

# Package ‘bioDist’

February 1, 2026

**Title** Different distance measures  
**Version** 1.83.0  
**Author** B. Ding, R. Gentleman and Vincent Carey  
**Description** A collection of software tools for calculating distance measures.  
**Maintainer** Bioconductor Package Maintainer  
<maintainer@bioconductor.org>  
**License** Artistic-2.0  
**Depends** R (>= 2.0), methods, Biobase,KernSmooth  
**Suggests** locfit  
**biocViews** Clustering, Classification  
**git\_url** <https://git.bioconductor.org/packages/bioDist>  
**git\_branch** devel  
**git\_last\_commit** 65ed63e  
**git\_last\_commit\_date** 2025-10-29  
**Repository** Bioconductor 3.23  
**Date/Publication** 2026-02-01

## Contents

closest.top . . . . .	2
cor.dist . . . . .	3
euc . . . . .	4
KLD.matrix . . . . .	5
KLdist.matriX . . . . .	6
man . . . . .	7
mutualInfo . . . . .	8
spearman.dist . . . . .	9
tau.dist . . . . .	10
<b>Index</b>	<b>12</b>

closest.top

*Find the closest genes.*

---

**Description**

Find the closest genes to the supplied target gene based on the supplied distances.

**Usage**

```
closest.top(x, dist.mat, top)
```

**Arguments**

x	the name of the gene (feature) to use.
dist.mat	either a dist object or a matrix of distances.
top	the number of closest genes desired.

**Details**

The feature named x must be in the supplied distances. If so, then the top closest other features are returned.

**Value**

A vector of names of the top closest features.

**Author(s)**

Beiyang Ding

**See Also**

[cor.dist](#), [spearman.dist](#), [tau.dist](#), [euc](#), [man](#), [KLdist.matrix](#), [KLD.matrix](#), [mutualInfo](#)

**Examples**

```
data(sample.ExpressionSet)
sE <- sample.ExpressionSet[1:100,]
d1 <- KLdist.matrix(sE, sample = FALSE)
closest.top(featureNames(sE)[1], d1, 5)
```

---

cor.dist	<i>Pearson correlational distance</i>
----------	---------------------------------------

---

**Description**

Calculate pairwise Pearson correlational distances, i.e. 1-COR or 1-|COR|, and saves as a 'dist' object

**Usage**

```
cor.dist(x, ...)
```

**Arguments**

x	n by p matrix or ExpressionSet; if x is an ExpressionSet, then the function uses its 'exprs' slot.
...	arguments passed to cor.dist: <ul style="list-style-type: none"> <li>• absif TRUE, then 1- COR  else 1-COR, default is TRUE.</li> <li>• diagif TRUE, then the diagonal of the distance matrix will be displayed, default is FALSE.</li> <li>• upperif TRUE, then the upper triangle of the distance matrix will be displayed, default is FALSE.</li> <li>• samplefor objects of classes that extend eSet: if TRUE, then distances are computed between samples(columns) , otherwise, they are computed between features(rows).</li> </ul>

**Details**

The cor function is used to compute the pairwise distances between rows of an input matrix, except if the input is an object of a class that extends eSet and sample is TRUE.

**Value**

Pairwise Pearson correlational distance object

**Author(s)**

Beiyang Ding

**See Also**

[spearman.dist](#), [tau.dist](#), [euc](#), [man](#), [KLdist.matrix](#), [KLD.matrix](#), [mutualInfo](#)

**Examples**

```
x <- matrix(rnorm(200), nrow = 5)
cor.dist(x)
```

euc

*Euclidean distance***Description**

Calculate pairwise Euclidean distances and saves the result as a 'dist' object

**Usage**

```
euc(x, ...)
```

**Arguments**

**x** n by p matrix or an object of a class that extends eSet; if x is a matrix, pairwise distances are calculated between the rows of a matrix. If x is an object of a class that extends eSet, the method makes use of the 'exprs' method and pairwise distances are calculated between samples(columns) if sample is TRUE

**...** arguments passed to euc:

- **diagif** TRUE, then the diagonal of the distance matrix will be displayed; default is FALSE.
- **upperif** TRUE, then the upper triangle of the distance matrix will be displayed; default is FALSE.
- **sampleFor** objects of classes that extends eSet, pairwise distances are calculated between samples(columns) if sample is TRUE ; default value is TRUE

**Details**

The method calculates pairwise euclidean distances, assuming that all samples have the same number of observations

**Value**

An object of class `dist` with the pairwise Euclidean distance between rows except in case of objects of class that extend eSet when sample is TRUE

**Author(s)**

Beiyang Ding

**See Also**

[spearman.dist](#), [tau.dist](#), [man,KLdist.matrix,KLD.matrix,mutualInfo](#)

**Examples**

```
x <- matrix(rnorm(200), nrow = 5)
euc(x)
```

KLD.matrix

*Continuous version of Kullback-Leibler Distance (KLD)***Description**

Calculate KLD by estimating by smoothing  $\log(f(x)/g(x)) * f(x)$  and then integrating.

**Usage**

```
KLD.matrix(x, ...)
```

**Arguments**

- |     |  |
|-----|--|
| x   | n by p matrix or list or an object of a class that extends eSet; if x is an object of a class that extends eSet (eg ExpressionSet), then the function works against its 'exprs' slot.  |
| ... | arguments passed to KLD.matrix: <ul style="list-style-type: none"> <li>• methoduse locfit or density to estimate integrand; default is c("locfit", "density")(i.e. both methods).</li> <li>• suppper and lower limits of the integral; default is NULL in which case the limits of the integral are calculated from the range of the data.</li> <li>• subdivisionssubdivisions for the integration; default is 1000.</li> <li>• diagif TRUE, then the diagonal of the distance matrix will be displayed; default is FALSE.</li> <li>• upperif TRUE, then the upper triangle of the distance matrix will be displayed; default is FALSE.</li> <li>• samplefor ExpressionSet methods: if TRUE, then distances are computed between samples, otherwise, they are computed between genes.</li> </ul> |

**Details**

The distance is computed between rows of the input matrix (except if the input is an object of a class that extends eSet and sample is TRUE).

The presumption is that all samples have the same number of observations. The list method is meant for use when samples sizes are unequal.

**Value**

An object of class dist with the pairwise, between rows, Kullback-Leibler distances.

**Author(s)**

Beiyang Ding, Vincent Carey

**See Also**

[cor.dist](#), [spearman.dist](#), [tau.dist](#), [dist](#), [KLdist.matrix](#), [mutualInfo](#)

## Examples

```
x <- matrix(rnorm(100), nrow = 5)
KLdist.matrix(x, method = "locfit", supp = range(x))
```

---

KLdist.matrix

Discrete version of Kullback-Leibler Distance (KLD)

---

## Description

Calculate the KLD by binning continuous data.

KL distance is calculated using the formula

$$KLD(f_1(x), f_2(x)) = \sum_{i=1}^N f_1(x_i) * \log \frac{f_1(x_i)}{f_2(x_i)}$$

## Usage

```
KLdist.matrix(x, ...)
```

## Arguments

- |     |   |
|-----|---|
| x   | n by p matrix or a list or an object of a class that extends eSet. If x is an object of a class derived from eSet (ExpressionSet, SnpSet etc), then the values returned by the exprs function are used. |
| ... | arguments passed to KLdist.matrix:  |
|     | gridsize the number of grid points used to select the optimal bin width of the histogram used to estimate density. If no value is supplied, the grid size is calculated internally; default is NULL.    |
|     | symmetrize if TRUE, then symmetrize; the default is FALSE.  |
|     | diag if TRUE, then the diagonal of the distance matrix will be displayed; the default is FALSE.   |
|     | upper if TRUE, then the upper triangle of the distance matrix will be displayed; default is FALSE.  |
|     | sample for eSet methods: if TRUE, then the distances are computed between samples, otherwise, between features; the default is TRUE.  |

## Details

The data are binned, and then the KL distance between the two discrete distributions is computed and used. The distance is computed between rows of the input matrix (except if the input is an object of a class that extends eSet and sample is TRUE).

The presumption is that all samples have the same number of observations. The list method is meant for use when samples sizes are unequal.

**Value**

An object of class `dist` is returned.

**Author(s)**

Beiyong Ding

**See Also**

[cor.dist](#), [spearman.dist](#), [tau.dist](#), [euc](#), [man](#), [KLD.matrix](#), [mutualInfo](#)

**Examples**

```
x <- matrix(rnorm(100), nrow = 5)
KLDist.matrix(x, symmetrize = TRUE)
```

---

man	<i>Manhattan distance</i>
-----	---------------------------

---

**Description**

Calculate pairwise Manhattan distances and saves as a `dist` object.

**Usage**

```
man(x, ...)
```

**Arguments**

- |     |  |
|-----|--|
| x   | n by p matrix or an object of class that extends <code>eSet</code> . If x is an object of class that extends <code>eSet</code> , (eg <code>ExpressionSet</code> ) then the function uses its 'exprs' slot.   |
| ... | arguments passed to <code>man</code> : <ul style="list-style-type: none"> <li>• <code>diagif</code> TRUE, then the diagonal of the distance matrix will be displayed; default is FALSE.</li> <li>• <code>upperif</code> TRUE, then the upper triangle of the distance matrix will be displayed; default is FALSE.</li> </ul> |

**Details**

This is just an interface to `dist` with the right parameters set.

**Value**

An instance of the `dist` class with the pairwise Manhattan distances between the rows of x in case of a matrix or between the features (rows) in case of a class that extends `eSet`.

**Author(s)**

Beiyang Ding

**See Also**

[cor.dist](#), [spearman.dist](#), [tau.dist](#), [euc](#), [KLdist.matrix](#), [KLD.matrix](#), [mutualInfo](#)

**Examples**

```
x <- matrix(rnorm(200), nrow = 5)
man(x)
```

---

mutualInfo

*Mutual Information*


---

**Description**

Calculate mutual information via binning

**Usage**

```
mutualInfo(x, ...)
MIdist(x, ...)
```

**Arguments**

- |     |  |
|-----|--|
| x   | an n by p matrix or ExpressionSet; if x is an ExpressionSet, then the function uses its 'exprs' slot.  |
| ... | arguments passed to mutualInfo and MIdist: <ul style="list-style-type: none"> <li>• nbnumber of bins to calculate discrete probabilities; default is 10.</li> <li>• diagif TRUE, then the diagonal of the distance matrix will be displayed; default is FALSE.</li> <li>• upperif TRUE, then the upper triangle of the distance matrix will be displayed; default is FALSE.</li> <li>• samplefor ExpressionSet methods, if TRUE, then distances are computed between samples, otherwise, between genes.</li> </ul> |

**Details**

For mutualInfo each row of x is divided into nbins groups and then the mutual information is computed, treating the data as if they were discrete.

For MIdist we use the transformation proposed by Joe (1989),  $\delta^* = (1 - \exp(-2\delta))^{1/2}$  where  $\delta$  is the mutual information. The MIdist is then  $1 = \delta^*$ . Joe argues that this measure is then similar to Kendall's tau, [tau.dist](#).

**Value**

An object of class `dist` which contains the pairwise distances.

**Author(s)**

Robert Gentleman

**References**

H. Joe, Relative Entropy Measures of Multivariate Dependence, JASA, 1989, 157-164.

**See Also**

[dist](#), [KLdist.matrix](#), [cor.dist](#), [KLD.matrix](#)

**Examples**

```
x <- matrix(rnorm(100), nrow = 5)
mutualInfo(x, nbin = 3)
```

---

spearman.dist	<i>Spearman correlational distance</i>
---------------	--

---

**Description**

Calculate pairwise Spearman correlational distances, i.e. 1-SPEAR or 1-|SPEAR|, for all rows of a matrix and return a `dist` object.

**Usage**

```
spearman.dist(x, ...)
```

**Arguments**

- |                  |  |
|------------------|--|
| <code>x</code>   | n by p matrix or ExpressionSet; if x is an ExpressionSet, then the function uses its 'exprs' slot.   |
| <code>...</code> | arguments passed to <code>spearman.dist</code> : <ul style="list-style-type: none"> <li>• <code>absif</code> TRUE, then 1- SPEAR  else 1-SPEAR; default is TRUE.</li> <li>• <code>diagif</code> TRUE, then the diagonal of the distance matrix will be displayed; default is FALSE.</li> <li>• <code>upperif</code> TRUE, then the upper triangle of the distance matrix will be displayed; default is FALSE.</li> <li>• <code>samplefor</code> the ExpressionSet method: if TRUE (the default), then distances are computed between samples.</li> </ul> |

**Details**

We call `cor` with the appropriate arguments to compute the row-wise correlations.

**Value**

One minus the Spearman correlation, between rows of `x`, are returned, as an instance of the `dist` class.

**Author(s)**

Beiyang Ding

**See Also**

[cor.dist](#), [tau.dist](#), [euc](#), [man](#), [Kldist.matrix](#), [KLD.matrix](#), [mutualInfo](#), [dist](#)

**Examples**

```
x <- matrix(rnorm(200), nrow = 5)
spearman.dist(x)
```

---

tau.dist

*Kendall's tau correlational distance*


---

**Description**

Calculate pairwise Kendall's tau correlational distances, i.e. 1-TAU or 1-|TAU|, for all rows of the input matrix and return an instance of the `dist` class.

**Usage**

```
tau.dist(x, ...)
```

**Arguments**

- |                  |  |
|------------------|--|
| <code>x</code>   | <code>n</code> by <code>p</code> matrix or <code>ExpressionSet</code> ; if <code>x</code> is an <code>ExpressionSet</code> , then the function uses its 'exprs' slot.  |
| <code>...</code> | arguments passed to <code>tau.dist</code> : <ul style="list-style-type: none"> <li>• <code>absif</code> TRUE, then 1- TAU  else 1-TAU; default is TRUE.</li> <li>• <code>diagif</code> TRUE, then the diagonal of the distance matrix will be displayed; default is FALSE.</li> <li>• <code>upperif</code> TRUE, then the upper triangle of the distance matrix will be displayed; default is FALSE.</li> <li>• <code>samplefor</code> the <code>ExpressionSet</code> method: if TRUE (the default), then distances are computed between samples.</li> </ul> |

**Details**

Row-wise correlations are computed by calling the `cor` function with the appropriate arguments.

**Value**

One minus the row-wise Kendall's tau correlations are returned as an instance of the `dist` class. Note that this can be extremely slow for large data sets.

**Author(s)**

Beiyang Ding

**See Also**

[cor.dist](#), [spearman.dist](#), [euc](#), [man](#), [KLdist.matrix](#), [KLD.matrix](#), [mutualInfo](#)

**Examples**

```
x <- matrix(rnorm(200), nrow = 5)
tau.dist(x)
```

# Index

## \* manip

- KLD.matrix, [5](#)
- KLdist.matrixX, [6](#)
- man, [7](#)
- mutualInfo, [8](#)
- spearman.dist, [9](#)
- tau.dist, [10](#)

closest.top, [2](#)

cor.dist, [2](#), [3](#), [5](#), [7–11](#)

cor.dist,eSet-method (cor.dist), [3](#)

cor.dist,matrix-method (cor.dist), [3](#)

dist, [5](#), [9](#), [10](#)

euc, [2](#), [3](#), [4](#), [7](#), [8](#), [10](#), [11](#)

euc,eSet-method (euc), [4](#)

euc,matrix-method (euc), [4](#)

KLD.matrix, [2–4](#), [5](#), [7–11](#)

KLD.matrix,eSet-method (KLD.matrix), [5](#)

KLD.matrix,list-method (KLD.matrix), [5](#)

KLD.matrix,matrix-method (KLD.matrix), [5](#)

KLdist.matrix, [6](#)

KLdist.matrix, [2–5](#), [8–11](#)

KLdist.matrix (KLdist.matrixX), [6](#)

KLdist.matrix,eSet-method (KLdist.matrixX), [6](#)

KLdist.matrix,list-method (KLdist.matrixX), [6](#)

KLdist.matrix,matrix-method (KLdist.matrixX), [6](#)

man, [2–4](#), [7](#), [7](#), [10](#), [11](#)

man,eSet-method (man), [7](#)

man,matrix-method (man), [7](#)

MIdist (mutualInfo), [8](#)

MIdist,ExpressionSet-method (mutualInfo), [8](#)

MIdist,matrix-method (mutualInfo), [8](#)

mutualInfo, [2–5](#), [7](#), [8](#), [8](#), [10](#), [11](#)

mutualInfo,ExpressionSet-method (mutualInfo), [8](#)

mutualInfo,matrix-method (mutualInfo), [8](#)

spearman.dist, [2–5](#), [7](#), [8](#), [9](#), [11](#)

spearman.dist,ExpressionSet-method (spearman.dist), [9](#)

spearman.dist,matrix-method (spearman.dist), [9](#)

tau.dist, [2–5](#), [7](#), [8](#), [10](#), [10](#)

tau.dist,ExpressionSet-method (tau.dist), [10](#)

tau.dist,matrix-method (tau.dist), [10](#)