Package: parmigene (via r-universe)

October 28, 2026

Version 1.1.1

Date 2024-10-21

Title Parallel Mutual Information Estimation for Gene Network Reconstruction

Description Parallel estimation of the mutual information based on entropy estimates from k-nearest neighbors distances and algorithms for the reconstruction of gene regulatory networks (Sales et al, 2011 <doi:10.1093/bioinformatics/btr274>).

License AGPL-3

URL https://github.com/sales-lab/parmigene

BugReports https://github.com/sales-lab/parmigene/issues

NeedsCompilation yes

Author Gabriele Sales [aut, cre], Chiara Romualdi [aut]

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

Repository https://cran.r-universe.dev

Date/Publication 2024-10-21 13:50:07 UTC

RemoteUrl https://github.com/cran/parmigene

RemoteRef HEAD

RemoteSha d988e7a8740979498029bbb842a8bde5c3b32245

Contents

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

Index **10**

aracne.a *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

Adam A. Margolin, Ilya Nemenman, Katia Basso, Chris Wiggins, Gustavo Stolovitzky, Riccardo Dalla Favera, and Andrea Califano. *Aracne : An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context.* BMC Bioinformatics, 2006.

See Also

aracne.m

clr

mrnet

Examples

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

`aracne.m` *Algorithm for the Reconstruction of Accurate Cellular Networks*

Description

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

Usage

```
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

$$MI(i;j) < MI(j;k) * (1 - \tau$$

and

$$MI(i;j) < MI(i;k) * (1 - \tau)$$

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

Adam A. Margolin, Ilya Nemenman, Katia Basso, Chris Wiggins, Gustavo Stolovitzky, Riccardo Dalla Favera, and Andrea Califano. *Aracne : An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context.* BMC Bioinformatics, 2006.

See Also

aracne.a

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

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

and $mean(X_i)$ and $sd(X_i)$ 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

Jeremiah J. Faith, Boris Hayete, Joshua T. Thaden, Ilaria Mogno, Jamey Wierzbowski, Guillaume Cottarel, Simon Kasif, James J. Collins, and Timothy S. Gardner. *Large-scale mapping and validation of escherichia coli transcriptional regulation from a compendium of expression profiles.* PLoS Biology, 2007.

See Also

aracne.a

aracne.m

mrnet

Examples

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

---

knnmi *Parallel Mutual Information Estimation*

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

Kraskov, Alexander and Stogbauer, Harald and Grassberger, Peter. *Estimating mutual information.* Phys. Rev. E, 2004.

See Also

knnmi.cross

knnmi.all

Examples

```
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

Kraskov, Alexander and Stogbauer, Harald and Grassberger, Peter. *Estimating mutual information.* Phys. Rev. E, 2004.

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

Kraskov, Alexander and Stogbauer, Harald and Grassberger, Peter. *Estimating mutual information.* Phys. Rev. E, 2004.

See Also

knnmi

knnmi.all

Examples

```
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

```
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

$$I(X_k; Y) - mean(I(X_k; X_i))$$

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

H. Peng, F.long and C.Ding. *Feature selection based on mutual information: Criteria of max-dependency, max relevance and min redundancy.* IEEE transaction on Pattern Analysis and Machine Intelligence, 2005.

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