

# Package ‘synRNASeqNet’

April 20, 2015

**Type** Package

**Title** Synthetic RNA-Seq Network Generation and Mutual Information Estimates

**Version** 1.0

**Date** 2015-04-07

**Author** Luciano Garofano, Stefano Maria Pagnotta, Michele Ceccarelli

**Maintainer** Luciano Garofano <lucianogarofano88@gmail.com>

**Depends** R (>= 3.1.1), parallel, parmigene, GenKern, igraph

**URL** <https://github.com/lucgar/synRNASeqNet>

**Imports** KernSmooth

**Description** It implements various estimators of mutual information, such as the maximum likelihood and the Millow-Madow estimator, various Bayesian estimators, the shrinkage estimator, and the Chao-Shen estimator. It also offers wrappers to the kNN and kernel density estimators. Furthermore, it provides various index of performance evaluation such as precision, recall, FPR, F-Score, ROC-PR Curves and so on. Lastly, it provides a brand new way of generating synthetic RNA-Seq Network with known dependence structure.

**License** GPL (>= 3)

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2015-04-20 11:55:18

## R topics documented:

synRNASeqNet-package . . . . .	2
aucDisc . . . . .	3
DiscriminantPower . . . . .	4
entropyBayes . . . . .	5
entropyCS . . . . .	6
entropyML . . . . .	7
entropyMM . . . . .	8

entropyShrink . . . . .	9
Likelihoods . . . . .	10
mainNetFunction . . . . .	11
parEntropyEstimate . . . . .	12
parMIEstimate . . . . .	14
parMIKD . . . . .	17
performanceIndex . . . . .	18
performanceNET . . . . .	19
plotPR . . . . .	20
plotROC . . . . .	20
simulatedData . . . . .	21
YoudenIndex . . . . .	22

## Index 24

---

synRNASeqNet-package    *Synthetic RNA-Seq Network Generation and Mutual Information Estimates*

---

### Description

It implements various estimators of mutual information, such as the maximum likelihood and the Millow-Madow estimator, various Bayesian estimators, the shrinkage estimator, and the Chao-Shen estimator. It also offers wrappers to the kNN and kernel density estimators.

Furthermore, it provides various index of performance evaluation such as precision, recall, FPR, F-Score, ROC-PR Curves and so on.

Lastly, it provides a brand new way of generating synthetic RNA-Seq Network with known dependence structure.

### Index

<a href="#">synRNASeqNet-package:</a>	Synthetic RNA-Seq Network Generation and Mutual Information Estimates
<a href="#">parMIEstimate:</a>	Parallel Mutual Information Estimation
<a href="#">parEntropyEstimate:</a>	Parallel Entropy Estimation
<a href="#">entropyML:</a>	Maximum Likelihood Entropy Estimate
<a href="#">entropyMM:</a>	Miller-Madow corrected Entropy Estimate
<a href="#">entropyBayes:</a>	Bayesian Entropy Estimate
<a href="#">entropyCS:</a>	Chao-Shen Entropy Estimate
<a href="#">entropyShrink:</a>	James-Stein Shrinkage Entropy Estimate
<a href="#">parMIKD:</a>	Parallel Kernel Density Mutual Information Estimate
<a href="#">simulatedData:</a>	Random Generation Networks for RNA-Seq Data
<a href="#">mainNetFunction:</a>	Main Estimation and Evaluation Function
<a href="#">plotROC:</a>	Plot ROC Curve
<a href="#">plotPR:</a>	Plot PR Curve
<a href="#">aucDisc:</a>	Calculate Area Under a (ROC/PR) Curve
<a href="#">performanceIndex:</a>	Evaluate Performance Indices
<a href="#">performanceNET:</a>	Evaluate Performance Indices

<a href="#">YoudenIndex:</a>	Youden's Index
<a href="#">Likelihoods:</a>	Likelihood Indices
<a href="#">DiscriminantPower:</a>	Discriminant Power

**Author(s)**

Luciano Garofano <lucianogarofano88@gmail.com>, Stefano Maria Pagnotta, Michele Ceccarelli

---

aucDisc

*Calculate Area Under a (ROC/PR) Curve*

---

**Description**

aucDisc calculates the Area under a Curve giving the coordinate of each point.

**Usage**

```
aucDisc(fpr, tpr)
```

**Arguments**

fpr	the false positive rate (precision/abscissa) that forms the curve.
tpr	the true positive rate (recall/ordinate) that forms the curve.

**Details**

It uses the 2nd order Taylor series in order to calculate the area under the curve.

**Value**

The aucDisc returns a real number representing the Area Under the given Curve.

**Author(s)**

Luciano Garofano <lucianogarofano88@gmail.com>, Stefano Maria Pagnotta, Michele Ceccarelli

**References**

Swets, J.A. (1996). Signal detection theory and ROC analysis in psychology and diagnostics: collected papers. *Lawrence Erlbaum Associates*, Mahwah, NJ.

**See Also**

[performanceIndex](#)

**Examples**

```
simData <- simulatedData(p = 50, n = 100, mu = 100, sigma = 0.25,
                        ppower = 0.73, noise = FALSE)
counts <- simData$counts
adjMat <- simData$adjMat

miML <- parMIEstimate(counts, method = "ML", unit = "nat", nchips = 2)
valML <- performanceIndex(miML, adjMat)

AUROC <- aucDisc(valML[, "FPR"], valML[, "Recall"])
AUPR <- aucDisc(valML[, "Recall"], valML[, "Precision"])
```

---

DiscriminantPower

*Discriminant Power*

---

**Description**

The Discriminant Power summarizes sensitivity and specificity.

**Usage**

```
DiscriminantPower(resTable)
```

**Arguments**

`resTable` the data.frame generated from the performanceNET (or one of the row of the matrix generated from the performanceIndex) function.

**Value**

The DiscriminantPower function returns the value of the Discriminant Power for a given predicted adjacency matrix.

**Author(s)**

Luciano Garofano <lucianogarofano88@gmail.com>, Stefano Maria Pagnotta, Michele Ceccarelli

**References**

Sokolova M., Japkowicz N., Szpakowicz S. (2006). Beyond Accuracy, F-Score and ROC: A Family of Discriminant Measures for Performance Evaluation. *Lecture Notes in Computer Science* Volume 4304, 2006, pp 1015-1021.

**See Also**

[YoudenIndex](#), [Likelihoods](#)

**Examples**

```
simData <- simulatedData(p = 10, n = 20, mu = 100, sigma = 0.25,
                        ppower = 0.73, noise = FALSE)
counts <- simData$counts
adjMat <- simData$adjMat

miML <- parMIEstimate(counts, method = "ML", unit = "nat", nchips = 2)
miML <- aracne.a(miML, eps = 0.5)

valML <- performanceNET(miML, adjMat)
DP <- DiscriminantPower(valML)
```

---

entropyBayes

*Bayesian Entropy Estimate*

---

**Description**

Computing the Maximul Likelihood Entropy Estimate of cellCounts.

**Usage**

```
entropyBayes(cellCounts, unit = unit,
             priorHyperParam = priorHyperParam)
```

**Arguments**

cellCounts	an integer vector (or matrix) representing the number of times each particular count is obtained.
unit	the unit in which entropy is measured. One of "bit" (log2, default), "ban" (log10) or "nat" (natural units).
priorHyperParam	the prior distribution type for the Bayes estimation. One of "Jeffreys" (default, Jeffreys Prior, Krichevsky and Trofimov Estimator), "BLUnif" (Bayes-Laplace uniform Prior, Holste Estimator), "Perks" (Perks Prior, Schurmann and Grassberger Estimator), or "MiniMax" (MiniMax Prior), can be abbreviated.

**Value**

The entropyBayes function returns the value of the entropy of that gene  $H(X)$  (or pair of genes  $H(X, Y)$ ).

**Author(s)**

Luciano Garofano <lucianogarofano88@gmail.com>, Stefano Maria Pagnotta, Michele Ceccarelli

## References

- Jeffreys H. (1946). An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London*, vol. 186 no. 1007 pp. 453-461.
- Krichevsky R.E., Trofimov V.K. (1981). The performance of universal encoding. *IEEE Transactions on Information Theory*, vol. 27 pp. 199-207.
- Holste D., Hertz H. (1998). Bayes' estimators of generalized entropies. *Journal of Physics A*, vol. 31 pp. 2551-2566.
- Perks W. (1947). Some observations on inverse probability including a new indifference rule. *Journal of the Institute of Actuaries*, vol. 73 pp. 285-334.
- Schurmann T., Grassberg P. (1996). Entropy estimation of symbol sequences. *Chaos*, vol. 6 pp. 414-427.
- Trybula S. (1958). Some problems of simultaneous minimax estimation. *The Annals of Mathematical Statistics*, vol. 29 pp. 245-253.

## See Also

[entropyML](#), [entropyMM](#), [entropyCS](#), [entropyShrink](#)

## Examples

```
simData <- simulatedData(p = 50, n = 100, mu = 100, sigma = 0.25,
                        ppower = 0.73, noise = FALSE)
cellCounts <- table(simData$counts[1, ])
eBJ <- entropyBayes(cellCounts, unit = "nat", priorHyperParam = "Jeffreys")
eBB <- entropyBayes(cellCounts, unit = "nat", priorHyperParam = "BLUnif")
eBP <- entropyBayes(cellCounts, unit = "nat", priorHyperParam = "Perks")
eBM <- entropyBayes(cellCounts, unit = "nat", priorHyperParam = "MiniMax")
```

---

entropyCS

*Chao-Shen Entropy Estimate*

---

## Description

Computing the Chao-Shen Entropy Estimate of cellCounts.

## Usage

```
entropyCS(cellCounts, unit = unit)
```

## Arguments

- |            |  |
|------------|--|
| cellCounts | an integer vector (or matrix) representing the number of times each particular count is obtained.            |
| unit       | the unit in which entropy is measured. One of "bit" (log2, default), "ban" (log10) or "nat" (natural units). |

**Value**

The entropyCS function returns the value of the entropy of that gene  $H(X)$  (or pair of genes  $H(X, Y)$ ).

**Author(s)**

Luciano Garofano <lucianogarofano88@gmail.com>, Stefano Maria Pagnotta, Michele Ceccarelli

**References**

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

**See Also**

[entropyML](#), [entropyMM](#), [entropyBayes](#), [entropyShrink](#)

**Examples**

```
simData <- simulatedData(p = 50, n = 100, mu = 100, sigma = 0.25,
                        ppower = 0.73, noise = FALSE)
cellCounts <- table(simData$counts[1, ])
eCS <- entropyCS(cellCounts, unit = "nat")
```

---

entropyML

*Maximum Likelihood Entropy Estimate*

---

**Description**

Computing the Maximul Likelihood Entropy Estimate of cellCounts.

**Usage**

```
entropyML(cellCounts, unit = unit)
```

**Arguments**

**cellCounts** an integer vector (or matrix) representing the number of times each particular count is obtained.

**unit** the unit in which entropy is measured. One of "bit" (log2, default), "ban" (log10) or "nat" (natural units).

**Value**

The entropyML function returns the value of the entropy of that gene  $H(X)$  (or pair of genes  $H(X, Y)$ ).

**Author(s)**

Luciano Garofano <lucianogarofano88@gmail.com>, Stefano Maria Pagnotta, Michele Ceccarelli

**References**

Paniski L. (2003). Estimation of Entropy and Mutual Information. *Neural Computation*, vol. 15 no. 6 pp. 1191-1253.

Meyer P.E., Laffitte F., Bontempi G. (2008). minet: A R/Bioconductor Package for Inferring Large Transcriptional Networks Using Mutual Information. *BMC Bioinformatics* 9:461.

Antos A., Kontoyiannis I. (2001). Convergence properties of functional estimates for discrete distributions. *Random Structures and Algorithms*, vol. 19 pp. 163-193.

Strong S., Koberle R., de Ruyter van Steveninck R.R., Bialek W. (1998). Entropy and Information in Neural Spike Trains. *Physical Review Letters*, vol. 80 pp. 197-202.

**See Also**

[entropyMM](#), [entropyBayes](#), [entropyCS](#), [entropyShrink](#)

**Examples**

```
simData <- simulatedData(p = 50, n = 100, mu = 100, sigma = 0.25,
                        ppower = 0.73, noise = FALSE)
cellCounts <- table(simData$counts[1, ])
eML <- entropyML(cellCounts, unit = "nat")
```

---

entropyMM

*Miller-Madow corrected Entropy Estimate*

---

**Description**

Computing the Miller-Madow corrected Entropy Estimate of cellCounts.

**Usage**

```
entropyMM(cellCounts, unit = unit)
```

**Arguments**

cellCounts	an integer vector (or matrix) representing the number of times each particular count is obtained.
unit	the unit in which entropy is measured. One of "bit" (log2, default), "ban" (log10) or "nat" (natural units).

**Value**

The entropyMM function returns the value of the entropy of that gene  $H(X)$  (or pair of genes  $H(X, Y)$ ).



**Author(s)**

Luciano Garofano <lucianogarofano88@gmail.com>, Stefano Maria Pagnotta, Michele Ceccarelli

**References**

Miller G.A. (1955). Note on the bias of information estimates. *Information Theory in Psychology*, II-B pp. 95-100.

**See Also**

[entropyML](#), [entropyBayes](#), [entropyCS](#), [entropyShrink](#)

**Examples**

```
simData <- simulatedData(p = 50, n = 100, mu = 100, sigma = 0.25,
                        ppower = 0.73, noise = FALSE)
cellCounts <- table(simData$counts[1, ])
eMM <- entropyMM(cellCounts, unit = "nat")
```

---

entropyShrink

*James-Stein Shrinkage Entropy Estimate*

---

**Description**

Computing the James-Stein Shrinkage Entropy Estimate of cellCounts.

**Usage**

```
entropyShrink(cellCounts, unit = unit, shrinkageTarget = shrinkageTarget)
```

**Arguments**

cellCounts      an integer vector (or matrix) representing the number of times each particular count is obtained.

unit            the unit in which entropy is measured. One of "bit" (log2, default), "ban" (log10) or "nat" (natural units).

shrinkageTarget      shrinkage target frequencies. If not specified (default) it is estimated in a James-Stein-type fashion (uniform distribution).

**Value**

The entropyShrink function returns the value of the entropy of that gene  $H(X)$  (or pair of genes  $H(X, Y)$ ).

**Author(s)**

Luciano Garofano <lucianogarofano88@gmail.com>, Stefano Maria Pagnotta, Michele Ceccarelli

**References**

James W., Stein C. (1961). Estimation with Quadratic Loss. *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1 pp. 361-379.

**See Also**

[entropyML](#), [entropyMM](#), [entropyBayes](#), [entropyCS](#)

**Examples**

```
simData <- simulatedData(p = 50, n = 100, mu = 100, sigma = 0.25,
                        ppower = 0.73, noise = FALSE)
cellCounts <- table(simData$counts[1, ])
eShrink <- entropyShrink(cellCounts, unit = "nat", shrinkageTarget = NULL)
```

---

Likelihoods

*Likelihood Indices*

---

**Description**

The Likelihood Indices accomodate both sensitivity and specificity, but treat them separately.

**Usage**

```
Likelihoods(resTable)
```

**Arguments**

`resTable` the data.frame generated from the performanceNET (or one of the row of the matrix generated from the performanceIndex) function.

**Value**

The Likelihoods function returns the value of the Likelihood Indices (rho - pos or neg) for a given predicted adjacency matrix.

**Author(s)**

Luciano Garofano <lucianogarofano88@gmail.com>, Stefano Maria Pagnotta, Michele Ceccarelli

## References

Sokolova M., Japkowicz N., Szpakowicz S. (2006). Beyond Accuracy, F-Score and ROC: A Family of Discriminant Measures for Performance Evaluation. *Lecture Notes in Computer Science* Volume 4304, 2006, pp 1015-1021.

## See Also

[YoudenIndex](#), [DiscriminantPower](#)

## Examples

```
simData <- simulatedData(p = 10, n = 20, mu = 100, sigma = 0.25,
                        ppower = 0.73, noise = FALSE)
counts <- simData$counts
adjMat <- simData$adjMat

miML <- parMIEstimate(counts, method = "ML", unit = "nat", nchips = 2)
miML <- aracne.a(miML, eps = 0.5)

valML <- performanceNET(miML, adjMat)
LH <- Likelihoods(valML)
```

---

mainNetFunction

*Main Estimation and Evaluation Function*

---

## Description

A function that computes the mutual information between all pairs of rows (or specified ones) of matrix counts using all the 10 different estimation methods and evaluate their performances.

## Usage

```
mainNetFunction(counts, adjMat, nchips, plotPath = "",
                tfList = NULL)
```

## Arguments

counts	a numeric matrix (for the reconstruction of gene regulatory networks, genes on rows and samples on columns).
adjMat	the adjacency matrix that encodes the graph structure that is going to be predicted.
nchips	the number of cpu's to be used for making the parallel calculation.
plotPath	the folder in which the plot will be saved.
tfList	the character vector specifying which genes from the rownames of the counts matrix is to be used as transcription factor for network reconstruction.

**Value**

miEst	a list containing the estimates of all methods.
valMet	a list containing the performance indices (i.e. "Recall", "FPR", "Precision", "Accuracy", "Fscore") calculated in all methods and usable for creating curves like ROC and PR.
resTable	a matrix with the best performances for each method.

**Author(s)**

Luciano Garofano <lucianogarofano88@gmail.com>, Stefano Maria Pagnotta, Michele Ceccarelli

**References**

Stehman, S.V. (1997). Selecting and interpreting measures of thematic classification accuracy. *Remote Sensing of Environment* 62 (1): 77-89.

**See Also**

[simulatedData](#)

**Examples**

```
simData <- simulatedData(p = 5, n = 10, mu = 50, sigma = 0.25,
                        ppower = 0.73, noise = FALSE)
counts <- simData$counts
adjMat <- simData$adjMat

#netData <- mainNetFunction(counts, adjMat, nchips = 2)
```

---

parEntropyEstimate      *Parallel Entropy Estimation*

---

**Description**

A function that computes the entropy between all pairs of rows (or specified ones) of matrix counts using the indirect methods.

**Usage**

```
parEntropyEstimate(idx, method = method, unit = unit,
                  priorHyperParam = priorHyperParam,
                  shrinkageTarget = shrinkageTarget, boot = boot)
```

**Arguments**

idx	the index of the cell which corresponds to the interaction going to be estimated.
method	a character string indicating which estimate is to be computed. One of "ML" (Maximum Likelihood Estimator, default), "MM" (Miller-Madow corrected Estimator), "Bayes" (Bayesian Estimators), "CS" (Chao-Shen Estimator), "Shrink" (James-Stein shrinkage Estimator), "KD" (kernel Density Estimator), or "KNN" (k-Nearest Neighbor Estimator), can be abbreviated. For the "Bayes" estimate it is needed to specify also which priorHyperParam is to be used; for "Shrink" is optional to specify values for the shrinkageTarget parameter; for "KNN" is needed to specify also the number of nearest neighbors k.
unit	the unit in which mutual information is measured. One of "bit" (log2, default), "ban" (log10) or "nat" (natural units).
priorHyperParam	the prior distribution type for the Bayes estimation. One of "Jeffreys" (default, Jeffreys Prior, Krichevsky and Trofimov Estimator), "BLUnif" (Bayes-Laplace uniform Prior, Holste Estimator), "Perks" (Perks Prior, Schurmann and Grassberger Estimator), or "MiniMax" (MiniMax Prior), can be abbreviated.
shrinkageTarget	shrinkage target frequencies. If not specified (default) it is estimated in a James-Stein-type fashion (uniform distribution).
boot	logical (FALSE as default). Used for calculating a null distribution in order to evaluate if such a interaction is true or obtained by chance.

**Details**

Internal of parMIEstimate.

**Value**

The parEntropyEstimate function returns the value of the entropy of that pair of genes  $H(X, Y)$ .

**Author(s)**

Luciano Garofano <lucianogarofano88@gmail.com>, Stefano Maria Pagnotta, Michele Ceccarelli

**References**

- Paniski L. (2003). Estimation of Entropy and Mutual Information. *Neural Computation*, vol. 15 no. 6 pp. 1191-1253.
- Meyer P.E., Laffitte F., Bontempi G. (2008). minet: A R/Bioconductor Package for Inferring Large Transcriptional Networks Using Mutual Information. *BMC Bioinformatics* 9:461.
- Antos A., Kontoyiannis I. (2001). Convergence properties of functional estimates for discrete distributions. *Random Structures and Algorithms*, vol. 19 pp. 163-193.
- Strong S., Koberle R., de Ruyter van Steveninck R.R., Bialek W. (1998). Entropy and Information in Neural Spike Trains. *Physical Review Letters*, vol. 80 pp. 197-202.

- Miller G.A. (1955). Note on the bias of information estimates. *Information Theory in Psychology*, II-B pp. 95-100.
- Jeffreys H. (1946). An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London*, vol. 186 no. 1007 pp. 453-461.
- Krichevsky R.E., Trofimov V.K. (1981). The performance of universal encoding. *IEEE Transactions on Information Theory*, vol. 27 pp. 199-207.
- Holste D., Hertz H. (1998). Bayes' estimators of generalized entropies. *Journal of Physics A*, vol. 31 pp. 2551-2566.
- Perks W. (1947). Some observations on inverse probability including a new indifference rule. *Journal of the Institute of Actuaries*, vol. 73 pp. 285-334.
- Schurmann T., Grassberg P. (1996). Entropy estimation of symbol sequences. *Chaos*, vol. 6 pp. 414-427.
- Trybula S. (1958). Some problems of simultaneous minimax estimation. *The Annals of Mathematical Statistics*, vol. 29 pp. 245-253.
- Chao A., Shen T.J. (2003). Nonparametric estimation of Shannon's index diversity when there are unseen species. *Environmental and Ecological Statistics*, vol. 10 pp. 429-443.
- James W., Stein C. (1961). Estimation with Quadratic Loss. *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1 pp. 361-379.

### See Also

[parMIEstimate](#), [parMIKD](#)

---

parMIEstimate

*Parallel Mutual Information Estimation*

---

### Description

A function that computes the mutual information between all pairs of rows (or specified ones) of matrix counts using 10 different estimation methods.

### Usage

```
parMIEstimate(counts,
              method = c("ML", "MM", "Bayes", "CS", "Shrink", "KD", "KNN"),
              unit = c("bit", "ban", "nat"), nchips,
              priorHyperParam = c("Jeffreys", "BLUnif", "Perks", "MiniMax"),
              shrinkageTarget, k = 3, tfList = NULL, boot = F)
```

**Arguments**

counts	a numeric matrix (for the reconstruction of gene regulatory networks, genes on rows and samples on columns).
method	a character string indicating which estimate is to be computed. One of "ML" (Maximum Likelihood Estimator, default), "MM" (Miller-Madow corrected Estimator), "Bayes" (Bayesian Estimators), "CS" (Chao-Shen Estimator), "Shrink" (James-Stein shrinkage Estimator), "KD" (kernel Density Estimator), or "KNN" (k-Nearest Neighbor Estimator), can be abbreviated. For the "Bayes" estimate it is needed to specify also which priorHyperParam is to be used; for "Shrink" is optional to specify values for the shrinkageTarget parameter; for "KNN" is needed to specify also the number of nearest neighbors k.
unit	the unit in which mutual information is measured. One of "bit" (log2, default), "ban" (log10) or "nat" (natural units).
nchips	the number of cpu's to be used for making the parallel calculation.
priorHyperParam	the prior distribution type for the Bayes estimation. One of "Jeffreys" (default, Jeffreys Prior, Krichevsky and Trofimov Estimator), "BLUnif" (Bayes-Laplace uniform Prior, Holste Estimator), "Perks" (Perks Prior, Schurmann and Grassberger Estimator), or "MiniMax" (MiniMax Prior), can be abbreviated.
shrinkageTarget	shrinkage target frequencies. If not specified (default) it is estimated in a James-Stein-type fashion (uniform distribution).
k	the number of nearest neighbors to consider for the estimate.
tfList	the character vector specifying which genes from the rownames of the counts matrix is to be used as transcription factor for network reconstruction.
boot	logical (FALSE as default). Used for calculating a null distribution in order to evaluate if such a interaction is true or obtained by chance.

**Value**

The parMIEstimate function returns a square matrix of dimension equal to the number of rows (number of genes) of the counts matrix, or a number of rows equal to the length of tfList.

**Author(s)**

Luciano Garofano <lucianogarofano88@gmail.com>, Stefano Maria Pagnotta, Michele Ceccarelli

**References**

- Paniski L. (2003). Estimation of Entropy and Mutual Information. *Neural Computation*, vol. 15 no. 6 pp. 1191-1253.
- Meyer P.E., Laffitte F., Bontempi G. (2008). minet: A R/Bioconductor Package for Inferring Large Transcriptional Networks Using Mutual Information. *BMC Bioinformatics* 9:461.
- Antos A., Kontoyiannis I. (2001). Convergence properties of functional estimates for discrete distributions. *Random Structures and Algorithms*, vol. 19 pp. 163-193.

- Strong S., Koberle R., de Ruyter van Steveninck R.R., Bialek W. (1998). Entropy and Information in Neural Spike Trains. *Physical Review Letters*, vol. 80 pp. 197-202.
- Miller G.A. (1955). Note on the bias of information estimates. *Information Theory in Psychology*, II-B pp. 95-100.
- Jeffreys H. (1946). An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London*, vol. 186 no. 1007 pp. 453-461.
- Krichevsky R.E., Trofimov V.K. (1981). The performance of universal encoding. *IEEE Transactions on Information Theory*, vol. 27 pp. 199-207.
- Holste D., Hertz H. (1998). Bayes' estimators of generalized entropies. *Journal of Physics A*, vol. 31 pp. 2551-2566.
- Perks W. (1947). Some observations on inverse probability including a new indifference rule. *Journal of the Institute of Actuaries*, vol. 73 pp. 285-334.
- Schurmann T., Grassberg P. (1996). Entropy estimation of symbol sequences. *Chaos*, vol. 6 pp. 414-427.
- Trybula S. (1958). Some problems of simultaneous minimax estimation. *The Annals of Mathematical Statistics*, vol. 29 pp. 245-253.
- Chao A., Shen T.J. (2003). Nonparametric estimation of Shannon's index diversity when there are unseen species. *Environmental and Ecological Statistics*, vol. 10 pp. 429-443.
- James W., Stein C. (1961). Estimation with Quadratic Loss. *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1 pp. 361-379.
- Moon Y., Rajagopalan B., Lall U. (1995). Estimation of mutual information using kernel density estimators. *Physical Review E*, vol. 52 n. 3 pp. 2318-2321.
- Kraskov A., Stogbauer H., Grassberger P. (2004.) Estimating mutual information. *Physical Review E*, vol 69.
- Sales G., Romualdi C. (2011). parmigene - a parallel R package for mutual information estimation and gene network reconstruction. *Bioinformatics*.

## See Also

[parEntropyEstimate](#)

## Examples

```
simData <- simulatedData(p = 5, n = 10, mu = 100, sigma = 0.25,
                        ppower = 0.73, noise = FALSE)
counts <- simData$counts
adjMat <- simData$adjMat

miML <- parMIEstimate(counts, method = "ML", unit = "nat", nchips = 2)
miBJ <- parMIEstimate(counts, method = "Bayes", unit = "nat",
                      nchips = 2, priorHyperParam = "Jeffreys")
miSH <- parMIEstimate(counts, method = "Shrink", unit = "nat",
                      nchips = 2)
miKD <- parMIEstimate(counts, method = "KD", nchips = 2)
miKNN <- parMIEstimate(counts, method = "KNN", unit = "nat", k = 3,
                       nchips = 2)
```



---

parMIKD

*Parallel Kernel Density Mutual Information Estimate*

---

## Description

A function that computes the mutual information between all pairs of rows (or specified ones) of matrix counts using Kernel Density Estimate. It is a wrapper of the functions of the package GenKern.

## Usage

```
parMIKD(idx)
```

## Arguments

`idx` a numeric matrix (for the reconstruction of gene regulatory networks, genes on rows and samples on columns).

## Details

Internal of parMIEstimate.

## Value

The parMIKD function returns the value of the mutual information of that pair of genes  $I(X, Y)$ .

## Author(s)

Luciano Garofano <lucianogarofano88@gmail.com>, Stefano Maria Pagnotta, Michele Ceccarelli

## References

Moon Y., Rajagopalan B., Lall U. (1995). Estimation of mutual information using kernel density estimators. *Physical Review E*, vol. 52 n. 3 pp. 2318-2321.

## See Also

[parEntropyEstimate](#), [parMIEstimate](#)

## Examples

```
simData <- simulatedData(p = 10, n = 50, mu = 100, sigma = 0.25,
                        ppower = 0.73, noise = FALSE)
counts <- simData$counts
adjMat <- simData$adjMat

miKD <- parMIEstimate(counts, method = "KD", nchips = 2)
```

---

performanceIndex	<i>Evaluate Performance Indices</i>
------------------	-------------------------------------

---

**Description**

A function that computes the performance indices (i.e. "Recall", "FPR", "Precision", "Accuracy", "Fscore") between a predicted adjacency matrix (testNet) and a true one (gsNet) for each mutual information threshold.

**Usage**

```
performanceIndex(testNet, gsNet)
```

**Arguments**

testNet	a predicted adjacency matrix to evaluate.
gsNet	the true adjacency matrix.

**Value**

The performanceIndex function returns a data.frame with the considered performance indices for each mutual information threshold.

**Author(s)**

Luciano Garofano <lucianogarofano88@gmail.com>, Stefano Maria Pagnotta, Michele Ceccarelli

**References**

Stehman, S.V. (1997). Selecting and interpreting measures of thematic classification accuracy. *Remote Sensing of Environment* 62 (1): 77-89.

**See Also**

[simulatedData](#), [mainNetFunction](#), [performanceNET](#)

**Examples**

```
simData <- simulatedData(p = 10, n = 20, mu = 100, sigma = 0.25,
                        ppower = 0.73, noise = FALSE)
counts <- simData$counts
adjMat <- simData$adjMat

miML <- parMIEstimate(counts, method = "ML", unit = "nat", nchips = 2)
valML <- performanceIndex(miML, adjMat)
```

---

performanceNET	<i>Evaluate Performance Indices</i>
----------------	-------------------------------------

---

**Description**

A function that computes the performance indices (i.e. "Recall", "FPR", "Precision", "Accuracy", "Fscore") between a predicted adjacency matrix (testNet) and a true one (gsNet) when the false interactions are already removed.

**Usage**

```
performanceNET(predAdjMat, valAdjMat)
```

**Arguments**

predAdjMat	a predicted adjacency matrix to evaluate when the false interactions are already removed.
valAdjMat	the true adjacency matrix.

**Value**

The performanceIndex function returns a data.frame with the considered performance indices.

**Author(s)**

Luciano Garofano <lucianogarofano88@gmail.com>, Stefano Maria Pagnotta, Michele Ceccarelli

**References**

Stehman, S.V. (1997). Selecting and interpreting measures of thematic classification accuracy. *Remote Sensing of Environment* 62 (1): 77-89.

**See Also**

[simulatedData](#), [mainNetFunction](#), [performanceIndex](#)

**Examples**

```
simData <- simulatedData(p = 10, n = 20, mu = 100, sigma = 0.25,
                        ppower = 0.73, noise = FALSE)
counts <- simData$counts
adjMat <- simData$adjMat

miML <- parMIEstimate(counts, method = "ML", unit = "nat", nchips = 2)
miML <- aracne.a(miML, eps = 0.5)

valML <- performanceNET(miML, adjMat)
```

---

plotPR *Plot PR Curve*

---

**Description**

A function that automatically produce a plot of the PR (Precision/Recall) Curve.

**Usage**

```
plotPR(piNet, ...)
```

**Arguments**

piNet            the matrix generated from the performanceIndex function.  
...              other parameters to be added to the plot function.

**Author(s)**

Luciano Garofano <lucianogarofano88@gmail.com>, Stefano Maria Pagnotta, Michele Ceccarelli

**See Also**

[plotROC](#)

**Examples**

```
simData <- simulatedData(p = 10, n = 20, mu = 100, sigma = 0.25,  
                          ppower = 0.73, noise = FALSE)  
counts <- simData$counts  
adjMat <- simData$adjMat  
  
miML <- parMIEstimate(counts, method = "ML", unit = "nat", nchips = 2)  
valML <- performanceIndex(miML, adjMat)  
  
plotPR(valML, col = "red")
```

---

plotROC *Plot ROC Curve*

---

**Description**

A function that automatically produce a plot of the ROC (Receiver Operating Characteristic) Curve.

**Usage**

```
plotROC(piNet, ...)
```

**Arguments**

piNet            the matrix generated from the performanceIndex function.  
 ...            other parameters to be added to the plot function.

**Author(s)**

Luciano Garofano <lucianogarofano88@gmail.com>, Stefano Maria Pagnotta, Michele Ceccarelli

**See Also**

[plotPR](#)

**Examples**

```
simData <- simulatedData(p = 10, n = 20, mu = 100, sigma = 0.25,
                        ppower = 0.73, noise = FALSE)
counts <- simData$counts
adjMat <- simData$adjMat

miML <- parMIEstimate(counts, method = "ML", unit = "nat", nchips = 2)
valML <- performanceIndex(miML, adjMat)

plotROC(valML, col = "red")
```

---

simulatedData

*Random Generation Networks for RNA-Seq Data*

---

**Description**

A function that use a stochastic BA-model for building a graph and the simulated RNA-Seq counts (from a Poisson (multivariate or over-dispersed) distribution) that encode the underlying graph structure.

**Usage**

```
simulatedData(p = 50, n = 100, mu = 100, sigma = 0.25, ppower = 1,
              noise = F, seed = NULL)
```

**Arguments**

p            the number of genes in the networks.  
 n            the number of simulated samples for each gene.  
 mu          the average mean of the simulated Poisson distributions.  
 sigma       the over-dispersed sd value in the case of over-dispersed Poisson simulation.  
 ppower      the power of the preferential attachment for the BA-model.  
 noise       logical. Should some noise be added to the data or not?  
 seed        a single value, interpreted as an integer, in order to control the simulated data.

**Value**

graph	the graph generated with the BA-model.
adjMat	the related adjacency matrix that encodes the underlying graph structure.
counts	the simulated RNA-Seq counts matrix.

**Author(s)**

Luciano Garofano <lucianogarofano88@gmail.com>, Stefano Maria Pagnotta, Michele Ceccarelli

**References**

Barabasi A.L., Albert R. (1999). Emergence of scaling in random networks. *Science*, 286 509-512.  
 Gallopin M., Rau A., Jaffrezic F. (2013). A Hierarchical Poisson Log-Normal Model for Network Inference from RNA Sequencing Data. *PLoSone*.

**See Also**

[mainNetFunction](#)

**Examples**

```
simData <- simulatedData(p = 50, n = 100, mu = 100, sigma = 0.25,
                        ppower = 0.73, noise = FALSE)
plot(simData$graph)

counts <- simData$counts
adjMat <- simData$adjMat
```

---

YoudenIndex

*Youden's Index*

---

**Description**

The Youden's Index evaluates the algorithm's ability to avoid failure.

**Usage**

```
YoudenIndex(resTable)
```

**Arguments**

resTable the data.frame generated from the performanceNET (or one of the row of the matrix generated from the performanceIndex) function.

**Value**

The YoudenIndex function returns the value of the Youden's Index for a given predicted adjacency matrix.

**Author(s)**

Luciano Garofano <lucianogarofano88@gmail.com>, Stefano Maria Pagnotta, Michele Ceccarelli

**References**

Sokolova M., Japkowicz N., Szpakowicz S. (2006). Beyond Accuracy, F-Score and ROC: A Family of Discriminant Measures for Performance Evaluation. *Lecture Notes in Computer Science* Volume 4304, 2006, pp 1015-1021.

**See Also**

[Likelihoods](#), [DiscriminantPower](#)

**Examples**

```
simData <- simulatedData(p = 10, n = 20, mu = 100, sigma = 0.25,
                        ppower = 0.73, noise = FALSE)
counts <- simData$counts
adjMat <- simData$adjMat

miML <- parMIEstimate(counts, method = "ML", unit = "nat", nchips = 2)
miML <- aracne.a(miML, eps = 0.5)

valML <- performanceNET(miML, adjMat)
YI <- YoudenIndex(valML)
```

# Index

- \*Topic **DiscriminantPower**
    - DiscriminantPower, 4
  - \*Topic **Likelihoods**
    - Likelihoods, 10
  - \*Topic **YoudenIndex**
    - YoudenIndex, 22
  - \*Topic **aucDisc**
    - aucDisc, 3
  - \*Topic **beta\_k**
    - entropyBayes, 5
  - \*Topic **entropyBayes**
    - entropyBayes, 5
  - \*Topic **entropyCS**
    - entropyCS, 6
  - \*Topic **entropyML**
    - entropyML, 7
  - \*Topic **entropyMM**
    - entropyMM, 8
  - \*Topic **entropyShrink**
    - entropyShrink, 9
  - \*Topic **mainNetFunction**
    - mainNetFunction, 11
  - \*Topic **package**
    - synRNASeqNet-package, 2
  - \*Topic **parEntropyEstimate**
    - parEntropyEstimate, 12
  - \*Topic **parMIEstimate**
    - parMIEstimate, 14
  - \*Topic **parMIKD**
    - parMIKD, 17
  - \*Topic **performanceIndex**
    - performanceIndex, 18
  - \*Topic **performanceNET**
    - performanceNET, 19
  - \*Topic **plotPR**
    - plotPR, 20
  - \*Topic **plotROC**
    - plotROC, 20
  - \*Topic **shrinkageIntensity**
    - entropyShrink, 9
- \*Topic **simulatedData**
    - simulatedData, 21
  - \*Topic **thetaBayes**
    - entropyBayes, 5
  - \*Topic **thetaGT**
    - entropyCS, 6
  - \*Topic **thetaML**
    - entropyML, 7
  - \*Topic **thetaShrink**
    - entropyShrink, 9
- aucDisc, 2, 3
- DiscriminantPower, 3, 4, 11, 23
- entropyBayes, 2, 5, 7–10
- entropyCS, 2, 6, 6, 8–10
- entropyML, 2, 6, 7, 7, 9, 10
- entropyMM, 2, 6–8, 8, 10
- entropyShrink, 2, 6–9, 9
- Likelihoods, 3, 5, 10, 23
- mainNetFunction, 2, 11, 18, 19, 22
- parEntropyEstimate, 2, 12, 16, 17
- parMIEstimate, 2, 14, 14, 17
- parMIKD, 2, 14, 17
- performanceIndex, 2, 4, 18, 19
- performanceNET, 2, 18, 19
- plotPR, 2, 20, 21
- plotROC, 2, 20, 20
- simulatedData, 2, 12, 18, 19, 21
- synRNASeqNet (synRNASeqNet-package), 2
- synRNASeqNet-package, 2, 2
- YoudenIndex, 3, 5, 11, 22