**# functions for JSD based clustering & analysis.**

dist.JSD <- function(inMatrix, pseudocount=0.000001, ...) {

KLD <- function(x,y) sum(x \*log(x/y))

JSD<- function(x,y) sqrt(0.5 \* KLD(x, (x+y)/2) + 0.5 \* KLD(y, (x+y)/2))

matrixColSize <- length(colnames(inMatrix))

matrixRowSize <- length(rownames(inMatrix))

colnames <- colnames(inMatrix)

resultsMatrix <- matrix(0, matrixColSize, matrixColSize)

inMatrix = apply(inMatrix,1:2,function(x) ifelse (x==0,pseudocount,x))

for(i in 1:matrixColSize) {

for(j in 1:matrixColSize) {

resultsMatrix[i,j]=JSD(as.vector(inMatrix[,i]),

as.vector(inMatrix[,j]))

}

}

colnames -> colnames(resultsMatrix) -> rownames(resultsMatrix)

as.dist(resultsMatrix)->resultsMatrix

attr(resultsMatrix, "method") <- "dist"

return(resultsMatrix)

}

pam.clustering=function(x,k) { # x is a distance matrix and k the number of clusters

require(cluster)

cluster = as.vector(pam(as.dist(x), k, diss=TRUE)$clustering)

return(cluster)

}

**#packages Installed for the 3d plot rendering**

install.packages("cluster")

install.packages("clusterSim")

library(cluster)

install.packages("rgl")

library(rgl)

library(clusterSim)

**#80 topic model recipe**

data=read.table("topics80.csv", header=T, row.names=1, dec=".", sep=",")

head(data, n=10)

data.dist=dist.JSD(data)

data.cluster=pam.clustering(data.dist, k=5)

require(clusterSim)

nclusters = index.G1(t(data), data.cluster, d = data.dist, centrotypes = "medoids")

nclusters=NULL

for (k in 1:20) {

if (k==1) {

nclusters[k]=NA

} else {

data.cluster\_temp=pam.clustering(data.dist, k)

nclusters[k]=index.G1(t(data),data.cluster\_temp, d = data.dist,

centrotypes = "medoids")

}

}

#CH index check

plot(nclusters, type="h", xlab="k clusters", ylab="CH index",main="Optimal number of clusters")

#noise removal

data=noise.removal(data, percent=0.01)

#plots

obs.pcoa=dudi.pco(data.dist, scannf=F, nf=3)

dev.new()

s.class(obs.pcoa$li, fac=as.factor(data.cluster), grid=F,sub="Principal coordiante analysis with K=80(topics)")

s.class(obs.pcoa$li,fac=as.factor(data.cluster), grid=F, cell=0, cstar=0)

s.label(obs.pcoa$li, xax = 1, yax = 2, label = row.names(obs.pcoa$li),sub="PCoA Labeled with K=80(topics)")

**#20 topic model recipe**

data=read.table("topics20.csv", header=T, row.names=1, dec=".", sep=",")

head(data, n=10)

data.dist=dist.JSD(data)

data.cluster=pam.clustering(data.dist, k=3)

require(clusterSim)

nclusters = index.G1(t(data), data.cluster, d = data.dist, centrotypes = "medoids")

nclusters=NULL

for (k in 1:20) {

if (k==1) {

nclusters[k]=NA

} else {

data.cluster\_temp=pam.clustering(data.dist, k)

nclusters[k]=index.G1(t(data),data.cluster\_temp, d = data.dist,

centrotypes = "medoids")

}

}

#CH index check

plot(nclusters, type="h", xlab="k clusters", ylab="CH index",main="Optimal number of clusters")

#noise removal

data=noise.removal(data, percent=0.01)

#plots

obs.pcoa=dudi.pco(data.dist, scannf=F, nf=3)

dev.new()

s.class(obs.pcoa$li, fac=as.factor(data.cluster), grid=F,sub="Principal coordiante analysis with K=20(topics)")

s.class(obs.pcoa$li,fac=as.factor(data.cluster), grid=F, cell=0, cstar=0)

s.label(obs.pcoa$li, xax = 1, yax = 2, label = row.names(obs.pcoa$li),sub="PCoA Labeled with K=20(topics)")