(G_Y|B, \beta) \propto \exp(-\sum_j \beta[j] \times |B[[j]] - M|)$. The normalizing term (denominator) for this probability is called a partition function. This function score.prior calculates the log of the numerator and partition.eff approximately calculates the log of the denominator.

Usage

```r
score.prior(M, beta, B)
```

Arguments

- **M**
  - Adjacency matrix.
- **beta**
  - A vector of beta values. beta[j] corresponds to B[[j]].
- **B**
  - A list of matrices to encode biological knowledge. Each matrix (B[[j]]) corresponds to one type of biological knowledge.

Details

If beta is close to 0, the prior is close to a uniform distribution and hence the contribution of biological knowledge is negligible. If \( \beta \rightarrow \text{infinity} \), the prior puts the most of probability on the structure closest to B.

Value

It returns a numeric value: \(-1 \times \sum_j \beta[j] \times |M - B[[j]]|\).
Author(s)
Jee Young Moon and Brian S. Yandell

References

See Also
partition.eff

Examples
```r
M.new <- diag(5)
M.new[1,2] <- M.new[2,3] <- 1
diag(M.new) <- 1

beta <- 1.2

B <- M.new
B[1,2] <- 0.8
B[2,3] <- 0.7
B <- list(B)

M <- diag(5)
diag(M) <- 0

score.prior(M.new, beta, B)
score.prior(M, beta, B)
```

sim.example1.cross
Generate an example of cross of 500 individuals with 5 phenotypes

Description
It generates an F2 cross object consisting of genetic information and 5 phenotypes for 500 individuals. See the details for the detailed information.

Usage
```r
sim.example1.cross(seed = 1234)
```

Arguments
- `seed` Random seed number
Details

sim.example1.cross generates an F2 cross object consisting of genetic information and 5 phenotypes for 500 individuals.

The underlying causal network between QTL and phenotypes is as following: Q1 -> Y1, (Q2, Y1) -> Y2, Y1 -> Y3, (Q4, Y1, Y3) -> Y4, (Q5, Y2, Y3, Y4) -> Y5. Q_t is a QTL in the middle of chromosome_t.

First, the genetic map is generated for 5 autosomal chromosomes of 100 cM with 10 unequally spaced markers in each chromosome.

Second, genetic additive effects, dominance effects, and partial regression coefficients are randomly generated. Genetic additive effects are sampled from U[0, 0.5], genetic dominance effects are sampled from U[0, 0.25], and partial regression coefficients are sampled from U[-0.5, 0.5].

Third, phenotypes are generated by adding a Gaussian noise of variance 1 to the calculated phenotypic value (immediate QTL effect + immediate upstream phenotypic effect).

Value

An object of class cross, which is a list of three components:

genov A list of genetic information for each chromosome
pheno A matrix of QTL genotypes
qtlgeno A data.frame of phenotypes

See Also

sim.cross, sim.map, read.cross

Examples

sim.example1.cross(356)

---

sim.knowledge

Simulate a knowledge matrix

Description

Simulate a knowledge matrix

Usage

sim.knowledge(B.N = 1, delta = 0.5, A)

Arguments

B.N Number of types of biological knowledge
delta A vector of accuracy for each knowledge type. Delta value is between -0.5 and 0.5. The higher the delta is, the more accurate the knowledge will be.
A True adjacency matrix of the network
**Details**

Each element corresponding to 1 in the true adjacency matrix \( A \) is simulated from a truncated normal distribution between 0 and 1 with mean \( 0.5 + \delta \) and sigma=0.1. Each element corresponding to 0 in the true adjacency matrix \( A \) is simulated from a truncated normal distribution between 0 and 1 with mean \( 0.5 - \delta \) and sigma=0.1.

**Value**

A list with \( B.N \) components and each component is a matrix of the same size as \( A \). Each component corresponds to one type of biological knowledge.

**Author(s)**

Jee Young Moon and Brian S. Yandell

**Examples**

```r
A <- diag(3)
diag(A) <- 0

sim.knowledge(B.N=2, delta=0.3, A)
sim.knowledge(B.N=2, delta=c(0.3, -0.3), A)
```
Index

*Top Topic package
  qtlnet.prior-package, 1
*Top Topic qtlnet.prior
  mcmc.qtlnet.prior, 3

all.config, 2, 6, 9, 10
mcmc.qtlnet, 5, 6
mcmc.qtlnet.prior, 2, 3

node.nbhd.size, 6, 8, 11, 12
partition.eff, 3, 6, 9, 14, 15
propose.new.beta, 10
propose.new.node.structure, 6, 9, 11, 11

qtlnet.prior (qtlnet.prior-package), 1
qtlnet.prior-package, 1

read.cross, 4, 6, 16
scanone, 4, 6
score.beta, 6, 12
score.model.qtlnet.prior, 6, 13
score.prior, 6, 10, 14
sim.cross, 16
sim.example1.cross, 15
sim.knowledge, 6, 16
sim.map, 16