Package ‘parmigene’

February 20, 2015

Version 1.0.2
Date 2012-07-18

Title  Parallel Mutual Information estimation for Gene Network reconstruction.

Author  Gabriele Sales <gabriele.sales@unipd.it>, Chiara Romualdi
         <chiara.romualdi@unipd.it>

Maintainer  Gabriele Sales <gabriele.sales@unipd.it>

Description  The package provides a parallel estimation of the mutual information based on entropy estimates from k-nearest neighbors distances and algorithms for the reconstruction of gene regulatory networks.

License  AGPL-3

Repository  CRAN

Date/Publication  2012-07-23 19:57:03

NeedsCompilation  yes

R topics documented:

aracne.a .................................................. 2
aracne.m .................................................. 3
clr .................................................. 4
knnmi .................................................. 5
knnmi.all .................................................. 6
knnmi.cross .................................................. 7
mrnet .................................................. 8

Index 10
Algorithm for the Reconstruction of Accurate Cellular Networks

Description

A function that implements the ARACNE algorithm for the reconstruction of gene interaction networks (additive model).

Usage

aracne.a(mi, eps=0.05)

Arguments

mi  matrix of the mutual information.
eps  a positive numeric value used to remove the weakest edge of each triple of nodes.

Details

This algorithm considers each triple of edges independently and removes the weakest one if

\[ MI(i; j) < MI(j; k) - \varepsilon \]

and

\[ MI(i; j) < MI(i; k) - \varepsilon \]

By default, the function uses all the available cores. You can set the actual number of threads used to N by exporting the environment variable OMP_NUM_THREADS=N.

Value

A square weighted adjacency matrix of the inferred network.

References


See Also

aracne.m
clr
mrnet
**Examples**

```r
mat <- matrix(rnorm(1000), nrow=10)
mi <- knnmi.all(mat)
grn <- aracne.a(mi, 0.05)
```

**Description**

A function that implements the ARACNE algorithm for the reconstruction of gene interaction networks (multiplicative model).

**Usage**

```r
aracne.m(mi, tau=0.15)
```

**Arguments**

- `mi`: matrix of the mutual information.
- `tau`: a positive numeric value used to remove the weakest edge of each triple of nodes.

**Details**

This algorithm considers each triple of edges independently and removes the weakest one if

\[
MI(i; j) < MI(j; k) \times (1 - \tau)
\]

and

\[
MI(i; j) < MI(i; k) \times (1 - \tau)
\]

By default, the function uses all the available cores. You can set the actual number of threads used to N by exporting the environment variable `OMP_NUM_THREADS=N`.

**Value**

A square weighted adjacency matrix of the inferred network.

**References**

See Also

aracneNa
clr
mrnet

Examples

mat <- matrix(rnorm(1000), nrow=10)
mi <- knnmi.all(mat)
grn <- aracne.m(mi, 0.15)

clr

Context Likelihood or Relatedness Network

Description

A function that infers the interaction network using the CLR algorithm.

Usage

clr(mi)

Arguments

mi

matrix of the mutual information.

Details

CLR computes the score

$$\sqrt{z_i^2 + z_j^2}$$

for each pair of variables $i, j$, where

$$z_i = \max(0, (I(X_i; X_j) - \text{mean}(X_i))/\text{sd}(X_i))$$

and $\text{mean}(X_i)$ and $\text{sd}(X_i)$ are the mean and the standard deviation of the mutual information values $I(X_i; X_k)$ for all $k = 1, \ldots, n$.

By default, the function uses all the available cores. You can set the actual number of threads used to N by exporting the environment variable OMP_NUM_THREADS=N.

Value

A square weighted adjacency matrix of the inferred network.
References

See Also
aracne.a
aracne.m
mrnet

Examples
mat <- matrix(rnorm(1000), nrow=10)
mi <- knnmi.all(mat)
grn <- clr(mi)

knnmi

Description
A function to perform a parallel estimation of the mutual information of vectors x and y using entropy estimates from K-nearest neighbor distances.

Usage
knnmi(x, y, k=3, noise=1e-12)

Arguments
x a numeric vector.
y a numeric vector with the same length of x.
k the number of nearest neighbors to be considered to estimate the mutual information. Must be less than the number of elements of x.
noise the magnitude of the random noise added to break ties.

Details
The function adds a small random noise to the data in order to break ties due to limited numerical precision.
By default, the function uses all available cores. You can set the actual number of threads used to N by exporting the environment variable OMP_NUM_THREADS=N.
References


See Also

`knnmi.cross`

`knnmi.all`

Examples

```r
x <- rnorm(100)
y <- rnorm(100)
knnmi(x, y, 5)
```

---

**knnmi.all**  
*Parallel Mutual Information Estimation Between All Matrix Rows*

Description

A function that computes the mutual information between all pairs of rows of matrix *mat* using entropy estimates from K-nearest neighbor distances.

Usage

`knnmi.all(mat, k=3, noise=1e-12)`

Arguments

- **mat**: a numeric matrix (for the reconstruction of gene regulatory networks, genes on rows and samples on columns).
- **k**: the number of nearest neighbors to consider to estimate the mutual information. Must be less than the number of columns of *mat*.
- **noise**: the magnitude of the random noise added to break ties.

Details

The function adds a small random noise to the data in order to break ties due to limited numerical precision.

By default, the function uses all the available cores. You can set the actual number of threads used to N by exporting the environment variable `OMP_NUM_THREADS=N`.

References

knnmi.cross

See Also

knnmi
knnmi.cross

Examples

mat <- matrix(rnorm(1000), nrow=10)
knnmi.all(mat, 5)

---

knnmi.cross  Parallel Mutual Information Estimation Between the Rows of Two Matrices

Description

A function that estimates the mutual information between all pairs of rows of matrices mat1 and mat2 using entropy estimates from K-nearest neighbor distances.

Usage

knnmi.cross(mat1, mat2, k=3, noise=1e-12)

Arguments

mat1  a numeric matrix (for the reconstruction of gene regulatory networks, genes on rows and samples on columns).
mat2  a numeric matrix with the same number of columns as mat1.
k  the number of nearest neighbors to consider to estimate the mutual information. Must be less than the number of columns of mat1.
noise  the magnitude of the random noise added to break ties.

Details

The function adds a small random noise to the data in order to break ties due to limited numerical precision.

By default, the function uses all the available cores. You can set the actual number of threads used to N by exporting the environment variable OMP_NUM_THREADS=N.

References


See Also

knnmi
knnmi.all
Examples

```r
mat1 <- matrix(rnorm(1000), nrow=10)
mat2 <- matrix(rnorm(1000), nrow=10)
knnmi.cross(mat1, mat2, 5)
```

---

mrnet  
*Maximum Relevance Minimum Redundancy*

Description

A function that infers the interaction network using the MRNET algorithm.

Usage

```r
mrnet(mi)
```

Arguments

- `mi` matrix of the mutual information.

Details

The MRNET approach starts by selecting the variable $X_i$ having the highest mutual information with the target $Y$.

Then, it repeatedly enlarges the set of selected variables $S$ by taking the $X_k$ that maximizes

$$I(X_k; Y) - \text{mean}(I(X_k; X_i))$$

for all $X_i$ already in $S$.

The procedure stops when the score becomes negative.

By default, the function uses all the available cores. You can set the actual number of threads used to $N$ by exporting the environment variable `OMP_NUM_THREADS=N`.

Value

A square weighted adjacency matrix of the inferred network.

References

See Also

aracne.a
aracne.m
clr

Examples

mat <- matrix(rnorm(1000), nrow=10)
mi <- knnmi.all(mat)
grn <- mrnet(mi)
Index

aracne.a, 2, 4, 5, 9
aracne.m, 2, 3, 5, 9
clr, 2, 4, 4, 9
knnmi, 5, 7
knnmi.all, 6, 6, 7
knnmi.cross, 6, 7, 7
mrnet, 2, 4, 5, 8