Package 'miic'

February 2, 2018

Type Package

Title Learning Causal or Non-Causal Graphical Models Using Information Theory

Version 1.0.3

Date 2018-02-2

Description We report an information-theoretic method which learns a large class of causal or non-causal graphical models from purely observational data, while including the effects of unobserved latent variables, commonly found in many datasets. Starting from a complete graph, the method iteratively removes dispensable edges, by uncovering significant information contributions from indirect paths, and assesses edge-specific confidences from randomization of available data. The remaining edges are then oriented based on the signature of causality in observational data. This approach can be applied on a wide range of datasets and provide new biological insights on regulatory networks from single cell expression data, genomic alterations during tumor development and co-evolving residues in protein structures. For more information you can refer to: Verny et al. Plos Comput Biol. (2017) <doi:10.1371/journal.pcbi.1005662>.

Maintainer Nadir Sella <nadir.sella@curie.fr>

Imports MASS, igraph, bnlearn, ppcor, stats, Rcpp

License GPL (>= 2)

NeedsCompilation yes

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

LinkingTo Rcpp

Author Nadir Sella [aut, cre],
Louis Verny [aut],
Severine Affeldt [aut],
Hervé Isambert [aut]

Repository CRAN

Date/Publication 2018-02-02 14:29:57 UTC

R topics documented:

cosmicCancer 2
cosmicCancer_stateOrder 3
hematoData 3
miic 4
miic.evaluate.effn 8
miic.plot 9
miic.write.network.cytoscape 9
miic.write.style.cytoscape 10
ohno 11
ohno_stateOrder 11

Index **12**

`cosmicCancer` *Genomic and ploidy alterations in breast tumors*

Description

The dataset contains 807 samples without predisposing Brca1/2 germline mutations and includes 204 somatic mutations (from whole exome sequencing) and expression level information for 91 genes.

Usage

```
data(cosmicCancer)
```

Format

A data.frame object.

References

Forbes SA, Beare D, Gunasekaran P, Leung K, Bindal N, et al. (2015) Nucleic Acids Res 43:D805–D811. (PubMed link)

cosmicCancer_stateOrder

Genomic and ploidy alterations in breast tumors

Description

The dataset contains 807 samples without predisposing Brca1/2 germline mutations and includes 204 somatic mutations (from whole exome sequencing) and expression level information for 91 genes, cathegory order file.

Usage

```
data(cosmicCancer_stateOrder)
```

Format

A data.frame object.

References

Forbes SA, Beare D, Gunasekaran P, Leung K, Bindal N, et al. (2015) Nucleic Acids Res 43:D805–D811. (PubMed link)

hematoData

Early blood development: single cell binary gene expression data

Description

Binarized expression data of 33 transcription factors involved in early differentiation of primitive erythroid and endothelial cells (3934 cells).

Usage

```
data(hematoData)
```

Format

A data.frame object.

References

Moignard et al. (2015) Nat Biotechnol 33(3):269-76 (PubMed link)

`miic` *MIIC, causal network learning algorithm including latent variables*

Description

MIIC (Multivariate Information based Inductive Causation) combines constraint-based and information-theoretic approaches to disentangle direct from indirect effects amongst correlated variables, including cause-effect relationships and the effect of unobserved latent causes.

Usage

```
miic(inputData = NULL, categoryOrder = NULL, trueEdges = NULL,
  blackBox = NULL, nThreads = 1, cplx = c("nml", "mdl"),
  orientation = TRUE, propagation = TRUE, latent = FALSE, neff = -1,
  edges = NULL, confidenceShuffle = 0, confidenceThreshold = 0,
  confList = NULL, verbose = FALSE)
```

Arguments

`inputData`	[a data frame] A data frame that contains the observational data. Each column corresponds to one variable and each row is a sample that gives the values for all the observed variables. The column names correspond to the names of the observed variables. Data must be discrete like.
`categoryOrder`	[a data frame] An optional data frame giving information about how to order the various states of categorical variables. It will be used to compute the signs of the edges (using partial correlation coefficient) by sorting each variable's levels accordingly to the given category order.
`trueEdges`	[a data frame] An optional data frame containing all the true edges of the graph. Each line corresponds to one edge.
`blackBox`	[a data frame] An optional data frame containing the pairs of variables that should be considered as independent. Each row contains one column for each of the two variables. The variable name must correspond to the one in the *inputData* data frame.
`nThreads`	[a positive integer] When set greater than 1, the miic algorithm allows to use multithreading in the skeleton initialization phase.
`cplx`	[a string; *c("nml", "mdl")*] In practice, the finite size of the input dataset requires that the 2-point and 3-point information measures should be *shifted* by a *complexity* term. The finite size corrections can be based on the Minimal Description Length (MDL) criterion (set the option with "mdl"). In practice, the MDL complexity criterion tends to underestimate the relevance of edges connecting variables with many different categories, leading to the removal of false negative edges. To avoid such biases with finite datasets, the (universal) Normalized Maximum Likelihood (NML) criterion can be used (set the option with "nml"). The default is "nml" (see Affeldt *et al.*, UAI 2015).

`orientation` [a boolean value] The miic network skeleton can be partially directed by orienting and propagating edge directions, based on the sign and magnitude of the conditional 3-point information of unshielded triples. The propagation procedure relyes on probabilities; for more details, see Verny *et al.*, PLoS Comp. Bio. 2017). If set to FALSE the orientation step is not performed.

`propagation` [a boolean value] If set to FALSE, the skeleton is partially oriented with only the v-structure orientations. Otherwise, the v-structure orientations are propagated to downstream undirected edges in unshielded triples following the orientation method

`latent` [a boolean value] When set to TRUE, the network reconstruction is taking into account hidden (latent) variables. Dependence between two observed variables due to a latent variable is indicated with a '6' in the adjacency matrix and in the network edges.summary and by a bi-directed edge in the (partially) oriented graph.

`neff` [a positive integer] The N samples given in the *inputdata* data frame are expected to be independent. In case of correlated samples such as in time series or Monte Carlo sampling approaches, the effective number of independent samples *neff* can be estimated using the decay of the autocorrelation function (Verny *et al.*, PLoS Comp. Bio. 2017). This *effective* number *neff* of *independent* samples can be provided using this parameter.

`edges` [a data frame] The miic$edges object returned by an execution of the miic function. It represents the result of the skeleton step. If this object is provided, the skelethon step will not be done, and the required orientation will be performed using this edges data frame.

`confidenceShuffle`
[a positive integer] The number of shufflings of the original dataset in order to evaluate the edge specific confidence ratio of all inferred edges.

`confidenceThreshold`
[a positive floating point] The threshold used to filter the less probable edges following the skeleton step. See Verny *et al.*, PLoS Comp. Bio. 2017.

`confList` [a data frame] An optional data frame containing the confFile data frame returned by a miic execution. It is useful when a second run of the same input data set has to be performed with a different confidence threshold and the same confidenceShuffle value. In this way the computations based on the randomized dataset do not need to be performed again, and the values in this data frame are used instead.

`verbose` [a boolean value] If TRUE, debugging output is printed.

Details

Starting from a complete graph, the method iteratively removes dispensable edges, by uncovering significant information contributions from indirect paths, and assesses edge-specific confidences from randomization of available data. The remaining edges are then oriented based on the signature of causality in observational data.

Value

A *miic-like* object that contains:

- all.edges.summary: a data frame with information about the relationship between each pair of variables
  - *x:* X node
  - *y:* Y node
  - *type:* contains 'N' if the edge has been removed or 'P' for retained edges. If a true edges file is given, 'P' becomes 'TP' (True Positive) or 'FP' (False Positive), while 'N' becomes 'TN' (True Negative) or 'FN' (False Negative).
  - *ai:* the contributing nodes found by the method which participate in the mutual information between *x* and *y*, and possibly separate them.
  - *info:* provides the final mutual information times *Nxy_ai* for the pair (*x*, *y*) when conditioned on the collected nodes *ai*.
  - *cplx:* gives the computed complexity between the (*x*, *y*) variables taking into account the contributing nodes *ai*.
  - *Nxy_ai:* gives the number of samples on which the information and the complexity have been computed. If the input dataset has no missing value, the number of samples is the same for all pairs and corresponds to the total number of samples.
  - *log_confidence:* represents the *info* - *cplx* value. It is a way to quantify the strength of the edge (*x*, *y*).
  - *confidenceRatio:* this column is present if the confidence cut is > 0 and it represents the ratio between the probability to reject the edge (*x*, *y*) in the dataset versus the mean probability to do the same in multiple (user defined) number of randomized datasets.
  - *infOrt:* the orientation of the edge (*x*, *y*). It is the same value as in the adjacency matrix at row *x* and column *y*.
  - *trueOrt:* the orientation of the edge (*x*, *y*) present in the true edges file (if true edges file is provided).
  - *isOrtOk:* information about the consistency of the inferred graph's orientations with a reference graph is given (i.e. if true edges file is provided). Y: the orientation is consistent; N: the orientation is not consistent with the PAG derived from the given true graph.
  - *sign:* the sign of the partial correlation between variables *x* and *y*, conditioned on the contributing nodes *ai*.
  - *partial_correlation:* value of the partial correlation for the edge (*x*, *y*) conditioned on the contributing nodes *ai*.
- retained.edges.summary: a data frame in the format of all.edges.summary containing only the inferred edges.
- orientations.prob: this data frame lists the orientation probabilities of the two edges of all unshielded triples of the reconstructed network with the structure: node1 – mid-node – node2:
  - node1: node at the end of the unshielded triplet
  - p1: probability of the arrowhead node1 <- mid-node
  - p2: probability of the arrowhead node1 -> mid-node
  - mid-node: node at the center of the unshielded triplet
  - p3: probability of the arrowhead mid-node <- node2

  - p4: probability of the arrowhead mid-node -> node2
  - node2: node at the end of the unshielded triplet
  - NI3: 3 point (conditional) mutual information * N
- AdjMatrix: the adjacency matrix is a square matrix used to represent the inferred graph. The entries of the matrix indicate whether pairs of vertices are adjacent or not in the graph. The matrix can be read as a (row, column) set of couples where the row represents the source node and the column the target node. Since miic can reconstruct mixed networks (including directed, undirected and bidirected edges), we will have a different digit for each case:
  - 1: (x, y) edge is undirected
  - 2: (x, y) edge is directed as x -> y
  - -2: (x, y) edge is directed as x <- y
  - 6: (x, y) edge is bidirected

References

- Verny et al., *PLoS Comp. Bio. 2017.*

Examples

```
library(miic)

# EXAMPLE HEMATOPOIESIS
data(hematoData)

# execute MIIC (reconstruct graph)
miic.res = miic(inputData = hematoData, latent = TRUE,
confidenceShuffle = 10, confidenceThreshold = 0.001)

# plot graph
miic.plot(miic.res)
## Not run:

# write graph to graphml format. Note that to correctly visualize
# the network we created the miic style for Cytoscape (http://www.cytoscape.org/).

miic.write.network.cytoscape(g = miic.res, file.path(tempdir(),"/temp"))

# EXAMPLE CANCER
data(cosmicCancer)
data(cosmicCancer_stateOrder)
# execute MIIC (reconstruct graph)
miic.res = miic(inputData = cosmicCancer, categoryOrder = cosmicCancer_stateOrder, latent = TRUE,
confidenceShuffle = 100, confidenceThreshold = 0.001)

# plot graph
miic.plot(miic.res, igraphLayout=igraph::layout_on_grid)

# write graph to graphml format. Note that to correctly visualize
# the network we created the miic style for Cytoscape (http://www.cytoscape.org/).
miic.write.network.cytoscape(g = miic.res, file = file.path(tempdir(),"/temp"))
```

```
# EXAMPLE OHNOLOGS
data(ohno)
data(ohno_stateOrder)
# execute MIIC (reconstruct graph)
miic.res = miic(inputData = ohno, latent = TRUE, categoryOrder = ohno_stateOrder,
confidenceShuffle = 100, confidenceThreshold = 0.001)

# plot graph
miic.plot(miic.res)

# write graph to graphml format. Note that to correctly visualize
# the network we created the miic style for Cytoscape (http://www.cytoscape.org/).
miic.write.network.cytoscape(g = miic.res, file = file.path(tempdir(),"/temp"))

## End(Not run)
```

`miic.evaluate.effn` *Evaluate the effective number of samples*

Description

This function evaluates the effective number of samples in a dataset.

Usage

```
miic.evaluate.effn(inputData = NULL, plot = T)
```

Arguments

`inputData`	[a data frame] A data frame that contains the observational data. Each column corresponds to one variable and each row is a sample that gives the values for all the observed variables. The column names correspond to the names of the observed variables. Data must be discrete like.
`plot`	[a boolean value] if the autocorrelation plot has to be done. It will be performed only if all values of the correlation vector are positive.

Value

A list containing the autocorrelation decay, the effective number of samples, and the result of an exponentiality test with alpha = 0.05

miic.plot *Igraph plotting function for miic*

Description

This functions plots the network with the given layout (if specified) using the igraph package.

Usage

```
miic.plot(g, method = "log_confidence", igraphLayout = NULL,
  userLayout = NULL, verbose = F)
```

Arguments

g	[a miic graph object] The graph object returned by the miic execution.
method	[a string; *c("pcor", "log_confidence")*] The column used to plot the strength of the edges. Default: pcor.
igraphLayout	[an igraph layout object] When set it is used to plot the network. See the igraph manual for more information. Default: *layout_with_kk* or *layout.circle* if the network has more than 40 nodes.
userLayout	[a data frame] An optional data frame reporting the position of nodes. Each line corresponds to the *(x,y)* coordinates of each vertex. This data frame must have three columns, the first containing the name of the vertex as indicated in the colnames of the input data frame, the two others reporting the x and y positions.
verbose	[a boolean value] If TRUE, debugging output is printed.

Details

The plot reports the partial correlation or the log_confidence as strength of the edges.

miic.write.network.cytoscape *Graphml writing function for the miic graph*

Description

This function writes the network using the graphml format (http://graphml.graphdrawing.org/).

Usage

```
miic.write.network.cytoscape(g, layout = NULL, file)
```

Arguments

g	[a miic graph object] The graph object returned by the miic execution.
layout	[a data frame]. An optional data frame of 2 (or 3) columns containing the coordinate x and y for each node. The optional first column can contain node names. If node names is not given, the order of the input file will be assigned to the list of positions.
file	[a string] The file path of the output file (containing the file name without extension).

Details

The network is written in the graphml file format with all the features contained in the retained.edges.summary data frame

---

miic.write.style.cytoscape

Style writing function for the miic network

Description

This function writes the miic style for a correct visualization using the cytoscape tool (http://www.cytoscape.org/).

Usage

```
miic.write.style.cytoscape(file)
```

Arguments

file	[a string] The file path of the output file (containing the file name without extension).

Details

The style is written in the xml file format.

ohno *Tetraploidization in vertebrate evolution*

Description

20,415 protein-coding genes in the human genome from Ensembl (v70) and information on the retention of duplicates originating either from the two whole genome duplications at the onset of vertebrates ('ohnolog') or from subsequent small scale duplications ('SSD') as well as copy number variants ('CNV').

Usage

```
data(ohno)
```

Format

A data.frame object.

References

Verny et al., PLoS Comp. Bio. 2017.

ohno_stateOrder *Tetraploidization in vertebrate evolution*

Description

20,415 protein-coding genes in the human genome from Ensembl (v70) and information on the retention of duplicates originating either from the two whole genome duplications at the onset of vertebrates ('ohnolog') or from subsequent small scale duplications ('SSD') as well as copy number variants ('CNV'), cathegory order.

Usage

```
data(ohno_stateOrder)
```

Format

A data.frame object.

References

Verny et al., PLoS Comp. Bio. 2017.

Index

∗Topic **datasets**
cosmicCancer, 2
cosmicCancer_stateOrder, 3
hematoData, 3
ohno, 11
ohno_stateOrder, 11
∗Topic **data**
cosmicCancer, 2
cosmicCancer_stateOrder, 3
hematoData, 3
ohno, 11
ohno_stateOrder, 11

cosmicCancer, 2
cosmicCancer_stateOrder, 3

hematoData, 3

miic, 4
miic.evaluate.effn, 8
miic.plot, 9
miic.write.network.cytoscape, 9
miic.write.style.cytoscape, 10

ohno, 11
ohno_stateOrder, 11