`Lingua` *Lingua - logic-based data mining on gene expression data*

Description

Logic powered intriguing notions on gene microarray data. Lingua contains applied logic-based methods for microarray data analysis, e.g. for lookup of new notions out of microarray data.

Package:	lingua
Type:	Package
Version:	1.0-1
Date:	2009-04-13
License:	MIT
LazyLoad:	no

Usage

```
lingua.normalize(vect, robust = FALSE)
lingua.unitize(vect, method = "linear", scale = 0)
lingua.pairs(data, mode = "or", by.rows = FALSE)
lingua.triplets(data, mode = "or", by.rows = FALSE)
```

Arguments

`vect`	data vector `vect` to normalize
`robust`	whether to do `robust` normalization
`method`	linerar, normal, exponential `method`
`scale`	forcing the result data density
`data`	data matrix with properties to be combined
`mode`	`or` for max, `and` for min combinations
`by.rows`	having row-wise data properties

Details

Here, we present basic Lingua functions to do data set-up. The main functionality is implemented via Dinorms, Contifiers, and Clustions methods.

Value

`normalize` gives normalized vectors, `unitize` gives vectors set into the [-1,1] interval, `pairs` and `triplets` give data values of connectives-based tuplets.

Note

Look at the Bioplexity web for more information. http://www.bioplexity.org/analysis/

Author(s)

Martin Saturka

Maintainer: Martin Saturka <martin.saturka@bioplexity.org>

References

Bioplexity - www.bioplexity.org

See Also

dinorms, contifiers, clustions

Examples

```
library(lingua)
## data normalization into mean=0, var=1
vect <- rnorm(20, 7, 3)
vect <- lingua.normalize(vect)
## and putting the values into the [0,1] interval
lingua.unitize(vect, "normal", 1)
## combinations of data-matrix columns
data <- matrix(rnorm(20), nrow=5, ncol=4)
lingua.pairs(data, "or")
lingua.triplets(data, "and")
```

Contifiers *Contifiers - Continuous-data association quantifiers*

Description

Contifiers are functors for aggregation of [0,1]-interval valued pairs when we look for relations (associations) between the pair items. Particular value pairs are for data cases, i.e. individual experiments or shopping carts.

Usage

```
contifiers.thresholds(thresh)
contifiers.modes(modes)
contifiers.aggregate(vecx, vecy)
```

Arguments

thresh	either none or [0,1] value for the threshold
modes	either none or name-pair for directionality: one of "mutual", "directional", "inverse", and distance enumeration: one of "minimum", "product", "bold" modes
vecx,vecy	[0,1]-valued data vectors to aggregate

Details

The directional contifiers are for situations like "when `A` occurs then `B` frequently occurs too". It can be e.g. for expression of a gene `b` is triggered by a gene `a`. The mutual contifiers are for situations when the relation is bidirectional.

Value

`thresholds` and `modes` give the set option values, `aggregate` gives the aggregation results.

Note

Look at the Bioplexity and Enduce webs for more information. http://www.bioplexity.org/analysis/ http://www.tangloid.net/enduce/

Author(s)

Martin Saturka

References

Bioplexity www.bioplexity.org

See Also

See Also `lingua`, `dinorms`, `clustions`

Examples

```
library(lingua)
## initial settings
contifiers.thresholds(c(0.1))
contifiers.modes(c("dir", "pro"))
## some random vectors
vecx <- runif(20)
vecy <- runif(20)
## making the aggregation
contifiers.aggregate(vecx, vecy)
```

`Clustions` *Clustions - Notion clustering, based on contifiers*

Description

Clustions are for notion-wise clustering of [0,1]-valued data, possibly with the [-1,1] interval being used as compressed pairs of dichotomic data.

Usage

```
clustions.thresholds(thresh)
clustions.maxcycles(maxcyc)
clustions.cluster(data.cases, ini.centers, by.rows=FALSE)
clustions.cluster2(data.cases, ini.centers, by.rows=FALSE)
```

Arguments

`thresh`	either none or [0,1] value for the threshold
`maxcyc`	either none or the maximal k-means cycling count
`data.cases`	[0,1]-valued data vectors to aggregate
`ini.centers`	initial centers for the clustering
`by.rows`	having row-wise data properties to cluster

Details

The clustering proceeds like k-means with the similarity metrics based on Contifiers, especially the product ones. Cases with zero-only values are neglected, greater similar values count for, greater dissimilar values count against similarities.

Value

`thresholds` and `maxcyc` give the set option values, `cluster` and `cluster2` give the clustering results.

Note

Look at the Bioplexity and Enduce webs for more information. http://www.bioplexity.org/analysis/ http://www.tangloid.net/enduce/

Author(s)

Martin Saturka

References

Bioplexity www.bioplexity.org

See Also

See Also `lingua`, `dinorms`, `contifiers`

Examples

```
library(lingua)
## initial settings
contifiers.thresholds(c(0.1))
clustions.maxcycles(20)
## some random data matrix
data.cases <- matrix(runif(30), nrow=5, ncol=6)
## with some initial centers
ini.centers <- data.cases[,1:3]
## and to cluster it
clustions.cluster(data.cases, ini.centers)
## some [-1,1]-valued data (viewed as compressed pairs)
data.cases2 <- matrix(runif(30, -1, 1), nrow=5, ncol=6)
## some initial centers
ini.cens2neg <- -1 * data.cases2[,1:3]
ini.cens2neg[0.0 > ini.cens2neg] <- 0.0
ini.cens2pos <- data.cases2[,1:3]
ini.cens2pos[0.0 > ini.cens2pos] <- 0.0
ini.centers2 <- cbind(ini.cens2neg, ini.cens2pos)
## and to cluster it
clustions.cluster2(data.cases2, ini.centers2)
## columns were used as objects to cluster
```

`Dinorms` *Dinorms - Natural aggregation of dichotomic values*

Description

Dinorms are functions for aggregation of dichotomic values from the [-1,1] interval. The values are assumed to be supports for two opposite situations. It can be e.g. reasons for cold vs. hot situations, or underexpressed vs. overexpressed genes as reasons for positive vs. negative advices for a treatment.

Usage

```
dinorms.thresholds(thresh = NULL)
dinorms.modes(modes = NULL)
dinorms.aggregate(vect)
dinorms.boost(vect, powers, count, limit = 0)
```

Arguments

`thresh`	either none or vector of two [0,1] values for addition, combination thresholds
`modes`	either none or name-pair for addition: one of "maximum", "coproduct", "summation", and combination: one of "maximum", "codivision", "subtraction", "stabilized" `modes`
`vect`	[-1,1]-valued vector to aggregate
`powers`	[0,1]-valued vector of data strengths

count	number of resamples for the boosting
limit	to use data with >= limit power

Details

The Dinorms aggregation is based on fuzzy logic connectives, separated into twofold process. First, negative and positive values are aggregated separately in the natural way. Then the result values are combined into a final aggregation (e.g. prognosis).

Value

thresholds and modes give the set option values, aggregate gives the sequence of negative, positive, and combined results, boost gives strengthen results, with desc being c(mean, variance).

Note

Look at the Bioplexity and Enduce webs for more information. http://www.bioplexity.org/analysis/ http://www.tangloid.net/enduce/

Author(s)

Martin Saturka

References

Bioplexity www.bioplexity.org

See Also

See Also lingua, contifiers, clustions

Examples

```
library(lingua)
## initial settings
dinorms.thresholds(c(0.1, 0.1))
dinorms.modes(c("cop", "sub"))
## aggregate some random vector
vect <- runif(20, -1, 1)
dinres <- dinorms.aggregate(vect)
## boosting the dinorms
powers <- runif(length(vect))
dinorms.boost(vect, powers, 10, 0.5)
```