Package 'gSeg'

September 24, 2020

Version 1.0

Date 2020-9-21

Title Graph-Based Change-Point Detection (g-Segmentation)

Author Hao Chen, Nancy R. Zhang, Lynna Chu, and Hoseung Song

Maintainer Hao Chen <hxchen@ucdavis.edu>

Depends R (>= 3.0.1)

Suggests ade4

Description Using an approach based on similarity graph to estimate change-point(s) and the corresponding p-values. Can be applied to any type of data (high-dimensional, non-Euclidean, etc.) as long as a reasonable similarity measure is available.

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2020-09-24 09:10:12 UTC

R topics documented:

E1 *An edge matrix representing a similarity graph*

Description

This is the variable name for an edge matrix in the "Example" data. It is constructed from a sequence of n=200 observations with a change in mean at t = 100. E1 is a matrix with the number of rows the number of edges in the similarity graph and 2 columns. Each row contains the node indices of an edge.

E2 *An edge matrix representing a similarity graph*

Description

This is the variable name for an edge matrix in the "Example" data. It is constructed from a sequence of n=200 observations with a change in mean starting at t=45. E2 is a matrix with the number of rows the number of edges in the similarity graph and 2 columns. Each row contains the node indices of an edge.

E3 *An edge matrix representing a similarity graph*

Description

This is the variable name for an edge matrix in the "Example" data. It is constructed from a sequence of n=200 observations with a change in mean and variance starting at t = 145. E3 is a matrix with the number of rows the number of edges in the similarity graph and 2 columns. Each row contains the node indices of an edge.

E4 *An edge matrix representing a similiarity graph*

Description

This is the variable name for an edge matrix in the "Example" data. It is constructed from a sequence of n=200 observations with a change in mean and variance starting at t=50. E4 is a matrix with the number of rows the number of edges in the similarity graph and 2 columns. Each row contains the node indices of an edge.

E5   *An edge matrix representing a similiarity graph*

Description

This is the variable name for an edge matrix in the "Example" data. It is constructed from a sequence of n=200 observations with a change in mean on the interval t= 155 to t=185. E5 is a matrix with the number of rows the number of edges in the similarity graph and 2 columns. Each row contains the node indices of an edge.

---

gSeg   *Graph-Based Change-Point Detection*

Description

This package can be used to estimate change-points in a sequence of observations, where the observation can be a vector or a data object, e.g., a network. A similarity graph is required. It can be a minimum spanning tree, a minimum distance pairing, a nearest neighbor graph, or a graph based on domain knowledge.

For sequence with no repeated observations, if you believe the sequence has at most one change point, the function gseg1 should be used; if you believe an interval of the sequence has a changed distribution, the function gseg2 should be used. If you feel the sequence has multiple change-points, you can use gseg1 and gseg2 multiple times. See gseg1 and gseg2 for the details of these two function.

If you believe the sequence has repeated observations, the function gseg1_discrete should be used for single change-point. For a changed interval of the sequence, the function gseg2_discrete should be used. The function nnl can be used to construct the nearest neighbor link.

Author(s)

Hao Chen, Nancy R. Zhang, Lynna Chu, and Hoseung Song

Maintainer: Hao Chen (hxchen@ucdavis.edu)

References

Chen, Hao, and Nancy Zhang. (2015). Graph-based change-point detection. The Annals of Statistics, 43(1), 139-176.

Chu, Lynna, and Hao Chen. (2019). Asymptotic distribution-free change-point detection for modern data. The Annals of Statistics, 47(1), 382-414.

Song, Hoseung, and Hao Chen (2020). Asymptotic distribution-free change-point detection for data with repeated observations. arXiv:2006.10305

See Also

gseg1, gseg2, gseg1_discrete, gseg2_discrete, nnl

Examples

```
data(Example)
# Five examples, each example is a n-length sequence.
# Ei (i=1,...,5): an edge matrix representing a similarity graph constructed on the
# observations in the ith sequence.
# The following code shows how the Ei's were constructed.

require(ade4)
# For illustration, we use 'mstree' in this package to construct the similarity graph.
# You can use other ways to construct the graph.

## Sequence 1: change in mean in the middle of the sequence.
d = 50
mu = 2
tau = 100
n = 200
set.seed(500)
y = rbind(matrix(rnorm(d*tau),tau), matrix(rnorm(d*(n-tau),mu/sqrt(d)), n-tau))
y.dist = dist(y)
E1 = mstree(y.dist)
# For illustration, we constructed the minimum spanning tree.
# You can use other ways to construct the graph.

r1 = gseg1(n,E1, statistics="all")
# output results based on all four statistics
# the scan statistics can be found in r1$scanZ

r1_a = gseg1(n,E1, statistics="w")
# output results based on the weighted edge-count statistic

r1_b = gseg1(n,E1, statistics=c("w","g"))
# output results based on the weighted edge-count statistic
# and generalized edge-count statistic

# image(as.matrix(y.dist))
# run this if you would like to have some idea on the pairwise distance

## Sequence 2: change in mean away from the middle of the sequence.
d = 50
mu = 2
tau = 45
n = 200
set.seed(500)
y = rbind(matrix(rnorm(d*tau),tau), matrix(rnorm(d*(n-tau),mu/sqrt(d)), n-tau))
y.dist = dist(y)
E2 = mstree(y.dist)
r2 = gseg1(n,E2,statistic="all")
# image(as.matrix(y.dist))


## Sequence 3: change in both mean and variance away from the middle of the sequence.
```

```
d = 50
mu = 2
sigma=0.7
tau = 145
n = 200
set.seed(500)
y = rbind(matrix(rnorm(d*tau),tau), matrix(rnorm(d*(n-tau),mu/sqrt(d),sigma), n-tau))
y.dist = dist(y)
E3 = mstree(y.dist)
r3=gseg1(n,E3,statistic="all")
# image(as.matrix(y.dist))


## Sequence 4: change in both mean and variance away from the middle of the sequence.
d = 50
mu = 2
sigma=1.2
tau = 50
n = 200
set.seed(500)
y = rbind(matrix(rnorm(d*tau),tau), matrix(rnorm(d*(n-tau),mu/sqrt(d),sigma), n-tau))
y.dist = dist(y)
E4 = mstree(y.dist)
r4=gseg1(n,E4,statistic="all")
# image(as.matrix(y.dist))


## Sequence 5: change in both mean and variance happens on an interval.
d = 50
mu = 2
sigma=0.5
tau1 = 155
tau2 = 185
n = 200
set.seed(500)
y1 = matrix(rnorm(d*tau1),tau1)
y2 = matrix(rnorm(d*(tau2-tau1),mu/sqrt(d),sigma), tau2-tau1)
y3 = matrix(rnorm(d*(n-tau2)), n-tau2)
y = rbind(y1, y2, y3)
y.dist = dist(y)
E5 = mstree(y.dist)
r5=gseg2(n,E5,statistics="all")
# image(as.matrix(y.dist))

## Sequence 6: change in mean away from the middle of the sequence
## when data has repeated observations.
d = 50
mu = 2
tau = 100
n = 200
set.seed(500)
y1_temp = matrix(rnorm(d*tau),tau)
sam1 = sample(1:tau, replace = TRUE)
```

```
y1 = y1_temp[sam1,]
y2_temp = matrix(rnorm(d*(n-tau),mu/sqrt(d)), n-tau)
sam2 = sample(1:tau, replace = TRUE)
y2 = y2_temp[sam2,]
y = rbind(y1, y2)
# Data y has repeated observations
y_uni = unique(y)
E6 = nnl(dist(y_uni), 1)
cha = do.call(paste, as.data.frame(y))
id = match(cha, unique(cha))
r6 = gseg1_discrete(n, E6, id, statistics="all")
# image(as.matrix(y.dist))
```

---

gseg1   *Graph-Based Change-Point Detection for Single Change-Point*

Description

This function finds a break point in the sequence where the underlying distribution changes. It provides four graph-based test statistics.

Usage

```
gseg1(n, E, statistics=c("all","o","w","g","m"), n0=0.05*n, n1=0.95*n, pval.appr=TRUE,
 skew.corr=TRUE, pval.perm=FALSE, B=100)
```

Arguments

n	The number of observations in the sequence.
E	The edge matrix (a "number of edges" by 2 matrix) for the similarity graph. Each row contains the node indices of an edge.
statistics	The scan statistic to be computed. A character indicating the type of of scan statistic desired. The default is "all".
	"all": specifies to compute **a**ll of the scan statistics: original, weighted, generalized, and max-type;
	"o","ori" or "original": specifies the **o**riginal edge-count scan statistic;
	"w" or "weighted": specifies the **w**eighted edge-count scan statistic;
	"g" or "generalized": specifies the **g**eneralized edge-count scan statistic; and
	"m" or "max": specifies the **m**ax-type edge-count scan statistic.
n0	The starting index to be considered as a candidate for the change-point.
n1	The ending index to be considered as a candidate for the change-point.
pval.appr	If it is TRUE, the function outputs p-value approximation based on asymptotic properties.
skew.corr	This argument is useful only when pval.appr=TRUE. If skew.corr is TRUE, the p-value approximation would incorporate skewness correction.

pval.perm	If it is TRUE, the function outputs p-value from doing B permutations, where B is another argument that you can specify. Doing permutation could be time consuming, so use this argument with caution as it may take a long time to finish the permutation.
B	This argument is useful only when pval.perm=TRUE. The default value for B is 100.

Value

Returns a list scanZ with tauhat, Zmax, and a vector of the scan statistics for each type of scan statistic specified. See below for more details.

tauhat	An estimate of the location of the change-point.
Zmax	The test statistic (maximum of the scan statistics).
Z	A vector of the original scan statistics (standardized counts) if statistic specified is "all" or "o".
Zw	A vector of the weighted scan statistics (standardized counts) if statistic specified is "all" or "w".
S	A vector of the generalized scan statistics (standardized counts) if statistic specified is "all" or "g".
M	A vector of the max-type scan statistics (standardized counts) if statistic specified is "all" or "m".
R	A vector of raw counts of the original scan statistic. This output only exists if the statistic specified is "all" or "o".
Rw	A vector of raw counts of the weighted scan statistic. This output only exists if statistic specified is "all" or "w".
pval.appr	The approximated p-value based on asymptotic theory for each type of statistic specified.
pval.perm	This output exists only when the argument pval.perm is TRUE . It is the permutation p-value from B permutations and appears for each type of statistic specified (same for perm.curve, perm.maxZs, and perm.Z).
perm.curve	A B by 2 matrix with the first column being critical values corresponding to the p-values in the second column.
perm.maxZs	A sorted vector recording the test statistics in the B permutaitons.
perm.Z	A B by n matrix with each row being the scan statistics from each permutaiton run.

See Also

[gSeg](), [gseg2](), [gseg1_discrete]()

Examples

```
data(Example)
# Five examples, each example is a n-length sequence.
# Ei (i=1,...,5): an edge matrix representing a similarity graph constructed on the
# observations in the ith sequence.
# Check '?gSeg' to see how the Ei's were constructed.


## E1 is an edge matrix representing a similarity graph.
# It is constructed on a sequence of length n=200 with a change in mean
# in the middle of the sequence (tau = 100).

r1 = gseg1(n,E1, statistics="all")
# output results based on all four statistics
# the scan statistics can be found in r1$scanZ
r1_a = gseg1(n,E1, statistics="w")
# output results based on the weighted edge-count statistic
r1_b = gseg1(n,E1, statistics=c("w","g"))
# output results based on the weighted edge-count statistic
# and generalized edge-count statistic


## E2 is an edge matrix representing a similarity graph.
# It is constructed on a sequence of length n=200 with a change in mean
# away from the middle of the sequence (tau=45).
r2 = gseg1(n,E2,statistic="all")


## E3 is an edge matrix representing a similarity graph.
# It is constructed on a sequence of length n=200 with a change in both mean
# and variance away from the middle of the sequence (tau = 145).
r3=gseg1(n,E3,statistic="all")


## E4 is an edge matrix representing a similarity graph.
# It is constructed on a sequence of length n=200 with a change in both mean
# and variance away from the middle of the sequence (tau = 50).
r4=gseg1(n,E4,statistic="all")
```

gseg1_discrete	*Graph-Based Change-Point Detection for Single Change-Point for Data with Repeated Observations*

Description

This function finds a break point in the sequence where the underlying distribution changes when data has repeated observations. It provides four graph-based test statistics.

Usage

```
gseg1_discrete(n, E, id, statistics=c("all","o","w","g","m"), n0=0.05*n, n1=0.95*n,
    pval.appr=TRUE, skew.corr=TRUE, pval.perm=FALSE, B=100)
```

Arguments

n	The number of observations in the sequence.
E	The edge matrix (a "number of edges" by 2 matrix) for the similarity graph. Each row contains the node indices of an edge.
id	The index of observations (order of observations).
statistics	The scan statistic to be computed. A character indicating the type of of scan statistic desired. The default is "all".
	"all": specifies to compute **a**ll of the scan statistics: original, weighted, generalized, and max-type;
	"o","ori" or "original": specifies the **o**riginal edge-count scan statistic;
	"w" or "weighted": specifies the **w**eighted edge-count scan statistic;
	"g" or "generalized": specifies the **g**eneralized edge-count scan statistic; and
	"m" or "max": specifies the **m**ax-type edge-count scan statistic.
n0	The starting index to be considered as a candidate for the change-point.
n1	The ending index to be considered as a candidate for the change-point.
pval.appr	If it is TRUE, the function outputs p-value approximation based on asymptotic properties.
skew.corr	This argument is useful only when pval.appr=TRUE. If skew.corr is TRUE, the p-value approximation would incorporate skewness correction.
pval.perm	If it is TRUE, the function outputs p-value from doing B permutations, where B is another argument that you can specify. Doing permutation could be time consuming, so use this argument with caution as it may take a long time to finish the permutation.
B	This argument is useful only when pval.perm=TRUE. The default value for B is 100.

Value

Returns a list scanZ with tauhat, Zmax, and a vector of the scan statistics for each type of scan statistic specified. See below for more details.

tauhat_a	An estimate of the location of the change-point for averaging approach.
tauhat_u	An estimate of the location of the change-point for union approach.
Z_a_max	The test statistic (maximum of the scan statistics) for averaging approach.
Z_u_max	The test statistic (maximum of the scan statistics) for union approach.
Zo_a	A vector of the original scan statistics (standardized counts) for averaging approach if statistic specified is "all" or "o".
Zo_u	A vector of the original scan statistics (standardized counts) for union approach if statistic specified is "all" or "o".

Zw_a	A vector of the weighted scan statistics (standardized counts) for averaging approach if statistic specified is "all" or "w".
Zw_u	A vector of the weighted scan statistics (standardized counts) for union approach if statistic specified is "all" or "w".
S_a	A vector of the generalized scan statistics (standardized counts) for averaging approach if statistic specified is "all" or "g".
S_u	A vector of the generalized scan statistics (standardized counts) for union approach if statistic specified is "all" or "g".
M_a	A vector of the max-type scan statistics (standardized counts) for averaging appraoch if statistic specified is "all" or "m".
M_u	A vector of the max-type scan statistics (standardized counts) for union appraoch if statistic specified is "all" or "m".
Ro_a	A vector of raw counts of the original scan statistic for averaging approach. This output only exists if the statistic specified is "all" or "o".
Ro_u	A vector of raw counts of the original scan statistic for union approach. This output only exists if the statistic specified is "all" or "o".
Rw_a	A vector of raw counts of the weighted scan statistic for averaging appraoch. This output only exists if statistic specified is "all" or "w".
Rw_u	A vector of raw counts of the weighted scan statistic for union appraoch. This output only exists if statistic specified is "all" or "w".
pval.appr	The approximated p-value based on asymptotic theory for each type of statistic specified.
pval.perm	This output exists only when the argument pval.perm is TRUE . It is the permutation p-value from B permutations and appears for each type of statistic specified (same for perm.curve, perm.maxZs, and perm.Z).
perm.curve	A B by 2 matrix with the first column being critical values corresponding to the p-values in the second column.
perm.maxZs	A sorted vector recording the test statistics in the B permutaitons.
perm.Z	A B by n matrix with each row being the scan statistics from each permutaiton run.

See Also

gSeg, gseg1, nnl, gseg2_discrete

Examples

```
d = 50
mu = 2
tau = 100
n = 200

set.seed(500)
y1_temp = matrix(rnorm(d*tau),tau)
sam1 = sample(1:tau, replace = TRUE)
```

```
y1 = y1_temp[sam1,]
y2_temp = matrix(rnorm(d*(n-tau),mu/sqrt(d)), n-tau)
sam2 = sample(1:tau, replace = TRUE)
y2 = y2_temp[sam2,]

y = rbind(y1, y2)

# This data y has repeated observations
y_uni = unique(y)
E = nnl(dist(y_uni), 1)

cha = do.call(paste, as.data.frame(y))
id = match(cha, unique(cha))

r1 = gseg1_discrete(n, E, id, statistics="all")
# output results based on all four statistics
# the scan statistics can be found in r1$scanZ
r1_a = gseg1_discrete(n, E, id, statistics="w")
# output results based on the weighted edge-count statistic
r1_b = gseg1_discrete(n, E, id, statistics=c("w","g"))
# output results based on the weighted edge-count statistic
# and generalized edge-count statistic
```

---

gseg2   *Graph-Based Change-Point Detection for Changed Interval*

Description

This function finds an interval in the sequence where their underlying distribution differs from the rest of the sequence. It provides four graph-based test statistics.

Usage

```
gseg2(n, E, statistics=c("all","o","w","g","m"), l0=0.05*n, l1=0.95*n, pval.appr=TRUE,
  skew.corr=TRUE, pval.perm=FALSE, B=100)
```

Arguments

n	The number of observations in the sequence.
E	The edge matrix (a "number of edges" by 2 matrix) for the similarity graph. Each row contains the node indices of an edge.
statistics	The scan statistic to be computed. A character indicating the type of of scan statistic desired. The default is "all".
	"all": specifies to compute **a**ll of the scan statistics: original, weighted, generalized, and max-type;
	"o","ori" or "original": specifies the **o**riginal edge-count scan statistic;
	"w" or "weighted": specifies the **w**eighted edge-count scan statistic;
	"g" or "generalized": specifies the **g**eneralized edge-count scan statistic; and
	"m" or "max": specifies the **m**ax-type edge-count scan statistic.

l0	The minimum length of the interval to be considered as a changed interval.
l1	The maximum length of the interval to be considered as a changed interval.
pval.appr	If it is TRUE, the function outputs p-value approximation based on asymptotic properties.
skew.corr	This argument is useful only when pval.appr=TRUE. If skew.corr is TRUE, the p-value approximation would incorporate skewness correction.
pval.perm	If it is TRUE, the function outputs p-value from doing B permutations, where B is another argument that you can specify. Doing permutation could be time consuming, so use this argument with caution as it may take a long time to finish the permutation.
B	This argument is useful only when pval.perm=TRUE. The default value for B is 100.

Value

Returns a list scanZ with tauhat, Zmax, and a matrix of the scan statistics for each type of scan statistic specified. See below for more details.

tauhat	An estimate of the two ends of the changed interval.
Zmax	The test statistic (maximum of the scan statistics).
Z	A matrix of the original scan statistics (standardized counts) if statistic specified is "all" or "o".
Zw	A matrix of the weighted scan statistics (standardized counts) if statistic specified is "all" or "w".
S	A matrix of the generalized scan statistics (standardized counts) if statistic specified is "all" or "g".
M	A matrix of the max-type scan statistics (standardized counts) if statistic specified is "all" or "m".
R	A matrix of raw counts of the original scan statistic. This output only exists if the statistic specified is "all" or "o".
Rw	A matrix of raw counts of the weighted scan statistic. This output only exists if statistic specified is "all" or "w".
pval.appr	The approximated p-value based on asymptotic theory for each type of statistic specified.
pval.perm	This output exists only when the argument pval.perm is TRUE . It is the permutation p-value from B permutations and appears for each type of statistic specified (same for perm.curve, perm.maxZs, and perm.Z).
perm.curve	A B by 2 matrix with the first column being critical values corresponding to the p-values in the second column.
perm.maxZs	A sorted vector recording the test statistics in the B permutaitons.
perm.Z	A B by n-squared matrix with each row being the vectorized scan statistics from each permutaiton run.

See Also

gSeg, gseg1, gseg2_discrete

Examples

```
data(Example)
# Five examples, each example is a n-length sequence.
# Ei (i=1,...,5): an edge matrix representing a similarity graph constructed on the
# observations in the ith sequence.
# Check '?gSeg' to see how the Ei's were constructed.

## E5 is an edge matrix representing a similarity graph.
# It is constructed on a sequence of length n=200 with a change in both mean
# and variance on an interval (tau1 = 155, tau2 = 185).
r5=gseg2(n,E5,statistics="all")
```

---

gseg2_discrete	*Graph-Based Change-Point Detection for Changed Interval for Data with Repeated Observations*

Description

This function finds an interval in the sequence where their underlying distribution differs from the rest of the sequence when data has repeated observations. It provides four graph-based test statistics.

Usage

```
gseg2_discrete(n, E, id, statistics=c("all","o","w","g","m"), l0=0.05*n, l1=0.95*n,
    pval.appr=TRUE, skew.corr=TRUE, pval.perm=FALSE, B=100)
```

Arguments

n	The number of observations in the sequence.
E	The edge matrix (a "number of edges" by 2 matrix) for the similarity graph. Each row contains the node indices of an edge.
id	The index of observations (order of observations).
statistics	The scan statistic to be computed. A character indicating the type of of scan statistic desired. The default is "all".
	"all": specifies to compute **a**ll of the scan statistics: original, weighted, generalized, and max-type;
	"o","ori" or "original": specifies the **o**riginal edge-count scan statistic;
	"w" or "weighted": specifies the **w**eighted edge-count scan statistic;
	"g" or "generalized": specifies the **g**eneralized edge-count scan statistic; and
	"m" or "max": specifies the **m**ax-type edge-count scan statistic.
l0	The minimum length of the interval to be considered as a changed interval.

l1	The maximum length of the interval to be considered as a changed interval.
pval.appr	If it is TRUE, the function outputs p-value approximation based on asymptotic properties.
skew.corr	This argument is useful only when pval.appr=TRUE. If skew.corr is TRUE, the p-value approximation would incorporate skewness correction.
pval.perm	If it is TRUE, the function outputs p-value from doing B permutations, where B is another argument that you can specify. Doing permutation could be time consuming, so use this argument with caution as it may take a long time to finish the permutation.
B	This argument is useful only when pval.perm=TRUE. The default value for B is 100.

Value

Returns a list scanZ with tauhat, Zmax, and a vector of the scan statistics for each type of scan statistic specified. See below for more details.

tauhat_a	An estimate of the two ends of the changed interval for averaging approach.
tauhat_u	An estimate of the two ends of the changed interval for union approach.
Z_a_max	The test statistic (maximum of the scan statistics) for averaging approach.
Z_u_max	The test statistic (maximum of the scan statistics) for union approach.
Zo_a	A matrix of the original scan statistics (standardized counts) for averaging approach if statistic specified is "all" or "o".
Zo_u	A matrix of the original scan statistics (standardized counts) for union approach if statistic specified is "all" or "o".
Zw_a	A matrix of the weighted scan statistics (standardized counts) for averaging approach if statistic specified is "all" or "w".
Zw_u	A matrix of the weighted scan statistics (standardized counts) for union approach if statistic specified is "all" or "w".
S_a	A matrix of the generalized scan statistics (standardized counts) for averaging approach if statistic specified is "all" or "g".
S_u	A matrix of the generalized scan statistics (standardized counts) for union approach if statistic specified is "all" or "g".
M_a	A matrix of the max-type scan statistics (standardized counts) for averaging approach if statistic specified is "all" or "m".
M_u	A matrix of the max-type scan statistics (standardized counts) for union approach if statistic specified is "all" or "m".
Ro_a	A matrix of raw counts of the original scan statistic for averaging approach. This output only exists if the statistic specified is "all" or "o".
Ro_u	A matrix of raw counts of the original scan statistic for union approach. This output only exists if the statistic specified is "all" or "o".
Rw_a	A matrix of raw counts of the weighted scan statistic for averaging approach. This output only exists if statistic specified is "all" or "w".

Rw_a	A matrix of raw counts of the weighted scan statistic for union approach. This output only exists if statistic specified is "all" or "w".
pval.appr	The approximated p-value based on asymptotic theory for each type of statistic specified.
pval.perm	This output exists only when the argument pval.perm is TRUE . It is the permutation p-value from B permutations and appears for each type of statistic specified (same for perm.curve, perm.maxZs, and perm.Z).
perm.curve	A B by 2 matrix with the first column being critical values corresponding to the p-values in the second column.
perm.maxZs	A sorted vector recording the test statistics in the B permutaitons.
perm.Z	A B by n-squared matrix with each row being the vectorized scan statistics from each permutaiton run.

See Also

gSeg, gseg2, nnl, gseg1_discrete

Examples

```
d = 50
mu = 2
tau = 100
n = 200

set.seed(500)
y1_temp = matrix(rnorm(d*tau),tau)
sam1 = sample(1:tau, replace = TRUE)
y1 = y1_temp[sam1,]
y2_temp = matrix(rnorm(d*(n-tau),mu/sqrt(d)), n-tau)
sam2 = sample(1:tau, replace = TRUE)
y2 = y2_temp[sam2,]

y = rbind(y1, y2)

# This data y has repeated observations
y_uni = unique(y)
E = nnl(dist(y_uni), 1)

cha = do.call(paste, as.data.frame(y))
id = match(cha, unique(cha))

r1 = gseg2_discrete(n, E, id, statistics="all")
```

---

n *The Number of Observations in the Sequence*

Description

This is the variable name for the number of observations in the sequences in the "Example" data.

---

nnl   *Construct the Nearest Neighbor Link (NNL)*

Description

This function provides a method to construct the NNL.

Usage

```
nnl(distance, K)
```

Arguments

distance   The distance matrix on the distinct values (a "number of unique observations"
                 by "number of unique observations" matrix).

K   The value of k in "k-MST" or "k-NNL" to construct the similarity graph.

Value

E   The edge matrix representing the similarity graph on the distinct values with
                 the number of edges in the similarity graph being the number of rows and 2
                 columns. Each row records the subject indices of the two ends of an edge in the
                 similarity graph.

See Also

gSeg, gseg1_discrete, gseg2_discrete

Examples

```
n = 50
d = 10
dat = matrix(rnorm(d*n),n)
sam = sample(1:n, replace = TRUE)
dat = dat[sam,]

# This data has repeated observations
dat_uni = unique(dat)
E = nnl(dist(dat_uni), 1)
```

Index