KBase clustering service methods and types

Data checks (NA, row centered (mean/median/mode), col centered, lo variance, redundant)

In: Matrix2D

Out: Report

Example:

Data pre-processing (impute missing values, column mean center)

In: Matrix2D, Report

Out: Matrix2D

Example:

Determine K (K-means)

In: Matrix2D

Out: KMeansParam

Example:

library(clValid)

valid <- clValid(logratios\_median, nClust=c(2:50),maxitems=4338, clMethods=c("kmeans"),validation=c("internal"))

Cluster(HCL)

In: Matrix2D, HCLParam

Out: Dendrogram

Example:

library(amap)

hcout=hcluster(data,method="correlation")

Cluster(K-means)

In: Matrix2D, KMeansParam

Out: ClusteringResult

Example:

km <- kmeans(logratios\_median, 11, iter.max = 1000, nstart=1000,algorithm="Lloyd")

Cut tree (HCL)

In: Dendrogram, HCLCutParameters

Out: ClusteringResult

Example:

max1=max(hcout$height)

tab1=table(cutree(hcout,h=max1))

Cluster quality

In: ClusteringResult, QualityParam

Out: ClusteringQuality

Example:

source(“Miner.R”)

for(i in 1:max(clusts)) {

index <- which(clusts == i)

curdata <- logratios\_median[index,]

curdatamean <- rbind(curdata, colMeans(curdata))

MSEC <- ExpCritI(as.matrix(logratios\_median),index,1:dim[2],1,3,0)

MSECs <- c(MSECs, 1/(MSEC+1))

clusters <- c(clusters, paste(paste(index,collapse=","),"/",sep=""))

clusters\_ind[[i]] <- index

}

pvalsC <- c()

counts <- c()

num <- 10000

for(i in 1:length(MSECs)) {

print(i)

rgenes\_l <- vector("list", num)

countC <- 0

for(j in 1:num) {

rgenes\_l[[i]] <- sample.int(dim[1],size=length(clusters\_ind[[i]]),replace=F)

MSEC <- ExpCritI(as.matrix(logratios\_median),rgenes\_l[[i]],c(1,2,3),1,3,0)

#print(MSER)

critsC <- 1/ (MSEC +1)

if(MSECs[i] < critsC) {

countC <- countC+1

}

}

pvalsC <- c(pvalsC, countC/num)

counts <- c(counts, countC)

}

Cluster ranking

In: ClusteringResult, RankParam

Out: ClusteringResult

Example: