

Package ‘TDAvec’

July 30, 2022

Type Package

Title Vector Persistence Block

Version 0.1.0

Author Kit C. Chan, Umar Islambekov, Alexey Luchinsky, Rebecca Sanders

Maintainer Alexey Luchinsky <aluchi@bgsu.edu>

Description

Caclaulates the Vector Persistence Block. See paper <arXiv:2109.08239> for more details.

License GPL (>= 2)

Encoding UTF-8

LazyData true

Imports Rcpp (>= 1.0.9), TDAstats, TDA

LinkingTo Rcpp

Suggests knitr

VignetteBuilder knitr

NeedsCompilation yes

RoxygenNote 7.2.0

R topics documented:

computePI	2
computePL	3
computePS	3
computeVAB	4
computeVPB	4
rcpp_hello	5
Index	6

computePI

*Calculates the Persistence Image***Description**

Calculates the Persistence Image

Usage

```
computePI(D, homDim, res, sigma, minB, maxB, minP, maxP)
```

Arguments

D	N by 3 matrix (columns contain dimension, birth and persistence values respectively)
homDim	homological dimension (0 for H0, 1 for H1, etc.)
res	resolution parameter
sigma	sigma parameter
minB	minimal birth value
maxB	maximal birth value
minP	minimal persistance value
maxP	maxzimal persistence value

Examples

```
N <- 100
set.seed(123)
X <- TDA::circleUnif(N) + rnorm(2*N, mean = 0, sd = 0.2)
D <- TDA::ripsDiag(X, maxdimension = 1, maxscale = 2)$diagram
D[,3] <- D[,3] - D[,2]
colnames(D)[3] <- "Persistence"
res <- 5 # resolution or grid size

minPH0 <- min(D[D[,1]==0,3]); maxPH0 <- max(D[D[,1]==0,3])
sigma <- 0.5*(maxPH0-minPH0)/res
computePI(D, homDim=0, res, sigma, minB=NA, maxB=NA, minPH0, maxPH0)

minBH1 <- min(D[D[,1]==1,2]); maxBH1 <- max(D[D[,1]==1,2])
minPH1 <- min(D[D[,1]==1,3]); maxPH1 <- max(D[D[,1]==1,3])
sigma <- 0.5*(maxPH1-minPH1)/res # default way of selecting sigma - can be overridden
computePI(D, homDim=1, res, sigma, minBH1, maxBH1, minPH1, maxPH1)
```

computePL	<i>Persistence Landscape</i>
-----------	------------------------------

Description

Calculates the Persistence Landscape

Usage

```
computePL(D, homDim, k, scaleSeq)
```

Arguments

D	N by 3 matrix (columns contain dimension, birth and persistence values respectively)
homDim	homological dimension (0 for H0, 1 for H1, etc.)
k	order of landscape function
scaleSeq	sequence of scale values for vectorization

Examples

```
N <- 100
set.seed(123)
X <- TDA::circleUnif(N) + rnorm(2*N, mean = 0, sd = 0.2)
D <- TDA::ripsDiag(X, maxdimension = 1, maxscale = 2)$diagram
scaleSeq = seq(0, 2, length.out=11) # sequence of scale values
computePL(D, homDim=0, k=1, scaleSeq)
```

computePS	<i>Persistence Landscape</i>
-----------	------------------------------

Description

Calculates the Persistence Silhouettes

Usage

```
computePS(D, homDim, p, scaleSeq)
```

Arguments

D	N by 3 matrix (columns contain dimension, birth and persistence values respectively)
homDim	homological dimension (0 for H0, 1 for H1, etc.)
p	power of the weights for the silhouette function
scaleSeq	sequence of scale values for vectorization

Examples

```

N <- 100
set.seed(123)
X <- TDA::circleUnif(N) + rnorm(2*N, mean = 0, sd = 0.2)
D <- TDA::ripsDiag(X, maxdimension = 1, maxscale = 2)$diagram
scaleSeq = seq(0, 2, length.out=11) # sequence of scale values
computePS(D, homDim=0, p=1, scaleSeq)

```

computeVAB

*Vector Persistence Blocks***Description**

Calculates the Vector Persistence Block

Usage

```
computeVAB(D, homDim, scaleSeq)
```

Arguments

D	N by 3 matrix (columns contain dimension, birth and persistence values respectively)
homDim	homological dimension (0 for H0, 1 for H1, etc.)
scaleSeq	sequence of scale values for vectorization

Examples

```

N <- 100
set.seed(123)
X <- TDA::circleUnif(N) + rnorm(2*N, mean = 0, sd = 0.2)
D <- TDA::ripsDiag(X, maxdimension = 1, maxscale = 2)$diagram
scaleSeq = seq(0, 2, length.out=11)
computeVAB(D, homDim = 0, scaleSeq)
computeVAB(D, homDim = 1, scaleSeq)

```

computeVPB

*Vector Persistence Blocks***Description**

Calculates the Vector Persistence Block

Usage

```
computeVPB(D, homDim, xSeq, ySeq, tau)
```

Arguments

D	N by 3 matrix (columns contain dimension, birth and persistence values respectively)
homDim	homological dimension (0 for H0, 1 for H1, etc.)
xSeq	sequence of x (birth) values of the grid vertices
ySeq	sequence of y (persistence) values of the grid vertices
tau	parameter (between 0 and 1) controlling block width. weight function $w(x,y) = x+y$

Examples

```
X <- TDAstats::circle2d + rnorm(200,mean = 0,sd = 0.1)
D <- TDAstats::calculate_homology(X,dim = 1,threshold = 2)
D[,3] <- D[,3] - D[,2]
xSeq <- unique(quantile(D[D[,1]==1,2],probs = seq(0,1,by=0.1)))
ySeq <- unique(quantile(D[D[,1]==1,3],probs = seq(0,1,by=0.1)))
computeVPB(D,homDim = 0, xSeq=xSeq, ySeq,tau = 0.5)
```

rcpp_hello

*Hello, Rcpp!***Description**

Returns an R list containing the character vector `c("foo", "bar")` and the numeric vector `c(0,1)`.

Usage

```
rcpp_hello()
```

Examples

```
rcpp_hello()
```

Index

computePI, [2](#)
computePL, [3](#)
computePS, [3](#)
computeVAB, [4](#)
computeVPB, [4](#)
rcpp_hello, [5](#)