Assignment 2: Nonlinear Dimensionality Reduction

Gonzalo Solera, Zofia Tarant

6 01 2021

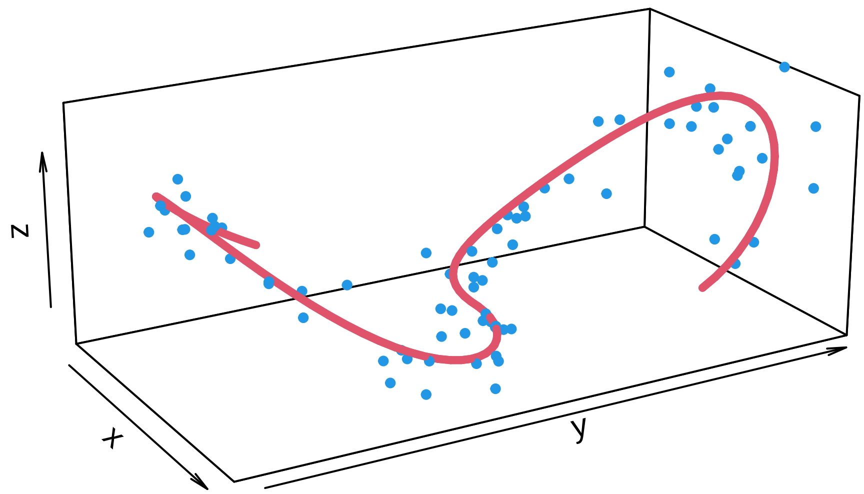
Dataset

```
t <- seq(-1.5*pi,1.5*pi,l=100)
R<- 1
n<-75
sd.eps <- .15

set.seed(1)
y <- R*sign(t) - R*sign(t)*cos(t/R)
x <- -R*sin(t/R)
z <- (y/(2*R))^2
rt <- sort(runif(n)*3*pi - 1.5*pi)
eps <- rnorm(n)*sd.eps
ry <- R*sign(rt) - (R+eps)*sign(rt)*cos(rt/R)
rx <- -(R+eps)*sin(rt/R)
rz <- (ry/(2*R))^2 + runif(n,min=-2*sd.eps,max=2*sd.eps)
XYZ <- cbind(rx,ry,rz)

distXYZ <- dist(XYZ)


lines3D(x,y,z,colvar = NULL,
         phi = 20, theta = 60, r =sqrt(3), d =3, scale=FALSE,
         col=2,lwd=4,as=1,
         xlim=range(rx),ylim=range(ry),zlim=range(rz))
points3D(rx,ry,rz,col=4,pch=19,cex=.6,add=TRUE)
```



Local Continuity Meta-criteria

Q1. Write a function that computes MadjK as a function of two distance matrices between points.

```
LC <- function(D, X, K_) {
  # D - high dim. matrix
  # X - euclidean distance matrix
  wD <- as.matrix(dissWeights(D, type = "knn", k = K_))
  wX <- as.matrix(dissWeights(X, type = "knn", k = K_))
  N.K_ = mean(rowSums(wD & wX))
  M.K_ <- N.K_/K_
  M.K_-K_/(n-1)
}
```

Choosing the tuning parameters in Local MDS

Q2. Choose the pair (K,tau) maximizing the Local Continuity Meta-criteria.

```
Stress.LocalMDS <- function(conf,n,dist.orig,k=5,tau=1){
  require(smacof)
  q<-1
  mconf <- matrix(conf,nrow = n, byrow = FALSE) # configuration matrix
```

```r
  Eucl.dist <- dist(mconf)
  w <- dissWeights(dist.orig, type = "knn", k = k)
  w.1 <- (w==1)
  card.N.k <- sum(w.1)
  card.N.k.c <- n*(n-1)/2 - card.N.k
  t <- (card.N.k/card.N.k.c)*median(dist.orig[w.1])*tau
  return(sum((dist.orig[w.1]-Eucl.dist[w.1])^2) - t*sum(Eucl.dist[!w.1]))
}
conf0 <- as.numeric(cmdscale(distXYZ))

K.search <- c(5,10,15)
tau.search <- c(.1, .5, 1)
best_local_contuinity <- NULL
for(K in K.search) {
  for(tau in tau.search) {
    localMDS.S.res <- optim(par=conf0, fn=Stress.LocalMDS, n=n, dist.orig=distXYZ, k=K, tau=tau, method
    conf.localMDS.S.res <- matrix(localMDS.S.res$par, nrow = n, byrow = FALSE)

    K_ <- 10
    local_contuinity <- LC(D=distXYZ, X=dist(conf.localMDS.S.res[,1]), K_)
    if(is.null(best_local_contuinity) || best_local_contuinity < local_contuinity) {
      best_local_contuinity <- local_contuinity
      best_K <- K
      best_tau <- tau
      localMDS.max <- conf.localMDS.S.res
    }
  }
}

print(best_K)
```

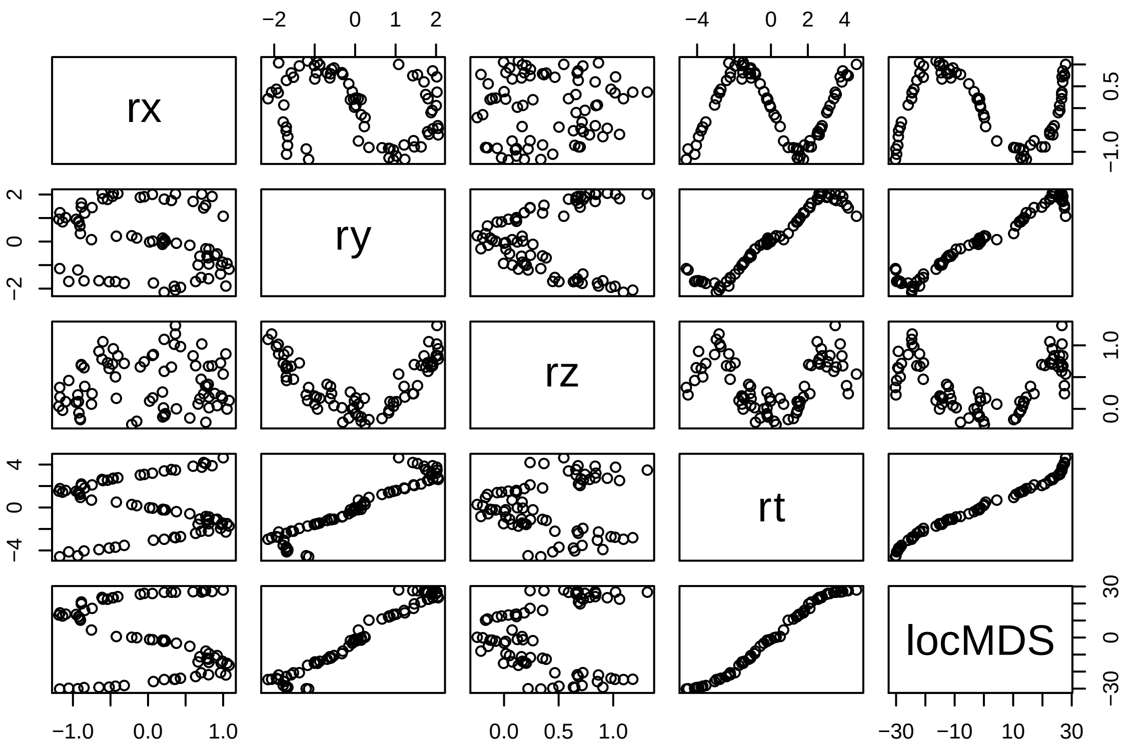
```
## [1] 5
```

```r
print(best_tau)
```

```
## [1] 0.5
```

Q3. Graphical representation of the Local MDS output For the optimal K and tau, the output of the Local MDS is a q-dimensional configuration, that is, a n×q matrix (in our case with q=1).

```r
pairs(cbind(XYZ, rt, locMDS=localMDS.max[,1]))
```

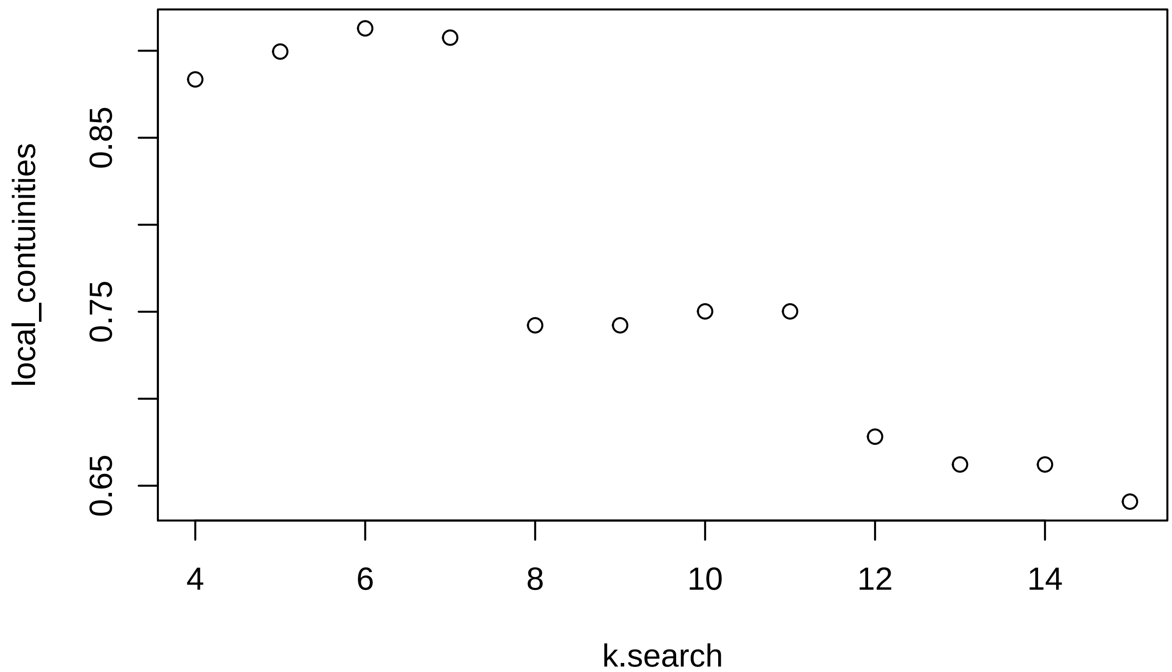


Choosing the tuning parameters in ISOMAP

Q4. Choose the parameter k maximizing the Local Continuity Meta-criteria.

```
k.search <- seq(4, 15)
K_ <- 10

local_contuinities <- map(k.search,
                          function(k)
                            LC(D=distXYZ, X=dist(isomap(distXYZ, k=k)$points[,1]), K_=K_))
plot(k.search, local_contuinities)
```

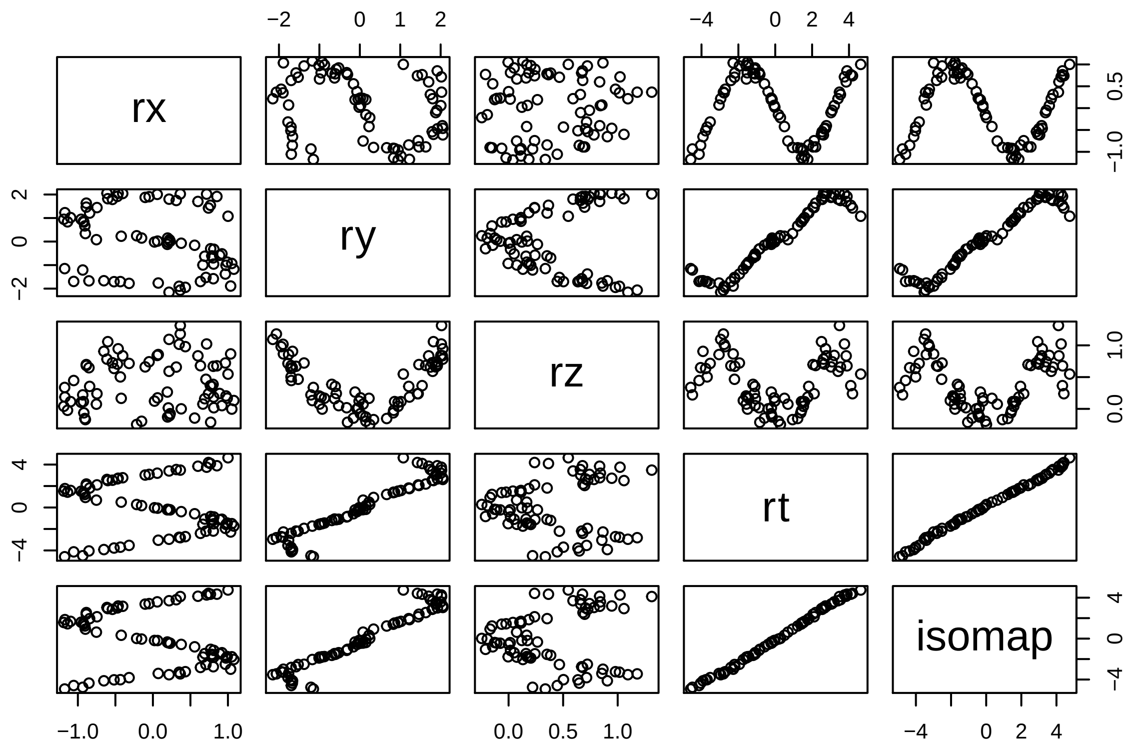



```
best_k <- k.search[which.max(local_contuinities)]
print(best_k)
```

```
## [1] 6
```

Q5. Graphical representation of the ISOMAP output

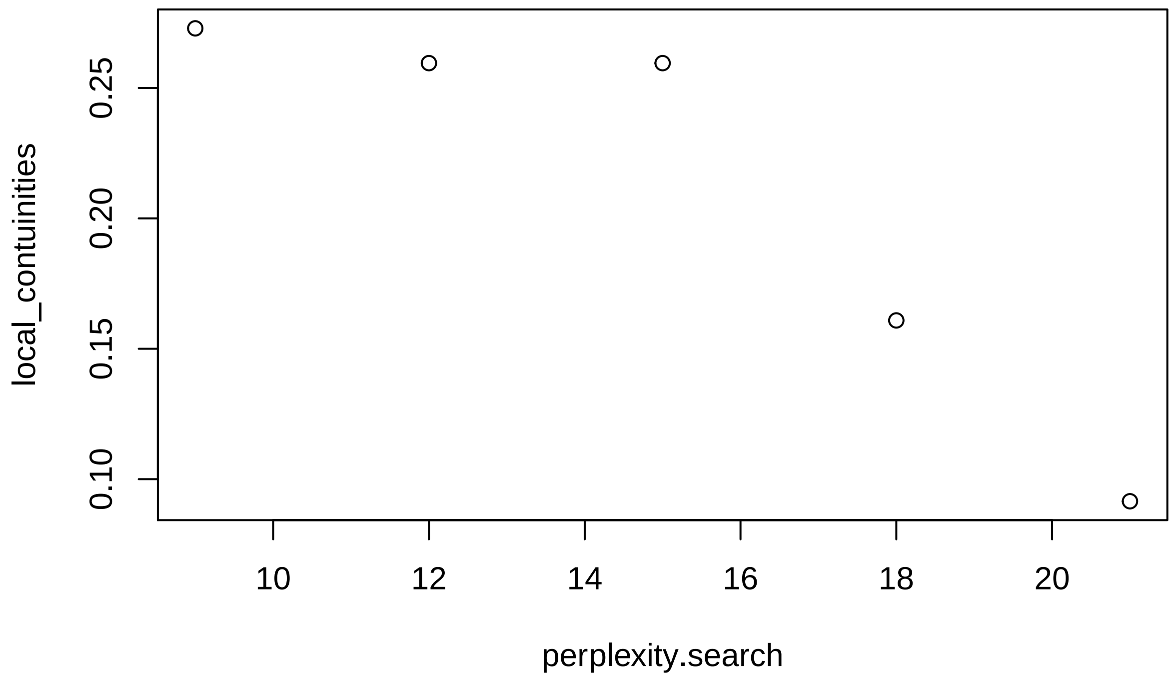
```
isomap.max <-isomap(distXYZ, k=best_k)
pairs(cbind(XYZ, rt, isomap=isomap.max$points[,1]))
```



Choosing the tuning parameters in t-SNE

Q6. Choose the parameter perplexity maximizing the Local Continuity Meta-criteria.

```
K_ <- 10
perplexity.search <- seq(9, 21, by=3)
local_contuinities <- map(perplexity.search,
                          function(p)
                            LC(
                              D=distXYZ,
                              X=dist(Rtsne(distXYZ, perplexity=p, dims=1, theta=0, pca=FALSE, max_iter=
                              K_=K_))
plot(perplexity.search, local_contuinities)
```

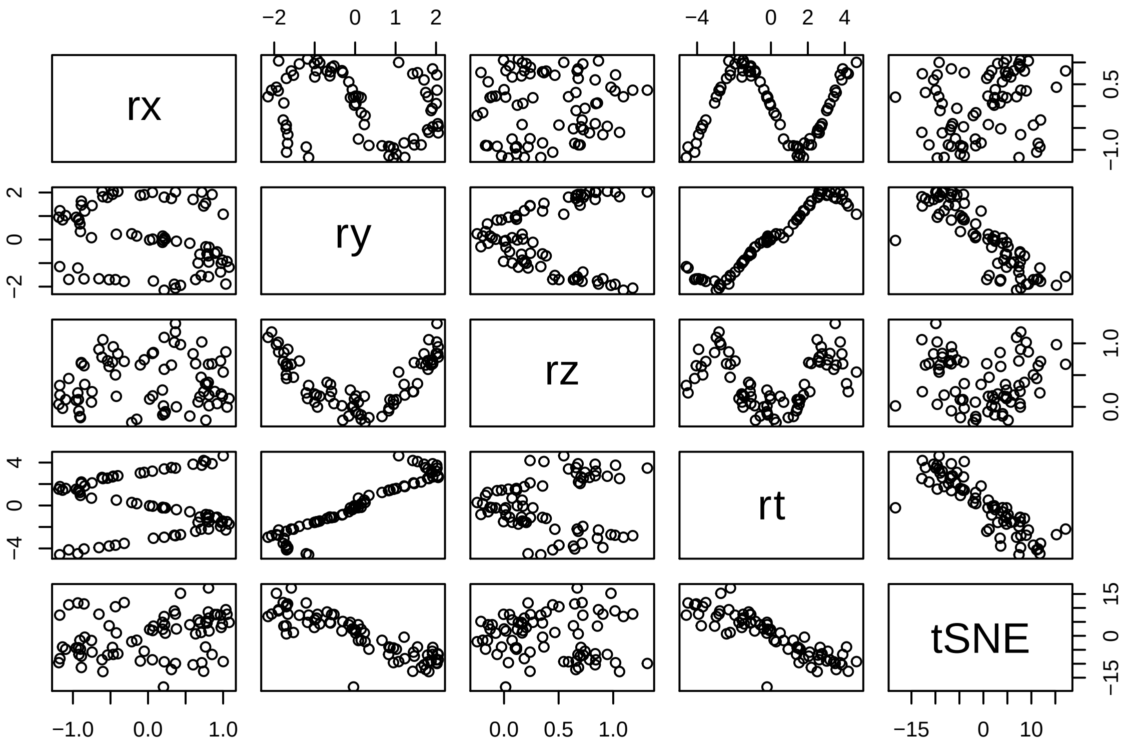



```
best_perplexity <- perplexity.search[which.max(local_contuinities)]
print(best_perplexity)
```

```
## [1] 9
```

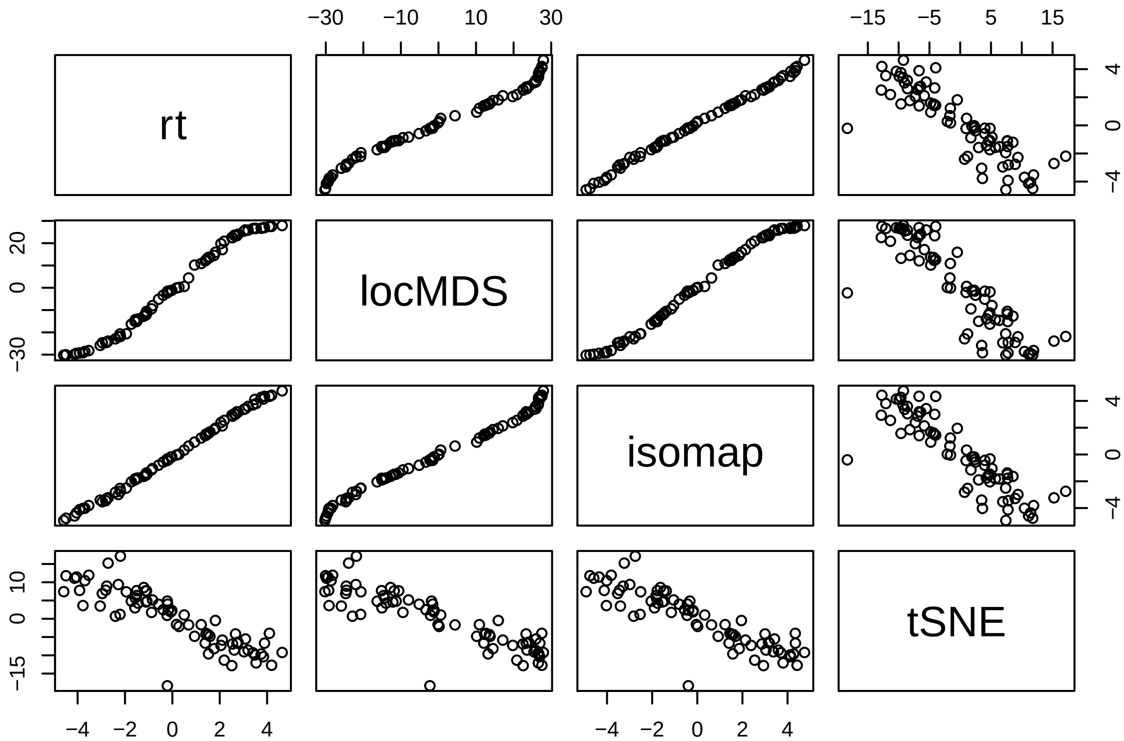
Q7. Graphical representation of the t-SNE output.

```
rtsne.max <- Rtsne(distXYZ, perplexity=best_perplexity, dims=1, theta=0, pca=FALSE, max_iter=200)
pairs(cbind(XYZ, rt, tSNE=rtsne.max$Y[,1]))
```



Q8. Compare graphically the results of the different techniques.

```
pairs(cbind(rt, locMDS=localMDS.max[,1],
            isomap=isomap.max$points[,1],
            tSNE=rtsne.max$Y[,1]))
```



The result obtained using isomap scaled the data the closest to a linear representation.