sDmatCluster

March 6, 2015

`sDmatCluster` *Function to partition a grid map into clusters*

Description

`sDmatCluster` is supposed to obtain clusters from a grid map. It returns an object of class "sBase".

Usage

```
sDmatCluster(sMap, which_neigh = 1, distMeasure = c("median", "mean",
"min",
"max"), clusterLinkage = c("average", "complete", "single", "bmh"),
reindexSeed = c("hclust", "svd", "none"))
```

Arguments

`sMap`	an object of class "sMap"
`which_neigh`	which neighbors in 2D output space are used for the calculation. By default, it sets to "1" for direct neighbors, and "2" for neighbors within neighbors no more than 2, and so on
`distMeasure`	distance measure used to calculate distances in high-dimensional input space. It can be one of "median", "mean", "min" and "max" measures
`clusterLinkage`	cluster linkage used to derive clusters. It can be "bmh", which accumulates a cluster just based on best-matching hexagons/rectanges but can not ensure each cluster is continuous. Instead, each cluster is continuous when using region-growing algorithm with one of "average", "complete" and "single" linkages
`reindexSeed`	the way to index seed. It can be "hclust" for reindexing seeds according to hierarchical clustering of patterns seen in seeds, "svd" for reindexing seeds according to svd of patterns seen in seeds, or "none" for seeds being simply increased by the hexagon indexes (i.e. always in an increasing order as hexagons radiate outwards)

Value

an object of class "sBase", a list with following components:

- `seeds`: the vector to store cluster seeds, i.e., a list of local minima (in 2D output space) of distance matrix (in input space). They are represented by the indexes of hexagons/rectangles
- `bases`: the vector with the length of nHex to store the cluster memberships/bases, where nHex is the total number of hexagons/rectanges in the grid
- `call`: the call that produced this result

Note

The first item in the return "seeds" is the first cluster, whose memberships are those in the return "bases" that equals 1. The same relationship is held for the second item, and so on

See Also

sPipeline, sDmatMinima, sBMH, sNeighDirect, sDistance, visDmatCluster

Examples

```
# 1) generate an iid normal random matrix of 100x10
data <- matrix( rnorm(100*10,mean=0,sd=1), nrow=100, ncol=10)

# 2) get trained using by default setup
sMap <- sPipeline(data=data)

# 3) partition the grid map into clusters based on different criteria
# 3a) based on "bmh" criterion
# sBase <- sDmatCluster(sMap=sMap, which_neigh=1, distMeasure="median", clusterLinkage="bmh")
# 3b) using region-growing algorithm with linkage "average"
sBase <- sDmatCluster(sMap=sMap, which_neigh=1, distMeasure="median",
clusterLinkage="average")

# 4) visualise clusters/bases partitioned from the sMap
visDmatCluster(sMap,sBase)
```