# Package 'fastGeneMI'

March 25, 2019

**Type** Package

**Title** A Suite of Mutual Information Estimators for Gene Regulatory
Network Inference from Microarray Expression Data

**Version** 1.0

**Date** 2019-03-22

**Author** Jonathan Ish-Horowicz

**Maintainer** Jonathan Ish-Horowicz <jonathan.ish-horowicz17@imperial.ac.uk>

**Description** Provides implementations of several mutual information estimators
for use with the gene network inference algorithms CLR, ARACNE and MRNET.
Computations are performed in C++ using Rcpp and RcppArmadillo.

**Depends** R (>= 2.10)

**License** MIT + file LICENSE

**LazyData** true

**Imports** Rcpp (>= 0.12.9), infotheo, PRROC, minet, testthat

**LinkingTo** Rcpp, RcppArmadillo

**RoxygenNote** 6.1.1

**Encoding** UTF-8

**NeedsCompilation** yes

## R topics documented:

---

| fastGeneMI-package | *A Suite 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 algorithms CLR, ARACNE and MRNET. Computations are performed in C++ using Rcpp and RcppArmadillo.

### Details

The DESCRIPTION file:

| | |
|---|---|
| Package: | fastGeneMI |
| Type: | Package |
| Title: | A Suite of Mutual Information Estimators for Gene Regulatory Network Inference from Microarray Expressi |
| Version: | 1.0 |
| Date: | 2019-03-22 |
| Author: | Jonathan Ish-Horowicz |
| Maintainer: | Jonathan Ish-Horowicz <jonathan.ish-horowicz17@imperial.ac.uk> |
| Description: | Provides implementations of several mutual information estimators for use with the gene network inference a |
| Depends: | R (>= 2.10) |
| License: | MIT + file LICENSE |
| LazyData: | true |
| Imports: | Rcpp (>= 0.12.9), infotheo, PRROC, minet, testthat |
| LinkingTo: | Rcpp, RcppArmadillo |
| RoxygenNote: | 6.1.1 |
| Encoding: | UTF-8 |

Index of help topics:

```
fastGeneMI-package      A Suite of Mutual Information Estimators for
                        Gene Regulatory Network Inference from
                        Microarray Expression Data
get.bayesian.blocks.bins
                        Bayesian Blocks discretisation of expression
                        data
get.mim.CS              Chao-Shen Likelihood mutual information
                        estimate
get.mim.ML              Maximum Likelihood mutual information estimate
get.mim.MM              Miller-Madow mutual information estimate
get.mim.bspline         B-spline mutual information estimate
get.mim.shrink          Shrinkage mutual information estimate
get.pr                  Compute the precision and recall of a network
                        inferred from pairwise mutual information
```

|           | estimates by comparing to a gold standard |
|-----------|-------------------------------------------|
| infer.net | Infer a regulatory network from a matrix of mutual information values between pairs of genes using one of CLR, MRNET or ARACNe |

Overview

### Author(s)

Jonathan Ish-Horowicz

Maintainer: Jonathan Ish-Horowicz <jonathan.ish-horowicz17@imperial.ac.uk>

### References

References

### See Also

Links

---

get.bayesian.blocks.bins

*Bayesian Blocks discretisation of expression data*

---

### Description

Returns the bin edges found using the Bayesian Blocks algorithm for the expression of a single gene

### Usage

```
get.bayesian.blocks.bins(expr.prof)
```

### Arguments

expr.prof     A vector of expression values of a single gene.

### Value

A vector of bin edges.

### References

Scargle, J.D. et al, 2013. Studies in astronomical time series analysis. VI. Bayesian block representations. The Astrophysical Journal, 764(2), p.167.

## Examples

```
# Compute the bin edges for the first variable of the iris
# dataset according to Bayesian Blocks
data("iris")
bin.edges <- get.bayesian.blocks.bins(iris[,1])
```

---

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 $j$ in nats.

## 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.

## References

Daub, C.O. et al, 2004. Estimating mutual information using B-spline functions-an improved similarity measure for analysing gene expression data. BMC Bioinformatics, 5(1), p.118.

## Examples

```
# Compute the mutual information matrix for the variables in the
# iris dataset using default discretisation parameters
# (N^1/3 equal width bins for N samples) and spline order (2).
# Computation is performed sequentially be default (i.e. using a
# single core)
data("iris")
expr.data <- iris[,1:4]
mim <- get.mim.bspline(expr.data, 2)
# Now use N^1/2 bins and use 2 cores
mim <- get.mim.bspline(expr.data, 2, as.integer(nrow(expr.data)^0.5),
                       n.cores=2)
# Infer the regulatory network using CLR. No evaluation is performed
# as no gold standard is provided
net <- infer.net(mim, "clr")
```

---

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 $j$ in nats.

## 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.

**References**

Chao, A. and Shen, T.J., 2003. Nonparametric estimation of Shannon's index of diversity when there are unseen species in sample. Environmental and Ecological Statistics, 10(4), pp.429-443.

Scargle, J.D. et al, 2013. Studies in astronomical time series analysis. VI. Bayesian block representations. The Astrophysical Journal, 764(2), p.167.

**Examples**

```
# Compute the mutual information matrix for the variables in the
# iris dataset using default discretisation parameters
# (N^1/3 equal width bins for N samples). Computation is performed
# sequentially be default (i.e. using a single core)
data("iris")
expr.data <- iris[,1:4]
mim <- get.mim.CS(expr.data, "equalwidth")
# Now use N^1/2 bins and use 2 cores
mim <- get.mim.CS(expr.data, "equalwidth", as.integer(nrow(expr.data)^0.5),
                  n.cores=2)
# Infer the regulatory network using CLR. No evaluation is performed
# as no gold standard is provided
net <- infer.net(mim, "clr")
```

---

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`.

**References**

Scargle, J.D. et al, 2013. Studies in astronomical time series analysis. VI. Bayesian block representations. The Astrophysical Journal, 764(2), p.167.

**Examples**

```
# Compute the mutual information matrix for the variables in the
# iris dataset using default discretisation parameters
# (N^1/3 equal width bins for N samples). Computation is performed
# sequentially be default (i.e. using a single core)
data("iris")
expr.data <- iris[,1:4]
mim <- get.mim.ML(expr.data, "equalwidth")
# Now use N^1/2 bins and use 2 cores
mim <- get.mim.ML(expr.data, "equalwidth", as.integer(nrow(expr.data)^0.5),
                  n.cores=2)
```

---

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`.

**References**

Miller, G., 1955. Note on the bias of information estimates. Information Theory in Psychology: Problems and Methods.

Scargle, J.D. et al, 2013. Studies in astronomical time series analysis. VI. Bayesian block representations. The Astrophysical Journal, 764(2), p.167.

**Examples**

```
# Compute the mutual information matrix for the variables in the
# iris dataset using default discretisation parameters
# (N^1/3 equal width bins for N samples). Computation is performed
# sequentially be default (i.e. using a single core)
data("iris")
expr.data <- iris[,1:4]
mim <- get.mim.MM(expr.data, "equalwidth")
# Now use N^1/2 bins and use 2 cores
mim <- get.mim.MM(expr.data, "equalwidth", as.integer(nrow(expr.data)^0.5),
                    n.cores=2)
```

---

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 $j$ in nats.

**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`.

## References

Hausser, J. and Strimmer, K., 2009. Entropy inference and the James-Stein estimator, with application to nonlinear gene association networks. Journal of Machine Learning Research, 10(Jul), pp.1469-1484.

Scargle, J.D. et al, 2013. Studies in astronomical time series analysis. VI. Bayesian block representations. The Astrophysical Journal, 764(2), p.167.

## Examples

```
# Compute the mutual information matrix for the variables in the
# iris dataset using default discretisation parameters
# (N^1/3 equal width bins for N samples). Computation is performed
# sequentially be default (i.e. using a single core)
data("iris")
expr.data <- iris[,1:4]
mim <- get.mim.shrink(expr.data, "equalwidth")
# Now use N^1/2 bins and use 2 cores
mim <- get.mim.shrink(expr.data, "equalwidth", as.integer(nrow(expr.data)^0.5),
                  n.cores=2)
# Infer the regulatory network using CLR. No evaluation is performed
# as no gold standard is provided
net <- infer.net(mim, "clr")
```

---

get.pr                          *Compute the precision and recall of a network inferred from pairwise*
                                *mutual information estimates by comparing to a gold standard*

---

## 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. If the regulatory relationship between two genes is unknown then use NA as the appropriate element of `gs.net`. These are ignnored when computing the precision and recall. This is a thin wrapper around `infer.net` that does not return the inferred network and is included for backward compatibility with a previous version.

## 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. NA values are ignored when computing the precscion and recall. |
| `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 (note that this is typically not the biological reality). |

## 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. |

## References

Faith, J.J. et al, 2007. Large-scale mapping and validation of Escherichia coli transcriptional regulation from a compendium of expression profiles. PLoS Biology, 5(1), p.e8.

Meyer, P.E. et al, 2007. Information-theoretic inference of large transcriptional regulatory networks. EURASIP Journal on Bioinformatics and Systems Biology, 2007, pp.8-8.

Margolin, A.A. et al, 2006, March. ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. BMC Bioinformatics (Vol. 7, No. 1, p. S7).

---

| | |
|---|---|
| infer.net | *Infer a regulatory network from a matrix of mutual information values between pairs of genes using one of CLR, MRNET or ARACNe* |

---

## Description

From a symmetric matrix of mutual information values between the expression of pairs of genes, infer a regulatory network. If a gold standard network is provided the inferred network is evaluated and the resulting precision, recall and the area under precision-recall is computed and returned along with the inferred network.

## Usage

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

## Arguments

| | |
|---|---|
| mim | A symmetric matrix of mutual information values. The $ij^{th}$ element is the mutual information between the expression of genes $i$ and $j$ |
| inf.algo | The inference algorithm used to infer the network from the mutual information estimates. Must be one of "clr", "mrnet" or "aracne". Default is "CLR". |
| 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. Note that unknown edges are not included in the evaluation. |
| 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 compute the precision and recall. Default is NULL, in which case all the genes are potential regulators (note that this is typically not the biological reality). |
| plot.pr | Logical controlling whether or not to return the precision-recall curve. |

## Value

| | |
|---|---|
| network | The inferred regulatory network. A matrix whose $ij^{th}$ element represents the confidence of an edge between genes $i$ and $j$. |
| pr | A two-column matrix of the precision and recall values of the precision-recall curve resulting from evaluating the inferred network against a gold standard. The first column contains the recall and the second column contains the precision. Only returned if a gold standard network is provided. |
| auprc | The area under the precision-recall curve. Only returned if a gold standard network is provided. |
| plot | The precision-recall curve (if plot.pr=TRUE). |

## References

Faith, J.J. et al, 2007. Large-scale mapping and validation of Escherichia coli transcriptional regulation from a compendium of expression profiles. PLoS Biology, 5(1), p.e8.

Meyer, P.E. et al, 2007. Information-theoretic inference of large transcriptional regulatory networks. EURASIP Journal on Bioinformatics and Systems Biology, 2007, pp.8-8.

Margolin, A.A. et al, 2006, March. ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. BMC Bioinformatics (Vol. 7, No. 1, p. S7).

## Examples

```
# Compute the mutual information matrix for the variables in the
# iris dataset using default discretisation parameters
# (N^1/3 equal width bins for N samples). Computation is performed
# sequentially be default (i.e. using a single core)
data("iris")
expr.data <- iris[,1:4]
mim <- get.mim.ML(expr.data, "equalwidth")
```

```
# Now use N^1/2 bins and use 2 cores
mim <- get.mim.ML(expr.data, "equalwidth", as.integer(nrow(expr.data)^0.5),
                  n.cores=2)
# Infer the network without a gold standard - no precision recall curve will
# be returned since evalatuaion requires a gold standard
net <- infer.net(mim, "clr")
```

# Index