Rapport du projet : Exploration visuelle des données génétiques

Wafa AYAD

23/12/2016

1 Introduction

Dans le processus d'analyse des données, et avant l'application de l'apprentissage -que se soit supervisé ou non supervisé- sur les données que nous disposons, il est indispensable de les bien comprendre afin de mieux les interpreter et prédire le comportement futur des données arrivants et les associer à leurs classes d'appartenance. Pour ce faire, l'exploration visuelle de ces données vise à donner une image claire de la base des données et facilite aux explorateurs leur manipulation.

Dans ce documents, nous allons faire une synthèse des différentes méthodes de visualisation étudiées en cours appliquées à 2 jeux de données génétiques. Nous commençons par une analyse biève des données génétiques fournises, à savoir: *GORDON* et *pomeroy*. Par la suite, nous discutons les différentes méthodes de visualisation que nous avons choisis avec la justification de ces choix. Pour explorer les données, il existe plusieurs types d'approches telle que les méthodes de réduction de dimension que se soit par l'extraction de caractéristiques, ou selection de variables, les méthodes de sub-clustering, à savoir celles basées sur corrélation (ACP) ou bien les méthodes de co-clustering (clusterisation sur individus et variables).

Pour ce projet, nous utilisons des méthodes de visualisation réduisant la dimension des données d'une part linéaires (exemple: ACP, ADL et MDS), d'une autre part non lénéaires (LLE et ISOMAP) et l'algorithme basé reseaux de neuronnes (SOM). Nous finalisons cette étude par l'analyse et commentaire des résultats donnés par ces algorithmes, pour conclure sur les méthodes adéquates pour chaque jeu de données.le code complet en R est disponible dans les fichiers *gordon.r* et *pomeroy.r* joints.

2 L'analyse des jeux de données: *GORDON* ET *POMEROY*

2.1 *GORDON*

Ce jeu de données sert comme echantillon (182 individus) pour la distinction pathologique entre le mésothéliome pleural malin (MPM) et l'adénocarcinome (ADCA) du poumon en utilisant des rapports d'expression génique (1627 gènes).

```
dim(gordon)
```

```
## [1] 1627  182
```

2.2 *POMEROY*

Cette base contient 1380 gènes expressant 43 individus apparetenant à 5 types de tumeurs embryonnaire du système nerveux central à savoir: MD, Mglio, Rhab, Ncer et PNET.

```
dim(pomeroy)
```

```
## [1] 1380   43
```

Dans les deux banchmaks précédents, il n'y a pas des valeurs manquantes. Nous remarquons que le nombre de gènes est beaucoup elevé que le nombre de tuples. Donc, une première intuition qui vient à l'esprit est de soit selectionner les gènes les plus porteuses d'informations et les plus présentatrices des autres, ou bien procéder à l'extraction des caractéristiques des gènes les plus significative. Une autre idée est de normalisé les valeurs prises par les gènes par la sustraction de leurs moyenne de leurs valeurs et leurs mise en échelle que se soit par rapport aux lignes ou bien aux colonnes. Cela est discuté dans ce qui suit.

3 Prétraitement des données

Avant la visualisation des données, le prétraitement de ces dernières tels que leur normalisation par leur moyenne et leur mise à l'échelle, permet de les projeter dans un espace normalisé. Après le chargement des deux jeux données étudiés, nous remarquons que dans les deux cas le nombre de gènes est beaucoup plus elvé que le nombre d'individus, et que bouceaup de variables dans les deux cas sont corrélées. Donc, nous jugeons qu'il est nécéssaire de procéder à la normalisation, la mise en échelle et l'élimination des variables corrélées, afin de réduire la dimension de l'espace contenant nos données étudiées.

3.1 Mise à l'échelle (Feature Scaling)

Une des techniques performantes pour faciliter la manipulation des deux jeux de données est leur mise à l'echelle. A ce fin, nous rendons les valeurs prises par les gènes entre 0 et 1, pour qu'elles auront un rang comparable de valeurs.

```
gordon_scale <- scale(gordon_frame)
pomeroy_scale <- scale(pomeroy_frame)
```

3.2 Normalisation (Mean Normalisation)

Pour cette étape, nous calculons pour chaque gène sa moyenne, puis nous remplacons chaque valeur par le résultat de la sustraction de la moyenne de cette colonne de la valeur initiale. Cela permet à chaque gène d'avoir 0 comme moyenne.

```
gordon_norm= apply(gordon_scale, MARGIN = 2, FUN = function(X) (X - min(X))/diff(range(X)))
pomeroy_norm= apply(pomeroy_trans, MARGIN = 2, FUN = function(X) (X - min(X))/diff(range(X)))
```

3.3 Elimination des variables corrélées

Comme nous l'avons cité auparavant, le nombre de gènes est trés élevé par rapport aux nombre d'invidus dont nous disposons, nous appliquons un processus d'élimination de certaines gènes.

```
library(MASS)
library(klaR)
#gordon  <- greedy.wilks(gordon_norm[,1]~ ., data = gordon_norm[,-1], niveau = 0.5)
```

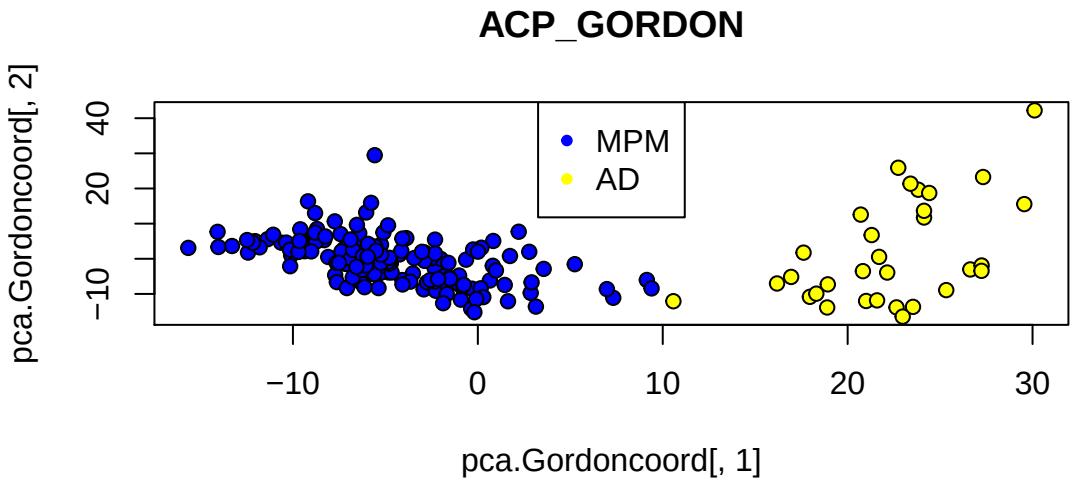
Cette instruction prends énormément de temps vu qu'elle calcule les matrice de corréllation entre les 1627 gènes que nous avons, puis elle selectionne les gènes les plus pertinentes. A la fin de cette étape, nous n'aurons que 55 gènes à manipuler. Les autres gènes sont éliminées car elles ne sont pas indispensables pour cette étude. Certaines d'elles peuvent être obtenues à partir d'autres gènes. Une autre méthode que nous proposons pour résoudre le problème précédement posé est de faire une boucle pour éliminer à chaque itération un nombre fini de variables collinéaires jusqu'à ce que nous obtenons un taux minimum de collinéairité entre les gènes (ou même nul !), sauf que ces méthodes nécissitent fortement une présence d'un expert permettant de juger sur l'effet de l'information apporté par les gènes supprimées et celles restantes.

4 Exploration des données avec les méthodes de visualisation

Dans cette partie, nous utilisons des méthodes d'extraction de caractéristiques linéaire et non linéaire permettant de trouver une meilleure séparation des individus dans l'espace de projection.

4.1 *GORDON*

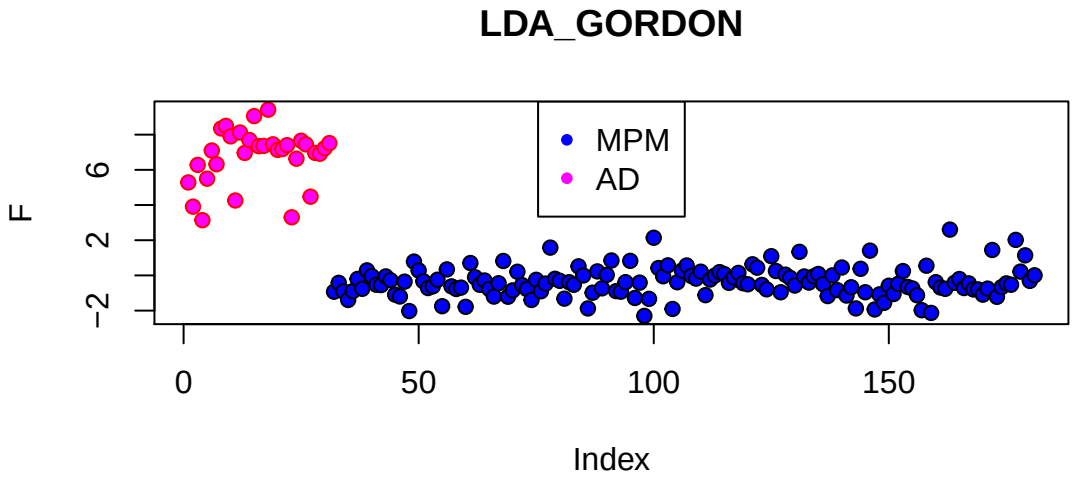
4.1.1 L'analyse des Composantes Principales (ACP)



On prenant les deux premières composantes de la PCA, les resultats donnés sont adéquats à nos attente, car on voit bien qu'on a deux classes bien séparées.

4.1.2 Linear Discriminant Analysis (LDA)

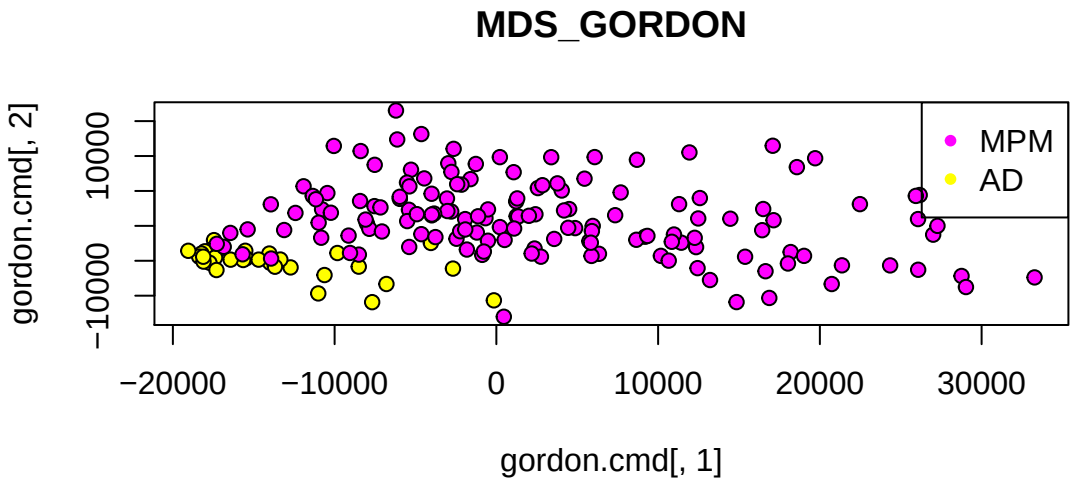
```
## Warning in lda.default(x, grouping, ...): variables are collinear
```



Il est très clair que les résultats de LDA sont les meilleurs, car nous voyons bien que la marge entre les deux classes MPM et AD est maximale.

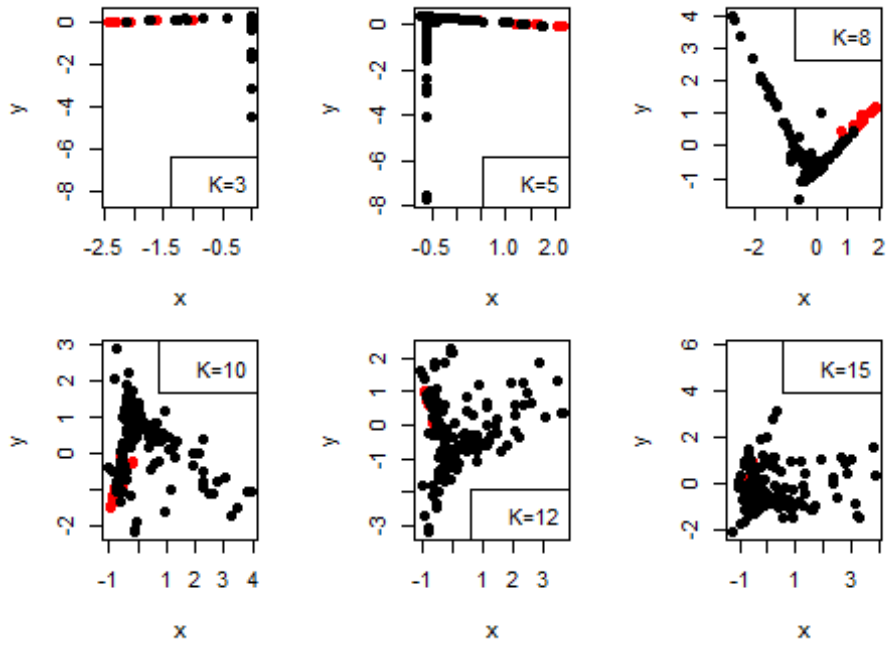
4.1.3 Multi-Dimentional Scaling (MDS)

MDS fonctionne sur des matrices carrées, contenant les distances entre les différentes variables. Dans ce qui suit, nous calculons la matrice des distances entre les différentes gènes. Dans notre application de l'algorithme MDS, nous utilisons sa version mtrique (MDS classic) basée sur les distance euclidiennes.



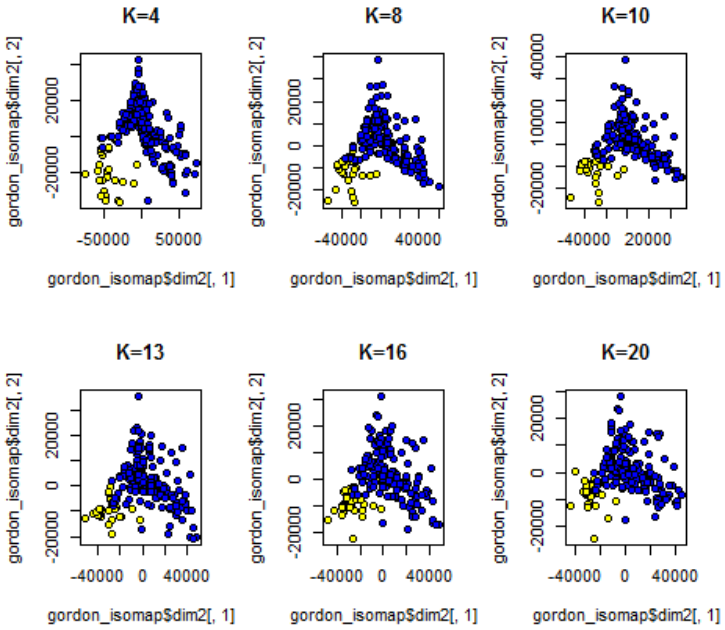
Nous remarquons que les réultats de la projection sur les deux axes d'MDS donnés sont moins bons que ceux de l'ACP et la LDA.

4.1.4 Locally Linear Embedding (LLE)



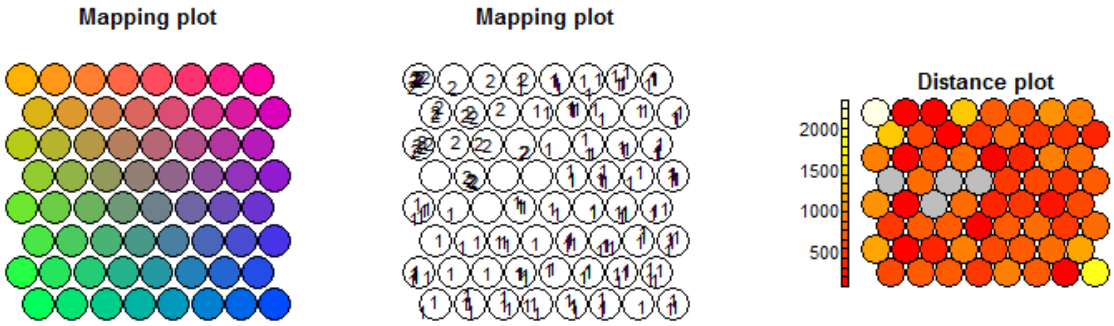
Dans le résultat ci-avant, nous remarquons qu'en variant la valeur de k, l'algorithme LLE arrive pour quelques valeurs de distinguer bien entre les deux classes (k=8 et k=10). pour k=12, nous voyons à peine la classe AD, par rapport a MPM qui est bien visualisée.

4.1.5 ISOMAP



L'application d'ISOMAP sur les données *GORDON* tout en variant la valeur de K entre 4 et 20 donne une marge de séparation très remarquable surtout en K=10.

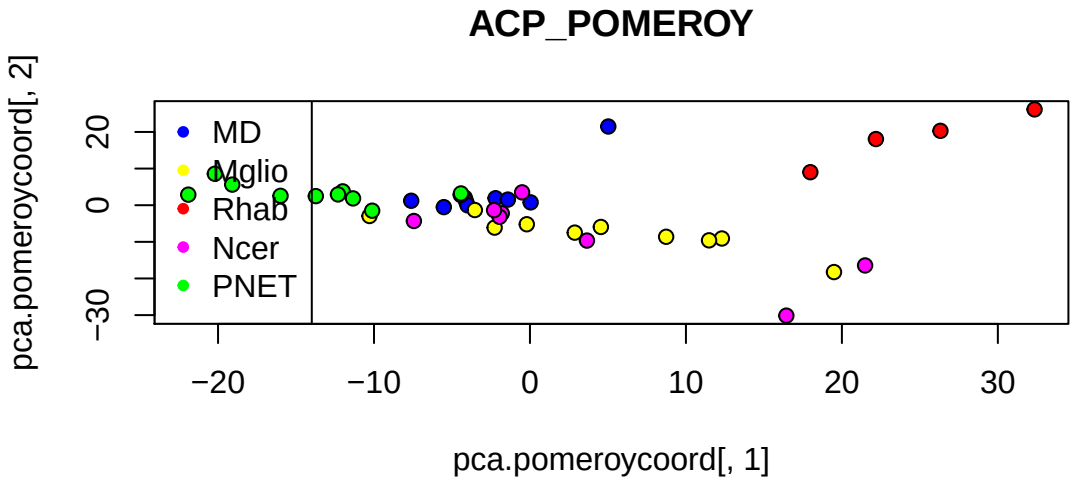
4.1.6 Self-Organizing Map (SOM)



Nous remarquons que l'algorithme basé réseaux de neuronnes map généralement les éléments de la classe MPM vers le haut (G et D), tandis que les individus apparetenant au cancer AD sont mapés en bas et au milieu. SOM a trouvé un plan plus au moins bon pour exposer les deux types du cancer en fonction des gènes les expressant.

4.2 *POMEROY*

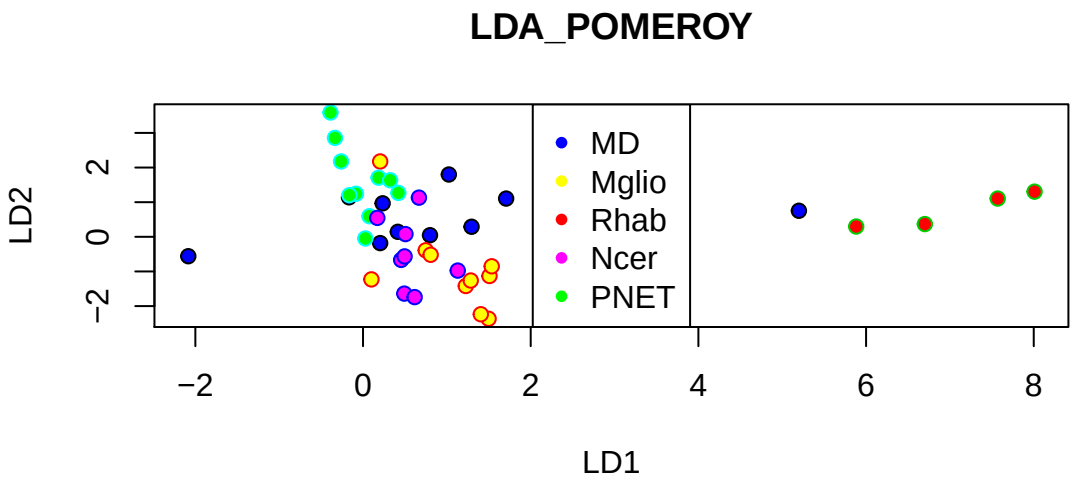
4.2.1 L'analyse des Composantes Principales (ACP)



Les classes là ou les données *POMEROY* appartiennent sont mal séparées par l'ACP. Nous ne pouvons pas distinguer l'une des autres.

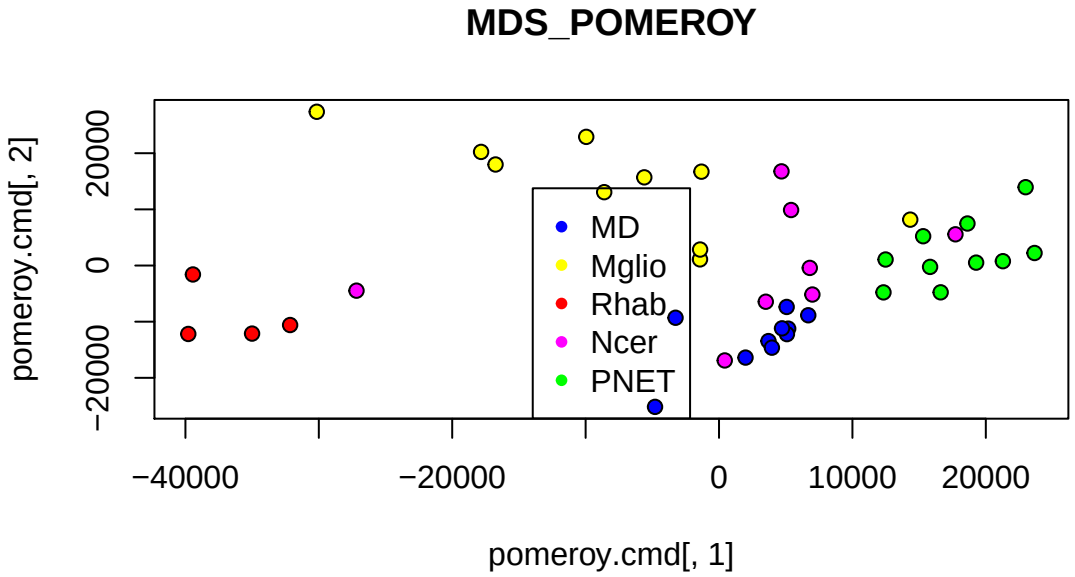
4.2.2 Linear Discriminant Analysis (LDA)

```
## Warning in lda.default(x, grouping, ...): variables are collinear
```



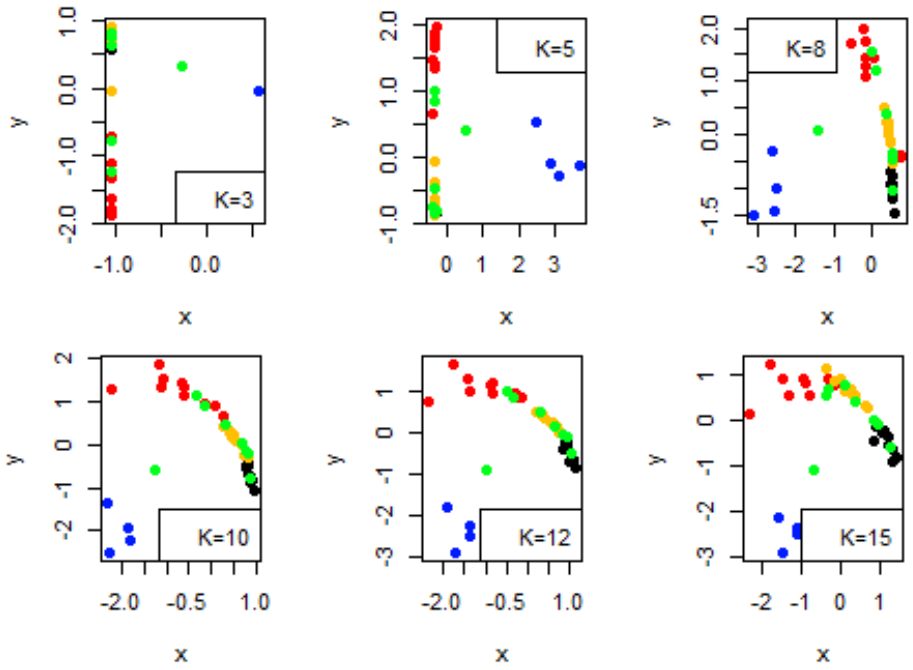
Pareil à l'ACP, la visualisation du jeux de données *POMEROY* est mal faite par la LDA, nnous constatons que ce jeux de données nécissite un types d'algorithme un peu spécial pour bien distinguer les classes les unes des autres.

4.2.3 Multi-Dimentional Scaling (MDS)



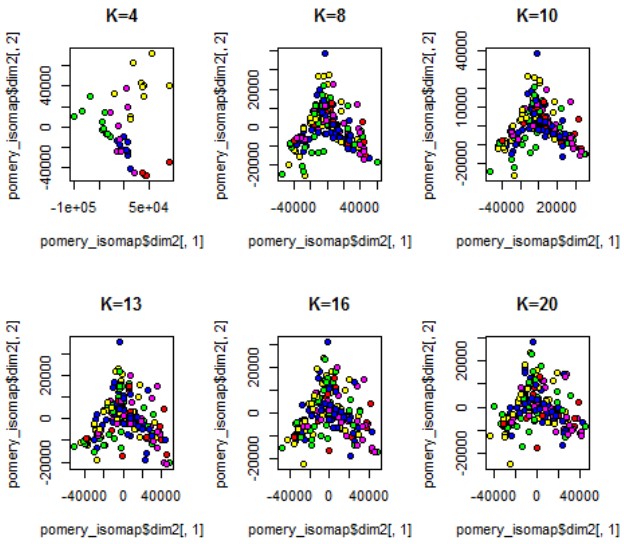
Les résultats montrés dans la figure précédente sont donnés par l'algorithme MDS classic basé sur la distance euclidienne. Nous remarquons que la séparation des 5 classe est mieux que celle données par l'ACP et la LDA.

4.2.4 Locally Linear Embedding (LLE)



Pour k=10, 12 et 15, nous remarquons que LLE arrive à séparer les éléments appartenant aux classes *Rhab* et *PNET* des autres. Cela est du au fait que les individus ayant le cancer du type (MD, Mglio et Ncer) sont exprimés par des valeurs trés proches des gènes.

4.2.5 ISOMAP



L'application d'ISOMAP sur les données *POMEROY* tout en variant la valeur de K entre 4 et 20 donne une marge de séparation un peu bonne en K=4. Dans les autres courbes, nous ne pouvons pas voir clairement les 5 classes.

4.2.6 Self-Organizing Map (SOM)

En choisissant **4*4** comme taille de la grille, nous remarquos que SOM fait le mapping des individus selon leurs classe en zones spécifiques pour chaque classe. Donc, nous pouvons bien regrouper les éléments en 5 classes différentes selon la densité de chaque noeud.

5 Conclusion

Dans ce rapport, nous avons décris brièvement les résultats de l'application des méthodes d'exploration visuelle vues en cours sur deux jeux de données génétiques *GORDON* et *POMEROY*. Nous avons vu que LDA est la meilleure approche permettant de séparer les classes du premier échantillon. Tandis que, pour *POMEROY*, il n'y a que l'algorithme SOM qui a donné des résultats plus au moins interessants, malgré que nous avons appliqué un prétraitement du jeu de données avant appliquer le processus de visualisation. Aucun des résultats donnés par les méthodes de réduction de dimension (linéaire et non linéaire) n'est satisfaisant. La restriction de cette étude au méthodes vues en cours nous a limité, car il est clair que ce type de jeux de données nécissite l'application des algorithmes plus avancés tels que les algorithmes de co-clustering permetant de donner des résultats mieux que ceux que nous avons eu dans cette étude comparative. Le traitement (visualisation, classification ... etc) des gènes en point de vue des individus, et le traitement des individus en fonction des gènes les exprimant est une approche trés utile pour ce type de données car, un ensemble de gènes peut être présent avec certains taux dans un ensemble de tumeurs et vice versa, un ensemble de tumeurs peut contenir un ensemble de gènes.

6 Annexe

Le code en r est disponible dans les deux fichiers **GORDON.r** et **POMEROY.r**