Package ‘fastGeneMI’

May 1, 2017

Type Package

Title A range of mutual information estimators for gene regulatory network inference from microarray expression data.

Version 1.0

Date 2017-02-08

Author Jonathan Ish-Horowicz

Maintainer Jonathan Ish-Horowicz <jonathan.horowicz@mrc-bsu.cam.ac.uk>

Description Provides implementations of several mutual information estimators for use with the gene network inference algorthims CLR, ARACNE and MRNET.

License MIT

LazyData true

Imports Rcpp (>= 0.12.9), infotheo, parmigene, PRROC, KernSmooth, parallel, minet

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 6.0.1

R topics documented:

fastGeneMI-package *A range of mutual information estimators for gene regulatory network inference from microarray expression data.*

Description

Provides implementations of several mutual information estimators for use with the gene network inference algorthims CLR, ARACNE and MRNET.

Details

The DESCRIPTION file: This package was not yet installed at build time.

Index: This package was not yet installed at build time.
Overview

Author(s)

Jonathan Ish-Horowicz

Maintainer: Jonathan Ish-Horowicz <jonathan.horowicz@mrc-bsu.cam.ac.uk>

References

References

See Also

Links

get.mim.bspline *B-spline mutual information estimate*

Description

Returns a symmetric matrix of pairwise B-spline mutual information estimates from a samples x genes matrix of expression values. The ij^{th} element is the mutual information estimate between the expression of genes i and j.

Usage

```
get.mim.bspline(expr.data, order, n.bins = as.integer(nrow(expr.data)^(1/3)),
  n.cores = 1L)
```

Arguments

expr.data A samples x genes matrix of expression values.

order The order of the B-spline used to smooth the histograms. This must be greater than 1 + the number of bins.

n.bins The number of bins into which the expression data will be discretised. The default is $N^{1/3}$ for N samples.

n.cores The number of cores to use for the computation. The default is 1.

Value

A symmetric matrix of pairwise mutual information between the expression of pairs of genes. These are the mutual information estimtates between pairs of columns of `expr.data`.

get.mim.CS *Chao-Shen Likelihood mutual information estimate*

Description

Returns a symmetric matrix of pairwise Chao-Shen mutual information estimates from a samples x genes matrix of expression values. The ij^{th} element is the mutual information estimate between the expression of genes i and j.

Usage

```
get.mim.CS(expr.data, discretisation = c("equalwidth", "equalfreq", "bb"),
  n.bins = as.integer(nrow(expr.data)^(1/3)), n.cores = 1L)
```

Arguments

expr.data A samples x genes matrix of expression values.

discretisation
The types of bins into which the expression data will be discretised. Can choose equal width bins (`"equalwidth"`), equal frequency bins (`"equalfreq"`) or Bayesian Blocks bins (`"bb"`)

n.bins The number of bins into which the expression data will be discretised. The default is $N^{1/3}$ for N samples. Ignored if using Bayesian Blocks discretisation.

n.cores The number of cores to use for the computation. The default is 1.

Value

A symmetric matrix of pairwise mutual information between the expression of pairs of genes. These are the mutual information estimtates between pairs of columns of `expr.data`.

get.mim.knn *k-nearest-neighbour mutual information estimate*

Description

Returns a symmetric matrix of pairwise k-nearest-neighbour mutual information estimates from a samples x genes matrix of expression values. The ij^{th} element is the mutual information estimate between the expression of genes i and j. This is a wrapper for the `parmigene` function `knnmi.all`.

Usage

```
get.mim.knn(expr.data, k = 3, noise = 1e-09)
```

Arguments

`expr.data`	A samples x genes matrix of expression values.
`k`	The number of nearest neighbours to use. The default is 3
`noise`	The amplitude of the random noise added to the expression values to break ties. Default value is 1e-09.

Value

A symmetric matrix of pairwise mutual information between the expression of pairs of genes. These are the mutual information estimtates between pairs of columns of `expr.data`.

get.mim.ML *Maximum Likelihood mutual information estimate*

Description

Returns a symmetric matrix of pairwise Maximum Likelihood mutual information estimates from a samples x genes matrix of expression values. The ij^{th} element is the mutual information estimate between the expression of genes i and j.

Usage

```
get.mim.ML(expr.data, discretisation = c("equalwidth", "equalfreq", "bb"),
  n.bins = as.integer(nrow(expr.data)^(1/3)), n.cores = 1L)
```

Arguments

`expr.data` A samples x genes matrix of expression values.

`discretisation` The types of bins into which the expression data will be discretised. Can choose equal width bins (`"equalwidth"`), equal frequency bins (`"equalfreq"`) or Bayesian Blocks bins (`"bb"`)

`n.bins` The number of bins into which the expression data will be discretised. The default is $N^{1/3}$ for N samples. Ignored if using Bayesian Blocks discretisation.

`n.cores` The number of cores to use for the computation. The default is 1.

Value

A symmetric matrix of pairwise mutual information between the expression of pairs of genes. These are the mutual information estimtates between pairs of columns of `expr.data`.

`get.mim.MM` *Miller-Madow mutual information estimate*

Description

Returns a symmetric matrix of pairwise Miller-Madow mutual information estimates from a samples x genes matrix of expression values. The ij^{th} element is the mutual information estimate between the expression of genes i and j.

Usage

```
get.mim.MM(expr.data, discretisation = c("equalwidth", "equalfreq", "bb"),
  n.bins = as.integer(nrow(expr.data)^(1/3)), n.cores = 1L)
```

Arguments

`expr.data` A samples x genes matrix of expression values.

`discretisation` The types of bins into which the expression data will be discretised. Can choose equal width bins (`"equalwidth"`), equal frequency bins (`"equalfreq"`) or Bayesian Blocks bins (`"bb"`)

`n.bins` The number of bins into which the expression data will be discretised. The default is $N^{1/3}$ for N samples. Ignored if using Bayesian Blocks discretisation.

`n.cores` The number of cores to use for the computation. The default is 1.

Value

A symmetric matrix of pairwise mutual information between the expression of pairs of genes. These are the mutual information estimtates between pairs of columns of `expr.data`.

`get.mim.shrink` *Shrinkage mutual information estimate*

Description

Returns a symmetric matrix of pairwise Shrinkage mutual information estimates from a samples x genes matrix of expression values. The ij^{th} element is the mutual information estimate between the expression of genes i and j.

Usage

```
get.mim.shrink(expr.data, discretisation = c("equalwidth", "equalfreq", "bb"),
  n.bins = as.integer(nrow(expr.data)^(1/3)), n.cores = 1L)
```

Arguments

`expr.data` A samples x genes matrix of expression values.

`discretisation` The types of bins into which the expression data will be discretised. Can choose equal width bins (`"equalwidth"`), equal frequency bins (`"equalfreq"`) or Bayesian Blocks bins (`"bb"`)

`n.bins` The number of bins into which the expression data will be discretised. The default is $N^{1/3}$ for N samples. Ignored if using Bayesian Blocks discretisation.

`n.cores` The number of cores to use for the computation. The default is 1.

Value

A symmetric matrix of pairwise mutual information between the expression of pairs of genes. These are the mutual information estimtates between pairs of columns of `expr.data`.

`get.pr` *Compute the precision and recall from a matrix of mutual information estimates*

Description

From a symmetric matrix of mutual information values between the expression of pairs of genes, infer a network then compute the precision, recall and the area under precision-recall curve by comparing the inferred network to the gold standard

Usage

```
get.pr(mim, gs.net, inf.algo = c("clr", "mrnet", "aracne"), n.reg = NULL)
```

Arguments

`mim` A symmetric matrix of mutual information values. The ij^{th} element is the mutual information between the expression of genes i and j

`gs.net` The symmetric adjacency matrix of the gold standard regulatory network. Should contain a 0 for no edge, 1 for an edge and `NA` if unknown.

`inf.algo` The inference algorithm used to infer the network from the mutual information estimates. Must be one of `"clr"`, `"mrnet"` or `"aracne"`

`n.reg` The number of genes that are designated as potential regulators. If it is an integer then the first `n.reg` genes are marked as regulators. If a vector then those gene indexes contained in the vector are regulators. Only interactions involving regulators are used to evaluate the precision and recall. Default is `NULL`, in which case all the genes are potential regulators.

Value

`pr` A two-column matrix of the precision and recall values of the precision-recall curve. The first column contains the recall and the second column contains the precision.

`auprc` The area under the precision-recall curve.