Package ‘conMItion’

July 14, 2025

Type Package

Title Conditional Mutual Information Estimation for Multi-Omics Data

Version 0.2.0

Description The biases introduced in association measures, particularly mutual information, are influenced by factors such as tumor purity, mutation burden, and hypermethylation. This package provides the estimation of conditional mutual information (CMI) and its statistical significance with a focus on its application to multi-omics data. Utilizing B-spline functions (inspired by Daub et al. (2004) <doi:10.1186/1471-2105-5-118>), the package offers tools to estimate the association between heterogeneous multi-omics data, while removing the effects of confounding factors. This helps to unravel complex biological interactions. In addition, it includes methods to evaluate the statistical significance of these associations, providing a robust framework for multi-omics data integration and analysis. This package is ideal for researchers in computational biology, bioinformatics, and systems biology seeking a comprehensive tool for understanding interdependencies in omics data.

Encoding UTF-8

RoxygenNote 7.3.2

NeedsCompilation yes

Author Gaojianyong Wang [aut, cre]

Maintainer Gaojianyong Wang <`gjywang@gmail.com`>

Repository CRAN

Date/Publication 2025-07-14 17:10:05 UTC

Contents

CMIBiCondimat2mat 2
CMIBiCondimat2matPermu 3
CMIBiCondimat2vec 4
CMIBiCondimat2vecPermu 5
CMImat2mat 6
CMImat2matPermu 7

CMImat2vec 8
CMImat2vecPermu 8
CORmat2vecPermu 9
getCMI 10
getCMIBiCondi 11
getEntropy 12
getEntropyBi 13
getEntropyQuadri 13
getEntropyTri 14
getMI 15
getMIBi 16
getPValue 16
MImat2mat 17
MImat2matPermu 18
MImat2vec 19
MImat2vecPermu 20

Index **21**

`CMIBiCondimat2mat` *Normalized Conditional Mutual Information Between Two Matrices Given Two Conditions*

Description

Computes the normalized conditional mutual information (CMI) between corresponding rows of two matrices, given two condition variables, normalized by their individual information content. CMI is calculated using the specified number of bins and spline order.

Usage

```
CMIBiCondimat2mat(mat1, mat2, condi1, condi2, bin = 6, sp_order = 2)
```

Arguments

`mat1`	A numeric matrix. For example, each row represents a gene and each column represents a sample.
`mat2`	Another numeric matrix to compare against. Must have the same dimensions as ‘mat1‘.
`condi1`	A numeric condition vector, matching the number of columns in ‘mat1‘.
`condi2`	Another numeric condition vector, matching the number of columns in ‘mat1‘.
`bin`	An integer specifying the number of bins. Default is 6.
`sp_order`	An integer specifying the spline order. Must be less than ‘bin‘. Default is 2.

Value

A numeric vector representing the normalized conditional mutual information (CMI) between pairs of rows from ‘mat1‘ and ‘mat2‘, conditioned on ‘condi1‘ and ‘condi2‘.

Examples

```
mat1 <- matrix(rnorm(10000), nrow = 100, ncol = 100)
mat2 <- matrix(rnorm(10000), nrow = 100, ncol = 100)
condi1 <- rnorm(100)
condi2 <- rnorm(100)
CMIBiCondimat2mat(mat1, mat2, condi1, condi2)
```

`CMIBiCondimat2matPermu` *Permuted Conditional Mutual Information Between Two Matrices Given Two Conditions*

Description

Computes the normalized conditional mutual information (CMI) between vectors sampled from two matrices, conditioned on two vectors, normalized by the individual information content. The sampling is done multiple times to generate a distribution.

Usage

```
CMIBiCondimat2matPermu(
  mat1,
  mat2,
  condi1,
  condi2,
  bin = 6,
  sp_order = 2,
  bulkIdx = 0,
  permutationTimes = 1000,
  seedNum = 99999999
)
```

Arguments

`mat1`	A numeric matrix. For example, each row represents a gene and each column represents a sample.
`mat2`	Another numeric matrix to compare against. Must have the same dimensions as ‘mat1‘.
`condi1`	A numeric condition vector, matching the number of columns in ‘mat1‘.
`condi2`	Another numeric condition vector, matching the number of columns in ‘mat‘.
`bin`	An integer specifying the number of bins. Default is 6.
`sp_order`	An integer specifying the spline order. Must be less than ‘bin‘. Default is 2.
`bulkIdx`	Index to divide the task when processing many permutations. Default is 0.
`permutationTimes`	Number of permutations for sampling. Default is 1000.
`seedNum`	Seed for random number generation. Default is 99999999.

Value

A numeric vector of normalized conditional mutual information (CMI) values for each permutation.

Examples

```
mat1 <- matrix(rnorm(10000), nrow = 100, ncol = 100)
mat2 <- matrix(rnorm(10000), nrow = 100, ncol = 100)
condi1 <- rnorm(100)
condi2 <- rnorm(100)
CMIBiCondimat2matPermu(mat1, mat2, condi1, condi2)
```

`CMIBiCondimat2vec` *Normalized Conditional Mutual Information Between Matrix and Vector Given Two Conditions*

Description

Computes the normalized conditional mutual information (CMI) between each row of a matrix and a vector, given two condition vectors, normalized by the mutual information of the vector with itself using the specified bins and spline order.

Usage

```
CMIBiCondimat2vec(mat, vec, condi1, condi2, bin = 6, sp_order = 2)
```

Arguments

`mat`	A numeric matrix. For example, each row represents a gene and each column represents a sample.
`vec`	A numeric vector, with length equal to the number of columns in ‘mat‘.
`condi1`	A numeric condition vector, matching the number of columns in ‘mat‘.
`condi2`	Another numeric condition vector, matching the number of columns in ‘mat‘.
`bin`	An integer specifying the number of bins. Default is 6.
`sp_order`	An integer specifying the spline order. Must be less than ‘bin‘. Default is 2.

Value

A numeric vector representing the normalized conditional mutual information (CMI) between each row of ‘mat‘ and ‘vec‘, given ‘condi1‘ and ‘condi2‘.

Examples

```
mat <- matrix(rnorm(10000), nrow = 100, ncol = 100)
vec <- rnorm(100)
condi1 <- rnorm(100)
condi2 <- rnorm(100)
CMIBiCondimat2vec(mat, vec, condi1, condi2)
```

`CMIBiCondimat2vecPermu`

Permuted Normalized Conditional Mutual Information Between Matrix and Vector Given Two Conditions

---

Description

Computes the conditional mutual information (CMI) between a random vector sampled from a matrix and a vector, conditioned on two vectors, normalized by the mutual information of the vector with itself. The sampling is done multiple times to generate a distribution.

Usage

```
CMIBiCondimat2vecPermu(
  mat,
  vec,
  condi1,
  condi2,
  bin = 6,
  sp_order = 2,
  bulkIdx = 0,
  permutationTimes = 1000,
  seedNum = 99999999
)
```

Arguments

`mat`	A numeric matrix. For example, each row represents a gene and each column represents a sample.
`vec`	A numeric vector, with length equal to the number of columns in ‘mat‘.
`condi1`	A numeric condition vector, matching the number of columns in ‘mat‘.
`condi2`	Another numeric condition vector, matching the number of columns in ‘mat‘.
`bin`	An integer specifying the number of bins. Default is 6.
`sp_order`	An integer specifying the spline order. Must be less than ‘bin‘. Default is 2.
`bulkIdx`	Index to divide the task when processing many permutations. Default is 0.
`permutationTimes`	Number of permutations for sampling. Default is 1000.
`seedNum`	Seed for random number generation. Default is 99999999.

Value

A numeric vector of normalized conditional mutual information (CMI) values for each permutation.

Examples

```
mat <- matrix(rnorm(10000), nrow = 100, ncol = 100)
vec <- rnorm(100)
condi1 <- rnorm(100)
condi2 <- rnorm(100)
CMIBiCondimat2vecPermu(mat, vec, condi1, condi2)
```

`CMImat2mat` *Normalized Conditional Mutual Information Between Two Matrices*

Description

Computes the normalized conditional mutual information (CMI) between corresponding rows of two matrices, given a condition variable, normalized by their individual information content. CMI is calculated using the specified number of bins and spline order.

Usage

```
CMImat2mat(mat1, mat2, condi, bin = 6, sp_order = 2)
```

Arguments

`mat1`	A numeric matrix. For example, each row represents a gene and each column represents a sample.
`mat2`	Another numeric matrix to compare against. Must have the same dimensions as ‘mat1‘.
`condi`	A numeric condition vector, matching the number of columns in ‘mat1‘.
`bin`	An integer specifying the number of bins. Default is 6.
`sp_order`	An integer specifying the spline order. Must be less than ‘bin‘. Default is 2.

Value

A numeric vector representing the normalized conditional mutual information (CMI) between pairs of rows from ‘mat1‘ and ‘mat2‘, conditioned on ‘condi‘.

Examples

```
mat1 <- matrix(rnorm(10000), nrow = 100, ncol = 100)
mat2 <- matrix(rnorm(10000), nrow = 100, ncol = 100)
condi <- rnorm(100)
CMImat2mat(mat1, mat2, condi)
```

`CMImat2matPermu` *Permuted Conditional Mutual Information Between Two Matrices*

Description

Computes the normalized conditional mutual information (CMI) between vectors sampled from two matrices, conditioned on another vector, normalized by the individual information content. The sampling is done multiple times to generate a distribution.

Usage

```
CMImat2matPermu(
  mat1,
  mat2,
  condi,
  bin = 6,
  sp_order = 2,
  bulkIdx = 0,
  permutationTimes = 1000,
  seedNum = 99999999
)
```

Arguments

`mat1`	A numeric matrix. For example, each row represents a gene and each column represents a sample.
`mat2`	Another numeric matrix to compare against. Must have the same dimensions as ‘mat1‘.
`condi`	A numeric condition vector, matching the number of columns in ‘mat1‘.
`bin`	An integer specifying the number of bins. Default is 6.
`sp_order`	An integer specifying the spline order. Must be less than ‘bin‘. Default is 2.
`bulkIdx`	Index to divide the task when processing many permutations. Default is 0.
`permutationTimes`	Number of permutations for sampling. Default is 1000.
`seedNum`	Seed for random number generation. Default is 99999999.

Value

A numeric vector of normalized conditional mutual information (CMI) values for each permutation.

Examples

```
mat1 <- matrix(rnorm(10000), nrow = 100, ncol = 100)
mat2 <- matrix(rnorm(10000), nrow = 100, ncol = 100)
condi <- rnorm(100)
CMImat2matPermu(mat1, mat2, condi)
```

`CMImat2vec` *Normalized Conditional Mutual Information Between Matrix and Vector*

Description

Computes the normalized conditional mutual information (CMI) between each row of a matrix and a vector, given a third condition vector, normalized by the mutual information of the vector with itself using the specified bins and spline order.

Usage

```
CMImat2vec(mat, vec, condi, bin = 6, sp_order = 2)
```

Arguments

`mat`	A numeric matrix. For example, each row represents a gene and each column represents a sample.
`vec`	A numeric vector, with length equal to the number of columns in ‘mat‘.
`condi`	A numeric condition vector, matching the number of columns in ‘mat‘.
`bin`	An integer specifying the number of bins. Default is 6.
`sp_order`	An integer specifying the spline order. Must be less than ‘bin‘. Default is 2.

Value

A numeric vector representing the normalized conditional mutual information (CMI) between each row of ‘mat‘ and ‘vec‘, given ‘condi‘.

Examples

```
mat <- matrix(rnorm(10000), nrow = 100, ncol = 100)
vec <- rnorm(100)
condi <- rnorm(100)
CMImat2vec(mat, vec, condi)
```

`CMImat2vecPermu` *Permuted Normalized Conditional Mutual Information Between Matrix and Vector*

Description

Computes the conditional mutual information (CMI) between a random vector sampled from a matrix and a vector, conditioned on a third vector, normalized by the mutual information of the vector with itself. The sampling is done multiple times to generate a distribution.

Usage

```
CMImat2vecPermu(
  mat,
  vec,
  condi,
  bin = 6,
  sp_order = 2,
  bulkIdx = 0,
  permutationTimes = 1000,
  seedNum = 99999999
)
```

Arguments

`mat`	A numeric matrix. For example, each row represents a gene and each column represents a sample.
`vec`	A numeric vector, with length equal to the number of columns in ‘mat‘.
`condi`	A numeric condition vector, matching the number of columns in ‘mat‘.
`bin`	An integer specifying the number of bins. Default is 6.
`sp_order`	An integer specifying the spline order. Must be less than ‘bin‘. Default is 2.
`bulkIdx`	Index to divide the task when processing many permutations. Default is 0.
`permutationTimes`	Number of permutations for sampling. Default is 1000.
`seedNum`	Seed for random number generation. Default is 99999999.

Value

A numeric vector of normalized conditional mutual information (CMI) values for each permutation.

Examples

```
mat <- matrix(rnorm(10000), nrow = 100, ncol = 100)
vec <- rnorm(100)
condi <- rnorm(100)
CMImat2vecPermu(mat, vec, condi)
```

`CORmat2vecPermu` *Permuted Correlation Between Matrix and Vector*

Description

Computes the correlation between a randomly sampled vector from a matrix and a given vector. The sampling is done multiple times to generate a distribution.

Usage

```
CORmat2vecPermu(
  mat,
  vec,
  cor_type = "pearson",
  bulkIdx = 0,
  permutationTimes = 1000,
  seedNum = 99999999
)
```

Arguments

`mat`	A numeric matrix. For example, each row represents a gene and each column represents a sample.
`vec`	A numeric vector, with length equal to the number of columns in ‘mat‘.
`cor_type`	Type of correlation to calculate: "Pearson", "Kendall", or "Spearman". Default is "Pearson".
`bulkIdx`	Index to divide the task when processing many permutations. Default is 0.
`permutationTimes`	Number of permutations for sampling. Default is 1000.
`seedNum`	Seed for random number generation. Default is 99999999.

Value

A numeric vector of correlation values for each permutation.

Examples

```
mat <- matrix(rnorm(10000), nrow = 100, ncol = 100)
vec <- rnorm(100)
CORmat2vecPermu(mat, vec)
```

`getCMI` *Calculate Conditional Mutual Information*

Description

Computes the conditional mutual information I(x_1; x_2 | x_3) using the specified number of bins and spline order.

Usage

```
getCMI(x_1, x_2, x_3, bin = 6, sp_order = 2)
```

Arguments

`x_1`	A numeric vector for the first variable.
`x_2`	A numeric vector for the second variable. Must match ‘x_1‘ length.
`x_3`	A numeric vector for the condition variable. Must match ‘x_1‘ length.
`bin`	An integer specifying the number of bins. Default is 6.
`sp_order`	An integer specifying the spline order. Must be less than ‘bin‘. Default is 2.

Value

A numeric value representing the conditional mutual information (CMI).

Examples

```
x_1 <- rnorm(100)
x_2 <- rnorm(100)
x_3 <- rnorm(100)
getCMI(x_1, x_2, x_3)
```

`getCMIBiCondi` *Calculate Bivariate Conditional Mutual Information*

Description

Computes conditional mutual information I(x_1; x_2 | x_3, x_4) using the specified number of bins and spline order.

Usage

```
getCMIBiCondi(x_1, x_2, x_3, x_4, bin = 6, sp_order = 2)
```

Arguments

`x_1`	A numeric vector for the first variable.
`x_2`	A numeric vector for the second variable. Must match ‘x_1‘ length.
`x_3`	A numeric vector for the first condition variable. Must match ‘x_1‘ length.
`x_4`	A numeric vector for the second condition variable. Must match ‘x_1‘ length.
`bin`	An integer specifying the number of bins. Default is 6.
`sp_order`	An integer specifying the spline order. Must be less than ‘bin‘. Default is 2.

Value

A numeric value representing the bivariate conditional mutual information (CMI).

Examples

```
x_1 <- rnorm(100)
x_2 <- rnorm(100)
x_3 <- rnorm(100)
x_4 <- rnorm(100)
getCMIBiCondi(x_1, x_2, x_3, x_4)
```

`getEntropy` *Calculate Univariate Entropy*

Description

This function calculates the univariate entropy of a numeric vector using the specified number of bins and spline order.

Usage

```
getEntropy(x_1, bin = 6, sp_order = 2)
```

Arguments

`x_1`	A numeric vector for the only variable.
`bin`	An integer specifying the number of bins. Default is 6.
`sp_order`	An integer specifying the spline order. Must be less than ‘bin‘. Default is 2.

Value

A numeric value representing the entropy of the vector.

Examples

```
x_1 <- rnorm(100)
getEntropy(x_1)
```

---

getEntropyBi *Calculate Joint Entropy for Two Variables*

Description

Computes the joint entropy of two numeric vectors using the specified number of bins and spline order.

Usage

```
getEntropyBi(x_1, x_2, bin = 6, sp_order = 2)
```

Arguments

`x_1`	A numeric vector for the first variable.
`x_2`	A numeric vector for the second variable. Must be the same length as ‘x_1‘.
`bin`	An integer specifying the number of bins. Default is 6.
`sp_order`	An integer specifying the spline order. Must be less than ‘bin‘. Default is 2.

Value

A numeric value representing the joint entropy of the two vectors.

Examples

```
x_1 <- rnorm(100)
x_2 <- rnorm(100)
getEntropyBi(x_1, x_2)
```

---

getEntropyQuadri *Calculate Joint Entropy for Four Variables*

Description

Computes the joint entropy of four numeric vectors using the specified number of bins and spline order.

Usage

```
getEntropyQuadri(x_1, x_2, x_3, x_4, bin = 6, sp_order = 2)
```

Arguments

`x_1`	A numeric vector for the first variable.
`x_2`	A numeric vector for the second variable. Must be the same length as ‘x_1‘.
`x_3`	A numeric vector for the third variable. Must be the same length as ‘x_1‘.
`x_4`	A numeric vector for the fourth variable. Must be the same length as ‘x_1‘.
`bin`	An integer specifying the number of bins. Default is 6.
`sp_order`	An integer specifying the spline order. Must be less than ‘bin‘. Default is 2.

Value

A numeric value representing the joint entropy of the four vectors.

Examples

```
x_1 <- rnorm(100)
x_2 <- rnorm(100)
x_3 <- rnorm(100)
x_4 <- rnorm(100)
getEntropyQuadri(x_1, x_2, x_3, x_4)
```

`getEntropyTri` *Calculate Joint Entropy for Three Variables*

Description

Computes the joint entropy of three numeric vectors using the specified number of bins and spline order.

Usage

```
getEntropyTri(x_1, x_2, x_3, bin = 6, sp_order = 2)
```

Arguments

`x_1`	A numeric vector for the first variable.
`x_2`	A numeric vector for the second variable. Must be the same length as ‘x_1‘.
`x_3`	A numeric vector for the third variable. Must be the same length as ‘x_1‘.
`bin`	An integer specifying the number of bins. Default is 6.
`sp_order`	An integer specifying the spline order. Must be less than ‘bin‘. Default is 2.

Value

A numeric value representing the joint entropy of the three vectors.

Examples

```
x_1 <- rnorm(100)
x_2 <- rnorm(100)
x_3 <- rnorm(100)
getEntropyTri(x_1, x_2, x_3)
```

`getMI` *Calculate Mutual Information Between Two Vectors*

Description

Computes the mutual information (MI) between two numeric vectors using the specified number of bins and spline order.

Usage

```
getMI(x_1, x_2, bin = 6, sp_order = 2)
```

Arguments

`x_1`	A numeric vector representing the first variable.
`x_2`	A numeric vector representing the second variable. Must be the same length as ‘x_1‘.
`bin`	An integer specifying the number of bins. Default is 6.
`sp_order`	An integer specifying the spline order. Must be less than ‘bin‘. Default is 2.

Value

A numeric value representing the mutual information (MI).

Examples

```
x_1 <- rnorm(100)
x_2 <- rnorm(100)
getMI(x_1, x_2)
```

`getMIBi` *Calculate Joint Mutual Information*

Description

Computes the joint mutual information I(x_1, x_2; x_3) using the specified number of bins and spline order.

Usage

```
getMIBi(x_1, x_2, x_3, bin = 6, sp_order = 2)
```

Arguments

`x_1`	A numeric vector for the first variable.
`x_2`	A numeric vector for the second variable. Must match the length of ‘x_1‘.
`x_3`	A numeric vector for the third variable. Must match the length of ‘x_1‘.
`bin`	An integer specifying the number of bins. Default is 6.
`sp_order`	An integer specifying the spline order. Must be less than ‘bin‘. Default is 2.

Value

A numeric value representing joint mutual information (MI).

Examples

```
x_1 <- rnorm(100)
x_2 <- rnorm(100)
x_3 <- rnorm(100)
getMIBi(x_1, x_2, x_3)
```

`getPValue` *Calculate P-Value from a Sorted Distribution*

Description

Computes the P-value for a given numeric value ‘x‘ based on its position within a sorted distribution. This function utilizes a binary search approach for efficient computation.

Usage

```
getPValue(x, sorted_Distri)
```

Arguments

`x`	A numeric value for which the P-value is to be calculated.
`sorted_Distri`	A numeric vector representing a sorted distribution. This distribution must be sorted in ascending order.

Value

A numeric value indicating the P-value, representing the proportion of values in ‘sorted_Distri‘ that are greater than or equal to ‘x‘.

Examples

```
x <- rnorm(1)
sorted_dist <- sort(rnorm(100))
getPValue(x, sorted_dist)
```

`MImat2mat` *Normalized Mutual Information Between Two Matrices*

Description

Computes the normalized mutual information (MI) between corresponding rows of two matrices normalized by their individual information content, using the specified number of bins and spline order.

Usage

```
MImat2mat(mat1, mat2, bin = 6, sp_order = 2)
```

Arguments

`mat1`	A numeric matrix. For example, each row represents a gene and each column represents a sample.
`mat2`	Another numeric matrix to compare against. Must have the same dimensions as ‘mat1‘.
`bin`	An integer specifying the number of bins. Default is 6.
`sp_order`	An integer specifying the spline order. Must be less than ‘bin‘. Default is 2.

Value

A numeric vector where each element corresponds to the normalized mutual information (MI) between respective rows of ‘mat1‘ and ‘mat2‘.

Examples

```
mat1 <- matrix(rnorm(10000), nrow = 100, ncol = 100)
mat2 <- matrix(rnorm(10000), nrow = 100, ncol = 100)
MImat2mat(mat1, mat2)
```

`MImat2matPermu` *Permuted Mutual Information Between Two Matrices*

Description

Computes the normalized mutual information (MI) between vectors sampled from two matrices normalized by the individual information content. The sampling is done multiple times to generate a distribution.

Usage

```
MImat2matPermu(
  mat1,
  mat2,
  bin = 6,
  sp_order = 2,
  bulkIdx = 0,
  permutationTimes = 1000,
  seedNum = 99999999
)
```

Arguments

`mat1`	A numeric matrix. For example, each row represents a gene and each column represents a sample.
`mat2`	Another numeric matrix to compare against. Must have the same dimensions as ‘mat1‘.
`bin`	An integer specifying the number of bins. Default is 6.
`sp_order`	An integer specifying the spline order. Must be less than ‘bin‘. Default is 2.
`bulkIdx`	Index to divide the task when processing many permutations. Default is 0.
`permutationTimes`	Number of permutations for sampling. Default is 1000.
`seedNum`	Seed for random number generation. Default is 99999999.

Value

A numeric vector of normalized mutual information (MI) values for each permutation.

Examples

```
mat1 <- matrix(rnorm(10000), nrow = 100, ncol = 100)
mat2 <- matrix(rnorm(10000), nrow = 100, ncol = 100)
MImat2matPermu(mat1, mat2)
```

`MImat2vec` *Normalized Mutual Information Between Matrix and Vector*

Description

Computes the normalized mutual information (MI) between each row of a matrix and a numeric vector normalized by the mutual information of the vector with itself using the specified number of bins and spline order.

Usage

```
MImat2vec(mat, vec, bin = 6, sp_order = 2)
```

Arguments

`mat`	A numeric matrix. For example, each row represents a gene and each column represents a sample.
`vec`	A numeric vector, with length equal to the number of columns in ‘mat‘.
`bin`	An integer specifying the number of bins. Default is 6.
`sp_order`	An integer specifying the spline order. Must be less than ‘bin‘. Default is 2.

Value

A numeric vector representing the normalized mutual information (MI) between each row of ‘mat‘ and ‘vec‘.

Examples

```
mat <- matrix(rnorm(10000), nrow = 100, ncol = 100)
vec <- rnorm(100)
MImat2vec(mat, vec)
```

`MImat2vecPermu` *Permuted Normalized Mutual Information Between Matrix and Vector*

Description

Computes the mutual information (MI) between a random vector sampled from a matrix and a vector, normalized by the mutual information of the vector with itself. The sampling is done multiple times to generate a distribution.

Usage

```
MImat2vecPermu(
  mat,
  vec,
  bin = 6,
  sp_order = 2,
  bulkIdx = 0,
  permutationTimes = 1000,
  seedNum = 99999999
)
```

Arguments

`mat`	A numeric matrix. For example, each row represents a gene and each column represents a sample.
`vec`	A numeric vector, with length equal to the number of columns in 'mat'.
`bin`	An integer specifying the number of bins. Default is 6.
`sp_order`	An integer specifying the spline order. Must be less than 'bin'. Default is 2.
`bulkIdx`	Index to divide the task when processing many permutations. Default is 0.
`permutationTimes`	Number of permutations for sampling. Default is 1000.
`seedNum`	Seed for random number generation. Default is 99999999.

Value

A numeric vector of normalized mutual information (MI) values for each permutation.

Examples

```
mat <- matrix(rnorm(10000), nrow = 100, ncol = 100)
vec <- rnorm(100)
MImat2vecPermu(mat, vec)
```

Index

CMIBiCondimat2mat, 2
CMIBiCondimat2matPermu, 3
CMIBiCondimat2vec, 4
CMIBiCondimat2vecPermu, 5
CMImat2mat, 6
CMImat2matPermu, 7
CMImat2vec, 8
CMImat2vecPermu, 8
CORmat2vecPermu, 9

getCMI, 10
getCMIBiCondi, 11
getEntropy, 12
getEntropyBi, 13
getEntropyQuadri, 13
getEntropyTri, 14
getMI, 15
getMIBi, 16
getPValue, 16

MImat2mat, 17
MImat2matPermu, 18
MImat2vec, 19
MImat2vecPermu, 20