Test of CompSign

October 24, 2018

`comp_lda` *It computes the LDA for categorical responses (??? need to change? they are actually the predictors)*

Description

It computes the LDA for categorical responses (??? need to change? they are actually the predictors)

Usage

```
comp_lda(x, indices_response)
```

`comp_lm` *It computes a linear regression with some numerical value as the predictor and the compositions as response*

Description

It computes a linear regression with some numerical value as the predictor and the compositions as response

Usage

```
comp_lm(x, indices_predictor)
```

`compare_populations` *Compare two groups*

Description

Compare two groups

Usage

```
compare_populations(predictors, response, ...)
```

`count_matrix<-` *Assign count matrix from object of class sign*

Description

Assign count matrix from object of class sign

Usage

```
count_matrix(object) <- value
```

`count_matrix` *Retrieve count matrix from object of class sign. X is of class sign*

Description

Retrieve count matrix from object of class sign. X is of class sign

Usage

```
count_matrix(X)
```

`createDendrogram` *Create a dendrogram, using Aithchison distance, of the samples in a merged object. The labels are coloured according to the one of the columns of its metadata dataframe (to be specified in name_clinical). There is also an option (bool_comparison) to add a second dendrogram using Euclidean distance, for comparison. Arguments for plot() are inherited (not tested). WARNING! colours for the second dendrogram need fixing*

Description

Create a dendrogram, using Aithchison distance, of the samples in a merged object. The labels are coloured according to the one of the columns of its metadata dataframe (to be specified in name_clinical). There is also an option (bool_comparison) to add a second dendrogram using Euclidean distance, for comparison. Arguments for plot() are inherited (not tested). WARNING! colours for the second dendrogram need fixing

Usage

```
createDendrogram(merged_object, name_clinical, bool_comparison, ...)
```

hello *Hello, World!*

Description

Prints 'Hello, world!'.

Usage

```
hello()
```

Examples

```
hello()
```

merged_compositional_to_sign *Convert a 'sign' object to a 'merged_compositional_to_sign' (note loss of information)*

Description

Convert a 'sign' object to a 'merged_compositional_to_sign' (note loss of information)

Usage

```
merged_compositional_to_sign(x)
```

to_sign *Converts a matrix to a sign object*

Description

Converts a matrix to a sign object

Usage

```
to_sign(x)
```

Index