Package 'bnClustOmics'

August 5, 2022

Title Bayesian Network-Based Clustering of Multi-Omics Data

Version 1.1.1

Description Unsupervised Bayesian network-based clustering of multi-omics data. Both binary and continuous data types are allowed as inputs. The package serves a dual purpose: it clusters (patient) samples and learns the multi-omics networks that characterize discovered clusters. Prior network knowledge (e.g., public interaction databases) can be included via black-listing and penalization matrices. For clustering, the EM algorithm is employed. For structure search at the M-step, the Bayesian approach is used. The output includes membership assignments of samples, cluster-specific MAP networks, and posterior probabilities of all edges in the discovered networks. In addition to likelihood, AIC and BIC scores are returned. They can be used for choosing the number of clusters.
References:
P. Suter et al. (2021) <doi:10.1101/2021.12.16.473083>,
J. Kuipers and P. Suter and G. Moffa (2022) <doi:10.1080/10618600.2021.2020127>,
J. Kuipers et al. (2018) <doi:10.1038/s41467-018-06867-x>.

License GPL-3

Imports BiDAG, mclust, clue, stats, RBGL, graph, gRbase, RColorBrewer, graphics, plotrix

Encoding UTF-8

LazyData true

RoxygenNote 7.2.0

NeedsCompilation no

Author Polina Suter [aut, cre], Jack Kuipers [aut]

Maintainer Polina Suter <polina.suter@gmail.com>

Depends R (>= 3.5.0)

Repository CRAN

Date/Publication 2022-08-05 14:50:06 UTC

R topics documented:

`adjustMixedDir` *Adjusting the PDAG matrix to model constraints This function can be used to adjust the adjacency matrix to model constraints, such as blacklist and background nodes*

Description

Adjusting the PDAG matrix to model constraints This function can be used to adjust the adjacency matrix to model constraints, such as blacklist and background nodes

Usage

```
adjustMixedDir(adj, bnnames, blacklist)
```

Arguments

adj	adjacency matrix representing a graph or posterior probabilities of all its edges
bnnames	object of class bnNames
blacklist	a square matrix with same dimensions as adj representing edges prohibited by the model

Value

returns a matrix where entries prohibited by the model or blacklist are 0 and equal to correspondnding values of adj otherwise

---

annotateEdges *Annotating edges from discovered networks*

Description

This function makes a data frame which contains all pairs of nodes connected in cluster-specific networks

Usage

```
annotateEdges(
  bnres,
  bnnames,
  sump = 1.2,
  minp = 0.5,
  minkp = 0.9,
  maxkp = NULL,
  dblist = NULL
)
```

Arguments

bnres	an object of class 'bnclustOmics'; see bnclustOmics
bnnames	an object of class 'bnInfo'; see bnInfo
sump	threshold for the sum of posterior probabilities in all discovered networks
minp	threshold for the minimum posterior probability in at least one network, when the sum of posterioirs is bigger than sump
minkp	threshold for the minimum posterior probability in at least one network, when the sum of posterioirs is less than sump
maxkp	(optional) threshold for the maximum posterior probability in at least one network; used to esclude cluster specific edges from the edges with high sum of posterioirs (>sump)
dblist	a list of known interactions, discovered edges will be annotated is the edge is present in this list; two columns must be present 'gene1' and 'gene2'

Value

returns a data frame where each filteres interaction is annotated with IDs of omics variables, omics types, posterior probabilities of the interaction in the discovred clusters and a flag indication if the interaction could be found in the interaction data base

Examples

```
bnnames<-bnInfo(simdata,c("b","c"),c("M","T"))
intlist<-annotateEdges(bnres3,bnnames,dblist=simint)
length(which(intlist$db))
```

`blInit` *Initializing blacklist*

Description

This function can be used to initialize a blacklist matrix for bnclustOmics clustering

Usage

```
blInit(
  bnnames,
  bldiag = TRUE,
  intra = NULL,
  interXX = list(from = NULL, to = NULL),
  interXY = list(from = NULL, to = NULL)
)
```

Arguments

`bnnames`	object of class bnInfo; see bnInfo
`bldiag`	logical, defines if diagonal should be blacklisted, TRUE by default
`intra`	(optional) a vector of characters defining omic types for which intra-type edges will be blacklisted
`interXX`	(optional) a list containing two vectors of characters defining omic types between which same gene (X.type.from -> X.type.to) edges will be blacklisted
`interXY`	(optional) a list containing two vectors of characters defining omic types between which different gene edges (X.type.from -> Y.type.to) will be blacklisted

Value

returns a binary matrix where 1 defines prohibited edges and 0 defines allowed edges

Author(s)

Polina Suter

blUpdate *Updating blacklist*

Description

This function can be used to update a blacklist matrix by blacklisting an edge between a pair of variables

Usage

```
blUpdate(blacklist, node1, node2)
```

Arguments

blacklist	object of class 'blacklist'
node1	name of omic variable from which the edge is prohibited
node2	name of omic variable to which the edge is prohibited

Value

returns a binary matrix where 1 defines prohibited interactions and 0 defines allowed interactions

Author(s)

Polina Suter

bnclustNetworks *Deriving consensus networks based on posterior probabilities of mixture model*

Description

This function derives consensus models of networks representing all clusters based on several threshold for posterior probabilities of individual edges.

Usage

```
bnclustNetworks(
  bnres,
  bnnames,
  sump = 1.2,
  minp = 0.5,
  minkp = 0.9,
  maxkp = NULL
)
```

Arguments

bnres	an object of class 'bnclustOmics'; see bnclustOmics
bnnames	an object of class 'bnInfo'; see bnInfo
sump	threshold for the sum of posterior probabilities in all discovered networks
minp	threshold for the minimum posterior probability in at least one network, when the sum of posterioirs is bigger than sump
minkp	threshold for the minimum posterior probability in at least one network, when the sum of posterioirs is less than sump
maxkp	(optional) threshold for the maximum posterior probability in at least one network; used to esclude cluster specific edges from the edges with high sum of posterioirs (>sump)

Value

returns a list of adjacency matrices, one for each cluster representing consensus models

Examples

```
bnnames<-bnInfo(simdata,c("b","c"),c("M","T"))
intlist<-bnclustNetworks(bnres3,bnnames)
```

---

bnclustOmics *Bayesian network based clustering of multi-omics data*

Description

Bayesian network-based clustering of multi-omics data. This function implements network-based clustering for multiomics data. The mandatory input is a list of matrices consisting from binary, ordinal or continuous variables. Each matrix corresponds to one omics type. At least one matrix with continuous variables must be present. Optional output includes the prior information about interactions between genes and gene products. This can be passed via parameters blacklist and edgepmat. Interactions in blacklist are excluded from the search space. Edgepmat imposes a graphical prior which penalizes certain interactions by a certain penalization factor. The output includes cluster assigments and MAP directed acycluc graphs (DAGs) representing discovered clusters. Optionally, the output may include posterior probabilities of all edges in the discovered graphs.

Usage

```
bnclustOmics(
  omicdata,
  bnnames,
  blacklist = NULL,
  edgepmat = NULL,
  kclust = 2,
  chixi = 0,
```

```
  seed = 100,
  err = 1e-06,
  maxEM = 10,
  hardlim = 6,
  deltahl = 5,
  nit = 5,
  epmatrix = TRUE,
  plus1it = 4,
  startpoint = "mclustPCA",
  baseprob = 0.4,
  commonspace = TRUE,
  verbose = TRUE
)
```

Arguments

omicdata	a list of matrices corresponding to omics types. For example, "M" (mutations), "CN" (copy numbers), "T" (transcriptome), "P" (proteome) and "PP" (phosphoproteome); at least one continuous type must be present
bnnames	object of class 'bnInfo'; see constructor function bnInfo
blacklist	adjacency matrix containing information about which edges will be blacklisted in structure search
edgepmat	penalization matrix of the edges in structure learning
kclust	the number of clusters (mixture components)
chixi	prior pseudocounts used for computing parameters for binary nodes
seed	integer number set for reproducibility
err	convergence criteria
maxEM	maximum number of outer EM iterations (structural search)
hardlim	maximum number of parents per node when learning networks
deltahl	additional number of parents when sampling from the common search space
nit	number of internal iteration (of parameter estimation) in the EM
epmatrix	(logical) indicates if the matrices containing posterior probabilities of single edges are be returned
plus1it	maximum number of search space expansion iterations when performing structure search
startpoint	defines which algorithm is used to define starting cluster memberships: possible values "random", "mclustPCA" and "mclust"
baseprob	defines the base probability of cluster membership when "mclustPCA" or "mclust" used as starting point
commonspace	(logical) defines if the sampling has to be performed from the common search space
verbose	defines if the output messages should be printed

Value

object of class 'bnclustOmics' containing the results of Bayesian-network based clustering: cluster assignments, networks representing the clusters

Author(s)

Polina Suter, Jack Kuipers

Examples

```
bnnames<-bnInfo(simdata,c("b","c"),c("M","T"))

fit<-bnclustOmics(simdata,bnnames,maxEM=4, kclust=2, startpoint = "mclustPCA")
clusters(fit)
checkmembership(clusters(fit),simclusters)
```

---

bnInfo *Constructing object of class bnInfo*

Description

This function constructs an object of class bnInfo which is needed for Bayesian network based clustering; see function bnclustOmics. In this object the names and types of omics data are stored as well as maappings containing the correspondance between gene names in each omic type and gene names used in blacklist and edge penalization matrices in the clustering step. These mappings are helpful for constructing such matrices. For example, transcriptome data often includes ensemble IDs and mutation data includes gene names. If we want to penalize all interactions which are not found in a specific interactions database, we need to pass an interaction list this list usually includes gene names and not ensemble IDs. Mappings pass the information needed to assign the edges between any IDs of gene X the specified penalization factor. If some omics types already have the same ID as in interaction list, corresponding mappings can be skipped.

Usage

```
bnInfo(omicdata, types, omics, mappings = NULL, attachtype = FALSE)
```

Arguments

omicdata	a list of matrices containing data, rows are observations, columns are variables (the order should be as following binary->ordinal->continuous)
types	a vector of characters equal in length to the number of provided omic matrices, "b" binary, "o" ordinal, "c" continuous
omics	a vector of omic names, must be the same as names of elements in omicdata, otherwise names of omicdata will be overwritten

mappings	mappings containing a gene symbol for each omic type, rownames have to contain column names of the parameter 'omicdata'; column "gene" must be present; if NULL for a certain omic type, than gene name will be taken from the column name of the corresponding matrix.
attachtype	when TRUE .O will be attached to each variable name, where O is omic name (see parameter 'omics'); when FALSE (default) .O is only attached to duplicated names

Value

an object of class bnInfo

Examples

```
#with mappings
bnnames<-bnInfo(toydata,c("b","o","c","c","c"),c("M","CN","T","P","PP"),mappings)
#no mappings
bnnames<-bnInfo(simdata,c("b","c"),c("M","T"))
```

---

bnres2 *bnres2*

Description

An object of class 'bnclustOmics' containing the results of one run of the function 'bnclustOmics' with the parameter k=2. The object contains membership assignments, estimated MAP graphs representing clusters as well as posterior probabilities of all edges for each cluster.

Usage

```
bnres2
```

Format

An object of class 'bnclustOmics'

---

bnres3 *bnres3*

Description

An object of class 'bnclustOmics' containing the results of one run of the function 'bnclustOmics' with the parameter k=3. The object contains membership assignments, estimated MAP graphs representing clusters as well as posterior probabilities of all edges for each cluster.

Usage

```
bnres3
```

Format

An object of class 'bnclustOmics'

---

bnres4 *bnres3*

Description

An object of class 'bnclustOmics' containing the results of one run of the function 'bnclustOmics' with the parameter k=4. The object contains membership assignments, estimated MAP graphs representing clusters as well as posterior probabilities of all edges for each cluster.

Usage

```
bnres4
```

Format

An object of class 'bnclustOmics'

---

checkmembership *Comparing estimated and ground truth membership*

Description

This function compares similarity between two clusterings.

Usage

```
checkmembership(estmemb, truememb)
```

Arguments

estmemb	estimated labels
truememb	ground truth labels

Value

a list containing different measures of similarity between two different clusterings, including accuracy, adjusted Rand index and precision

chooseK *Choosing the number of clusters*

Description

This function can be used for choosing the optimal number of clusters using AIC or BIC scores.

Usage

```
chooseK(bnlist, fun = c("AIC", "BIC", "likel"))
```

Arguments

bnlist	list of objects of class 'bnclustOmics'
fun	score function for choosing the optimal number of clusters; available options are 'AIC' or 'BIC'

Value

a list consisting of a vector of scores extracted from each object of class bnclustOmics and the optimal k

Examples

```
bnlist<-list()

#bnlist[[k]]<-bnclustOmics(simdata,bnnames,maxEM=4, kclust=k,startpoint = "mclustPCA")
bnlist[[2]]<-bnres2
bnlist[[3]]<-bnres3
bnlist[[4]]<-bnres4

chooseK(bnlist,fun="BIC")
chooseK(bnlist,fun="AIC")
```

clustDBN *DBN-based clustering*

Description

This function can be used for DBN-based clustering. It is the same function as bnclustOmics, but it also works for time series data.

Usage

```
clustDBN(
  dbndata,
  staticnodes = 0,
  blacklist = NULL,
  edgepmat = NULL,
  kclust = 2,
  chixi = 0.5,
  seed = 100,
  err = 1e-06,
  maxEM = 10,
  hardlim = 6,
  deltahl = 2,
  nit = 5,
  epmatrix = TRUE,
  plus1it = 4,
  nruns = 1,
  startpoint = "mclustPCA",
  baseprob = 0.4,
  commonspace = TRUE,
  verbose = TRUE,
  samestruct = TRUE,
  pickmax = TRUE
)
```

Arguments

dbndata	data matrix; rows are observations, columns are variables; static nodes have to be in the first column of the data
staticnodes	(integer) number of static nodes in a DBN
blacklist	adjacency matrix containing information about which edges will be blacklisted in structure search
edgepmat	penalization matrix of the edges in structure learning
kclust	the number of clusters (mixture components)
chixi	prior pseudocounts used for computing parameters for binary nodes
seed	integer number set for reproducibility
err	convergence criteria
maxEM	maximum number of EM iterations (structural)
hardlim	maximum number of parents per node when learning networks
deltahl	additional number of parents when sampling from the common search space
nit	number of internal iteration in structural EM
epmatrix	(logical) indicates if the matrices containing posterioir probabilities of single edges should be returned
plus1it	maximum number of search space expansion iterations when performing structure search

nruns	number of runs of the EM algorithm
startpoint	defines which algorithm is used to define starting cluster memberships: possible values "random", "mclustPCA" and "mclust"
baseprob	defines the base probability of cluster membership when "mclustPCA" or "mclust" used as starting point
commonspace	(logical) defines if the sampling has to be performed from the common search space
verbose	defines if the output messages should be printed
samestruct	(logical) defines if initial and intrinsic part of transition structures should be the same
pickmax	(logical) if TRUE only maximum EM run is returned

Value

object of class 'bnclustOmics' containing the results of Bayesian-network based clustering: cluster assignments, networks representing the clusters

Author(s)

Polina Suter

`clusters` *Extracting cluster memberships*

Description

This function extracts a vector with MAP cluster memberships assignments from the 'bnclustOmics' object

Usage

```
clusters(x, consensus = FALSE)
```

Arguments

x	object of class 'bnclustOmics'
consensus	logical, indicates if consensus clusters will be extracted; FALSE by default

Value

a vector of length of the number of observations corresponding to cluster assignments obtained by bnclustOmics

Examples

```
clusters(bnres3)
```

dags *Extracting edge posterior probabilities*

Description

This function extracts a list of matrices containing posterior probabilities of all edges in the graphs discovered by bnclustOmics when the parameter 'epmatrix' was set to TRUE

Usage

```
dags(x)
```

Arguments

x	object of class 'bnclustOmics'

Value

a list of matrices containing posterior probabilities of all edges in the graphs discovered by bnclustOmics when the parameter 'epmatrix' was set to TRUE

Examples

```
DAGs<-dags(bnres3)
```

getModels *Deriving consensus graphs*

Description

When the parameter 'epmatrix' is set to TRUE, the object of class 'bnclustOmics' includes postrior probabilitis of all edges in the discovered graphs. This function can be used to derive a consensus graph representing discovered clusters according to a specified posterior probability threshold. Only edges with posteriors above the threshold will be included in the resulting consensus models.

Usage

```
getModels(bnres, p)
```

Arguments

bnres	object of class 'bnclustOmics'
p	posterior probability threshold

Value

a list of adjacency matrices corresponding to consensus graphs representing discovered clusters

Author(s)

Polina Suter

Examples

```
MAPmod<-dags(bnres3)
CONSmod1<-getModels(bnres3,p=0.5)
CONSmod2<-getModels(bnres3,p=0.9)
library(BiDAG)
compareDAGs(MAPmod[[1]],simdags[[1]])
compareDAGs(CONSmod1[[1]],simdags[[1]])
compareDAGs(CONSmod2[[1]],simdags[[1]])
```

---

mappings *mappings*

Description

An example of mappings needed for constructing bnInfo objects; a list of data frames, one for each omics type.

Usage

```
mappings
```

Format

a list of data frames, whose names correspond to omics types. The row names of each data frame correspond to IDs used in the data. At least one column "gene" is needed to specify gene symbol corresponding to the ID.

---

penInit *Initializing penalization matrix*

Description

This function can be used to initialize a penalization matrix for bnclustOmics clustering

Usage

```
penInit(
  bnnames,
  pfbase = 1,
  intpf = pfbase,
  intlist = NULL,
  intsame = 1,
  usescore = FALSE
)
```

Arguments

bnnames	object of class bnInfo; see bnInfo
pfbase	a numeric value more or equal to 1, base penalization factor; 1 by default (no penalization)
intpf	(optional) a numeric value more or equal to 1, this value will be used to penalize interactions from 'intlist'
intlist	(optional) a matrix or data frame containing a list of interactions and optionally their scores; 2 columns are necessary 'gene1' and 'gene2'
intsame	penalization factor for edges connecting the same genes
usescore	(logical) when TRUE, interactions score from column 'score' of the parameter 'intlist' will be used to define penalization factor

Value

returns a square matrix containing edge specific penalization factors

Author(s)

Polina Suter

---

penUpdateInter *Updating penalization matrix (between two omics types)*

Description

This function can be used to update an existing penalization matrix

Usage

```
penUpdateInter(
  penmat,
  bnnames,
  type1,
  type2,
  intlist = NULL,
  pfbase = 2,
  intpf = 1,
  intsame = 1,
  bi = FALSE
)
```

Arguments

penmat	a square penalization matrix; to initialize use penInit
bnnames	object of class bnInfo; see bnInfo
type1	name of omics type (from)
type2	name of omics type (to)
intlist	(optional) a matrix or data frame containing a list of interactions and optionally their scores; 2 columns are necessary 'gene1' and 'gene2'
pfbase	a numeric value more or equal to 1, base penalization factor; 2 by default (1 corresponds to no penalization)
intpf	(optional) a numeric value more or equal to 1, this value will be used to penalize interactions from 'intlist'
intsame	penalization factor for edges connecting the same genes
bi	(logical) indicates if interactions should be considered bi-directed

Value

returns a square matrix containing edge specific penalization factors

Author(s)

Polina Suter

---

penUpdateIntra *Updating penalization matrix (intra one omics type)*

Description

This function can be used to update an existing penalization matrix

Usage

```
penUpdateIntra(
  penmat,
  bnnames,
  type,
  intlist,
  pfbase = 2,
  intpf = 1,
  intsame = 1,
  bi = FALSE
)
```

Arguments

penmat	a square penalization matrix; to initialize use penInit
bnnames	object of class bnInfo; see bnInfo
type	name of omic type
intlist	(optional) a matrix or data frame containing a list of interactions and optionally their scores; 2 columns are necessary 'gene1' and 'gene2'
pfbase	a numeric value more or equal to 1, base penalization factor; 2 by default (1 corresponds to no penalization)
intpf	(optional) a numeric value more or equal to 1, this value will be used to penalize interactions from 'intlist'
intsame	penalization factor for edges connecting the same genes
bi	(logical) indicates if interactions should be considered bi-directed

Value

returns a square matrix containing edge specific penalization factors

Author(s)

Polina Suter

---

plotNode *Plotting all connections of one node*

Description

This function plots all connections (incoming and outgoing) of a specific node in one network or in all network in the discovered model.

Usage

```
plotNode(localint, node, p = 0.3, rmult = 7, dbcheck = TRUE, cex = 0.5)
```

Arguments

localint	an annotated list of interactions obtained by the function annotateEdges
node	center node name
p	defines a threshold for the posterior probability; edges whose posterior is higher than the threshold will be plotted
rmult	defines the raduis of the circle
dbcheck	logical, defines if interactions absent in the database are denoted with a dashed line
cex	regulates font size

Value

plots a graph consisting of a specified node and its neighbours in the networks representing clusters identified by 'bnclustOmics'

Author(s)

Polina Suter

Examples

```
bnnames<-bnInfo(simdata,c("b","c"),c("M","T"))
allInteractions<-annotateEdges(bnres3,bnnames,sump=1.2,minp=0.5,minkp=0.9,dblist=simint)
plotNode(allInteractions,"T43",p=0.5)
plotNode(allInteractions,"T43",p=0.5,dbcheck=FALSE)
```

---

posteriors *Extracting edge posterior probabilities*

Description

This function extracts a list of matrices containing posterior probabilities of all edges in the graphs discovered by bnclustOmics when the parameter 'epmatrix' was set to TRUE

Usage

```
posteriors(x)
```

Arguments

x	object of class 'bnclustOmics'

Value

a list of matrices containing posterior probabilities of all edges in the graphs discovered by bnclustOmics when the parameter 'epmatrix' was set to TRUE

Examples

```
post<-posteriors(bnres4)
```

relabelSimulation *Relabeling clusters*

Description

When running simulations studies, discovered cluster labels may differ from the ground truth cluster labels. This functions can be used to perform relabeling in order to compare discovered graphs to the ground truth graphs correctly.

Usage

```
relabelSimulation(res, trueclusters)
```

Arguments

res	object of class 'bnclustOmics'
trueclusters	ground truth clustering

Value

object of class 'bnclustOmics'

simclusters *simclusters*

Description

Vector containing true cluster assigments for data in the dataset "simdata"

Usage

```
simclusters
```

Format

a vector of 90 integers

simdags *simdags*

Description

A list of three matrices representing adjacency matrices of DAGs used to generate the simulated dataset 'simdata'. Each DAG consists of 20 binary (mutations) and 50 continuous nodes (Gaussian).

Usage

```
simdags
```

Format

a list of three binary matrices, each of size 70x70

simdata *simdata*

Description

A list of two matrices containing simulated mutations and transcriptome data (normalized, transformed) for three clusters. The generative model is the mixture of Bayesian networks (linear Gaussian model). The networks are stored in the dataset 'simdags'. Ground truth cluster assigments are stored in the dataset 'simclusters'

Usage

```
simdata
```

Format

a list of two matrices: 'M', 90 rows (samples) and 20 columns (mutations), 'T' 90 rows and 100 columns (gene expression)

simint *simint*

Description

A list of interactions derived from the networks stored in 'simdags' that were used to generate the dataset 'simdata'

Usage

```
simint
```

Format

a data frame with two columns "gene1" and "gene2" and 73 rows

stringint *stringint*

Description

An example of interactions list used for constructing graphical prior (penalization and blacklist) matrices.

Usage

```
stringint
```

Format

a data frames that includes three columns, gene1, gene2 and score. Column score is optional and may be skipped when constructing prior.

toydata *toydata*

Description

Toy dataset containing five omics matrices that can be used for testing purposes.

Usage

```
toydata
```

Format

A list of five matrices, one for each omics type. Each matrix contains 45 rows corresponding to patient samples. The genes and gene products are in the columns.

- M mutations, binary, 20 columns
- CN copy number changes, ordinal, 20 columns
- T transcriptome, continuous (Gaussian), 20 columns
- P proteome, continuous (Gaussian), 15 columns
- PP phospho-proteome, continuous (Gaussian), 20 columns

Index