

Exercice 1

```
1. distX = as.matrix(X)
   distX = distX ^2

2. Mthode 1
   XC = scale(X, scale=T)
   W = XC\%*\%t(XC)

   Mthode 2
   QN = diag(nrow(X)) - matrix(1, nrow(X), nrow(X))/nrow(X)
   W = -1/2*QN\%*\%distX\%*\%QN

3. Pour vrifier si elle est dfinie semi-positive, il suffit de vrifier que les valeurs propres
   eigen(W)

4. L = eigen(W)$values
   L = diag(nrow(X))*L

   V = eigen(W)$vectors

5. C = V\%*\%sqrt(L)
   pas oublier de retirer les NaN

   plot(C)
   idem biplot(princomp(X))
```

Exercice 2

```
m = as.vector(mutation)
b = cmdscale(mutation, 2, T)
c = as.vector(dist(b$points))
plot(b,c) problme mais on est pas loin
qualit calculer avec les valeurs propres b[,1]$eigen etc... / sum

on refait de mme avec cmdscale(mutation, 3, T) jusqu' 5
```

Exercice 3

```
library(cluster) clusplot
```

Iris

```
iris = iris[,1:4]
res2 = kmeans(iris, 2)
plot(iris, col= res2$cluster)
clusplot(iris, res2$cluster)
res2 = kmeans(iris, 3)
> plot(iris, col= res2$cluster)
```

2 types diffrents : 143 ou 78.9 pour l'inertie des classes (tot.withinss)

```
$for(j in 2:10){
  for(i in 1:100){
    test[j, i] = kmeans(iris, j)$tot.withinss
  }
}$
apply(test, 2, min)
```

La solution qui apparait en 3 classes n'est pas flagrante avec le tableau des minimums des i
Une solution serait de pnaliser un grand nombre de classes par le nombre d'individus prsents

Crabs

```
library(MASS)
```

Mutations

```
res = kmeans(mutations2, 2)
plot(cmdscale(mutations), col=res$cluster)
```

Avec 3 vert au milieu des noirs

4 cluster seulement un point dans le dernier

tableau de contingence pour comparer les partitions table(res\$cluster, res2\$cluster)