UNIVERSITÉ DE SHERBROOKE

Faculté des Sciences

Département d'Informatique

IFT799 - Science des Données

RAPPORT - TP 1 :

Compréhension et Visualisation des Données

Enseignants :

Shengrui Wang

Etienne G. Tajeuna

Auteurs :

Abdoul Madjid SANOUSSI LABO (CIP : SANA2909)

Abdou Rahime DAOUDA (CIP: DAOA2504)

Mohamed BOUBACAR BOUREIMA (CIP: BOUM3688)

Table des Matières

Présentation 2

Exploration et statistiques descriptives de données 2

Méthode 1 : Sans visualisation des données (Utilisation de mesures de distance) 5

1. *Choix des variables à utiliser (cas de mahalanobis)* 6
2. *Calcul des distances intra-classe et inter-classe* 7
3. *Les résultats* 8

Méthode 2 : Avec visualisation des données (Utilisation des graphe et réduction de dimension) 11

1. *Les visualisations* 11

a) *Distributions et Nuages des points des différentes classes (coloriés par classe) avec les deux variables choisies (gene_1 et gene_2)* 11

b) *Distribution conjointe des paires de classes pour les deux variables (gene_1 et gene_2)* 13

2. *Reduction de dimensionnalité (ACP, TSNE et UMAP)* 17

Conclusion 18

Présentation

Le présent document a été rédigé dans le cadre de l'exploration du thème « Sciences de Données ». Ce document présente la mise en pratique de l'essentiel sur la compréhension, la visualisation des données et les autres modules du thème. Notre rapport se base sur les matériaux présentés en cours et les documents qui ont été mis à notre disposition.

Exploration et statistiques descriptives de données

L'exploration et les statistiques descriptives sont des étapes cruciales pour mieux comprendre nos données avant de prendre des décisions plus avancées concernant notre analyse. Elles nous permettent également de repérer d'éventuelles anomalies ou problèmes dans les données. (Voir notre fichier *Code - TP1 IFT799-Equipe 2.ipynb*)

	gene_0	gene_1	gene_2	gene_3	gene_4	.....	gene_20528	gene_20529	gene_20530
count	801.000000	801.000000	801.000000	801.000000	801.000000		801.000000	801.000000	801.000000
mean	0.026642	3.010909	3.095350	6.722305	9.813612		9.590726	5.528177	0.095411
std	0.136850	1.200828	1.065601	0.638819	0.506537		0.563849	2.073859	0.364529
min	0.000000	0.000000	0.000000	5.009284	8.435999		7.864533	0.593975	0.000000
25%	0.000000	2.299039	2.390365	6.303346	9.464466		9.244219	4.092385	0.000000
50%	0.000000	3.143687	3.127006	6.655893	9.791599		9.566511	5.218618	0.000000
75%	0.000000	3.883484	3.802534	7.038447	10.142324		9.917888	6.876382	0.000000
max	1.482332	6.237034	6.063484	10.129528	11.355621		12.813320	11.205836	5.254133

Cette sortie nous donne pour chacune des variables (20531 variables quantitatives) quelques mesures descriptives sur les 801 individus présents. Notamment, la moyenne qui est le centre l'ensemble des données (dont on aura à scinder en des centres par classe lors du calcul des mesures de cohésion et de séparation) par

exemple la variable **gene_0** avec une moyenne de **0.026642,** l'écart type qui donne une idée sur la variation des individus autour de la moyenne, **0.13685** comme valeur pour la variable **gene_0,** les min et max, qui sont les valeurs extrêmes pour chacune des variables, pouvant renseigner sur l'existence des valeur Abbrantes (si (max+min)/2 n'avoisine pas la moyenne, AN : pour gene_0 (0+1.482/2)= 0.741 >> 0.02, un facteur de de proportionnalité d'environ 28, laisse croire l'existence de quelque valeurs aberrantes pour la variable gene_0). Enfin, on dispose de la médiane et des valeurs interquartiles (pour gene_0 toujours, la médiane, qui divise les observations en part égale, est de 0 et l'intervalle interquartile est Q3-Q1 = 0-0 = 0. Ce qui implique une distribution asymétrique de la variable).

D'autres part, on s'est intéressé à la présence des valeurs manquantes, et on n'en trouve aucune présente sur ce jeu de donnée. Ce qui est plutôt intéressant pour l'analyse qui suit.

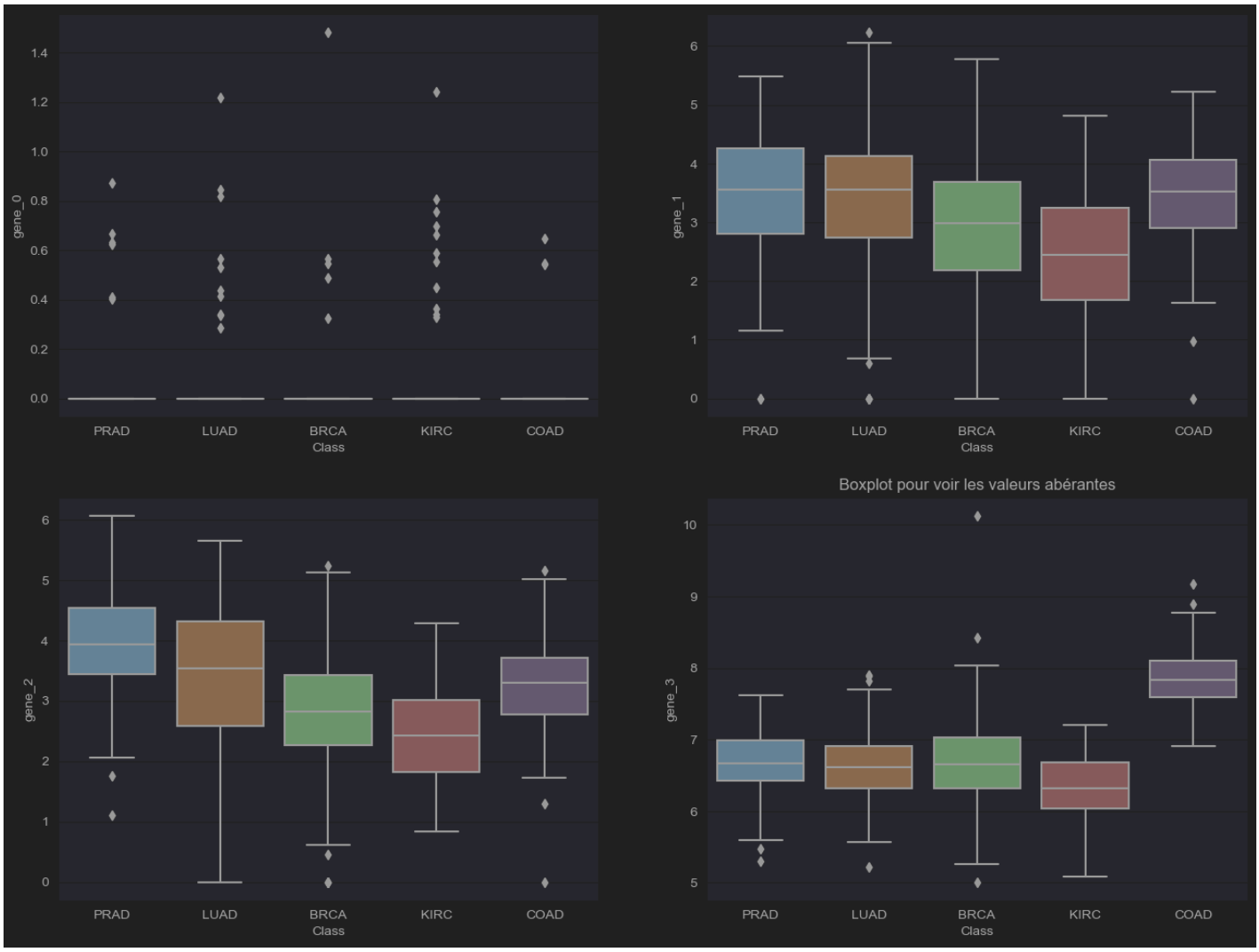
Jeu de données : Data

5 rows × 4 columns pd.DataFrame

	Types	Nb_Uniques	Nb manquants	Ratio manquants%
Unnamed: 0	object	801	0	0.0
gene_0	float64	36	0	0.0
gene_1	float64	770	0	0.0
gene_10	float64	415	0	0.0
gene_100	float64	799	0	0.0

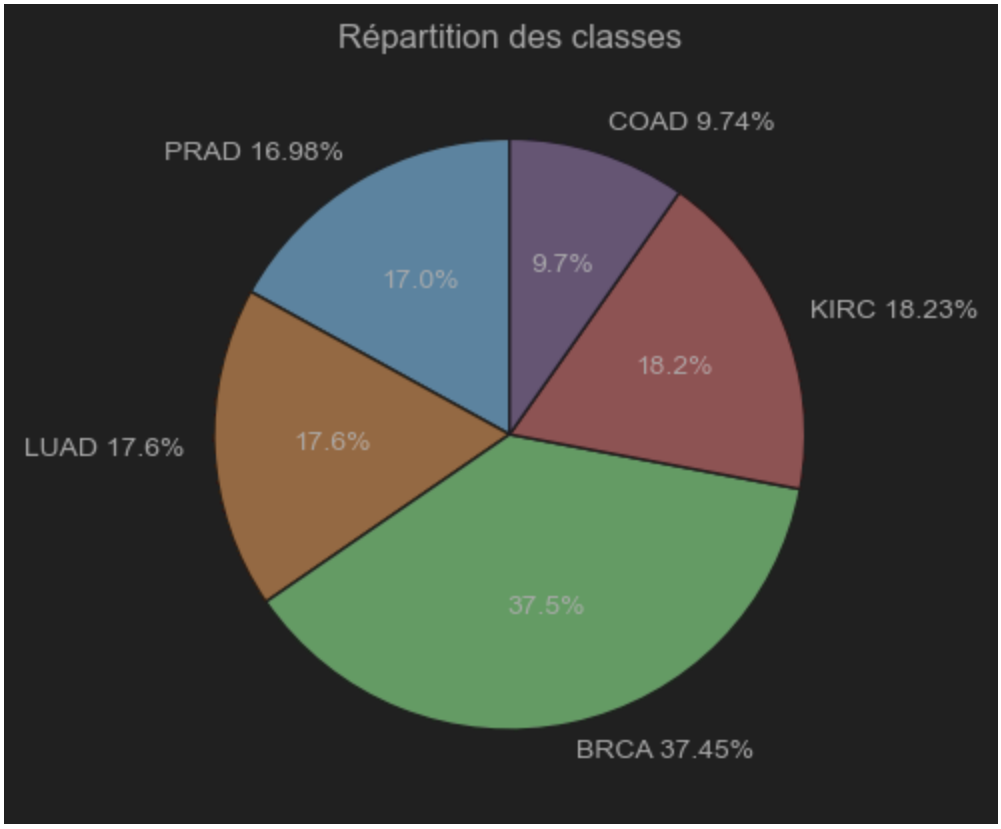
```
Total ratio =  0.0
Total valeurs manquantes =  0
Nombre d'occurence de la valeur 0 pour la variable gene_0 =  35
```

Cependant, diverses variables, contienne beaucoup de 0 sur les 801 individus, ce qui pourrait causer problème pour l'étude. gene_0, seulement 35 individues ont une valeur non nulle. Ce qui justifie un groupe d'outliers. On peut les visualiser avec un boxplot, comme suit :



gene_0, comparativement à **gene_1, gene_2 et gene_3**, ne contient que des valeurs aberrantes, en plus d'une distribution centrée a 0. Pour ce qui est des autres variables **gene_1, gene_2 et gene_3,** la distribution semble moyennement symétrique et par classe et par individus.

Il serait également important de connaitre la distribution des classes présente, tout autant que leur nombre. Le camembert ci-dessous donne un aperçu sur cette connaissance des étiquettes.



BRCA est majoritairement présente dans notre jeu de donnée (37,45%), et inversement COAD, qui est minoritaire (9.74%). Les données ne sont pas balancées, une classe pourrait être discriminer, lors de sa prédiction, base sur un modèle conçu sur cette base de donnée.

Méthode 1 : Sans visualisation des données (Utilisation de mesures de distance)

Cette méthode évalue la séparation entre les classes de cancers en calculant les distances intra-classe et inter-classe, puis en utilisant l'indicateur Overlap pour quantifier la séparation. Une valeur d'Overlap inférieure à 1 indique une bonne séparation entre les classes. Cette approche nous permettra de déterminer si les classes de cancers sont bien distinctes les unes-des-autres en fonction des profils génomiques des patients. 3 types de mesure de distances seront explorées, à savoir la distance Euclidienne, la Distance de Cosinus et la distance de Mahalanobis.

1. Choix des variables à utiliser (cas de Mahalanobis)

Afin d'éviter la rencontre des matrices de singularités, on est amené à réduire notre jeu de donnée à un jeu plus restreint en variables.

La distance de Mahalanobis est définie comme suit :

$$dist(p, q) = (p - q)^T \Sigma^{-1} (p - q)$$

Avec Σ la matrice de covariance sur l'ensemble de données $X \in \mathbb{R}^{n \times d}$ calculée comme suit :

$$\Sigma = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \mu)(x_i - \mu)^T$$

Singularité : Lors du calcul de la distance de Mahalanobis, si la matrice de covariance est singulière (c'est-à-dire qu'elle a une déterminante égale à zéro), alors elle n'a pas d'inverse. Cela peut se produire lorsque certaines variables sont linéairement dépendantes les unes des autres, ce qui signifie qu'elles portent la même information. Pour résoudre ce problème, nous envisageons l'approches suivante :

Supprimer les variables redondantes : Identifiez les variables qui sont linéairement dépendantes et supprimez-en certaines. Nous pouvons utiliser des techniques d'analyse de la colinéarité, telles que la recherche de valeurs propres proches de zéro dans la matrice de covariance, pour identifier les variables à supprimer. Ceci revient presque à retenir les variables dont la variance est loin d'être nulle. Pour ceci, nous utilisons un test statistique, en utilisant la gaussienne, pour déterminer un intervalle de confiance de valeurs à retenir. Cette procédure, est plus expliquée en détail dans la section du code.

Sans trop approfondir les calculs, nous obtenons un nouveau jeu de donnée réduit à 95 variables, qui pourrait être consulte dans la section code.

Comme attendu, la variable gene_0 n'en fait plus partie, n'étant qu'en partialité nulle (Potentielle corrélation élevée entre une variable qui tend vers 0, avec toutes les autres variables).

Remarque : la Class COAD toutefois, n'échappe pas à la singularité. Une des variables (gene_4353) de ces données contient que des 0. Ce qui revient au problème de singularité malgré la réduction des variables. D'autre variables, tendent

vers 0 également, donc colinéaires aux autres variables. Nous allons alors supprimer gene_4353, pour espérer éviter la singularité lors des calculs de distance de type Mahalanobis pour cette classe, et attribuer la valeur nulle (juste une annotation et non la valeur obtenue par calcul de la distance) si la singularité est obtenue, pour la suite de l'analyse.

2. Calcul des distances intra-classe et inter-classe

Pour chaque classe de cancer, nous calculons la distance intra-classe et inter-classe.

- Pour la distance intra-classe, il revient à calculer la distance maximale entre un patient d'une classe et le centre de cette classe (généralement la moyenne des profils génomiques). Formellement, étant donnée une classe $C1 = \{x1, x2 \ldots, xn1\}$ de n1 patients, la distance intra-classe ($\boldsymbol{dist_{intra(C1)}}$) est définie comme suit :

$$\boldsymbol{dist_{intra(C1)} = max\{mes(x_i, x_{ic1})|\forall x_i \in C1\}}$$

Avec $\boldsymbol{x_{c1}}$ le centre de la classe C1 (généralement représentée comme étant la moyenne).

- D'autre part étant donnée deux classes $C1 = \{x1, x2 \ldots, xn1\}$ et $C2 = \{x1, x2 \ldots, xn2\}$ de n1 et n2 patients, la distance inter-classe ($\boldsymbol{dist_{inter(C1,C2)}}$) est définie comme suit :

$$\boldsymbol{dist_{inter(C1,C2)} = min\{dist(C_1, C_2), dist(C_2, C_1)\}|}$$

Avec

$$\boldsymbol{dist(C_1, C_2) = min\{mes(X_i, X_{c2}) \mid \forall x_i \in C_1\}}$$

$$\boldsymbol{dist(C_2, C_1) = min\{mes(X_i, X_{c1}) \mid \forall x_i \in C_2\}}$$

$\boldsymbol{X_{c1}}$ et $\boldsymbol{X_{c2}}$ étant donné les centres respectifs des classes $\boldsymbol{C_1}$ et $\boldsymbol{C_2}$.

Et ***mes()*** est l'une des métriques suivantes : distance **Euclidienne**, distance de **Cosinus** et distance de **Mahalanobis,** car les résultats sont fonction de la mesure utilisée.

3. Les résultats

Les tableaux ci-dessous montrent les résultats quantitatifs obtenus à la sortie des fonctions de calcul des distances intra-classe et inter-class : On a en diagonale (en vert), les distances intra-classe c'est-à-dire pour la classe elle-même et les autres valeurs du tableau, correspondent à la distant inter-classe, entre les deux classes indexées.

➢ Distance Euclidienne :

	PRAD	LUAD	BRCA	KIRC	COAD
#Classes					
PRAD	239.574	186.998	177.901	226.669	218.541
LUAD	186.998	258.418	167.303	200.051	181.116
BRCA	177.901	167.303	256.959	215.223	197.589
KIRC	226.669	200.051	215.223	270.134	222.970
COAD	218.541	181.116	197.589	222.970	255.525

➢ Distance Cosinus :

	PRAD	LUAD	BRCA	KIRC	COAD
#Classes					
PRAD	0.025	0.014	0.013	0.021	0.020
LUAD	0.014	0.029	0.011	0.016	0.014
BRCA	0.013	0.011	0.029	0.019	0.016
KIRC	0.021	0.016	0.019	0.030	0.020
COAD	0.020	0.014	0.016	0.020	0.024

➢ Distance Mahalanobis :

	PRAD	LUAD	BRCA	KIRC	COAD
#Classes					
PRAD	11.229	26.889	15.816	33.684	0.000
LUAD	26.889	11.185	12.076	22.054	0.000
BRCA	15.816	12.076	14.262	14.100	0.000
KIRC	33.684	22.054	14.100	11.696	0.000
COAD	0.000	0.000	0.000	0.000	11.413

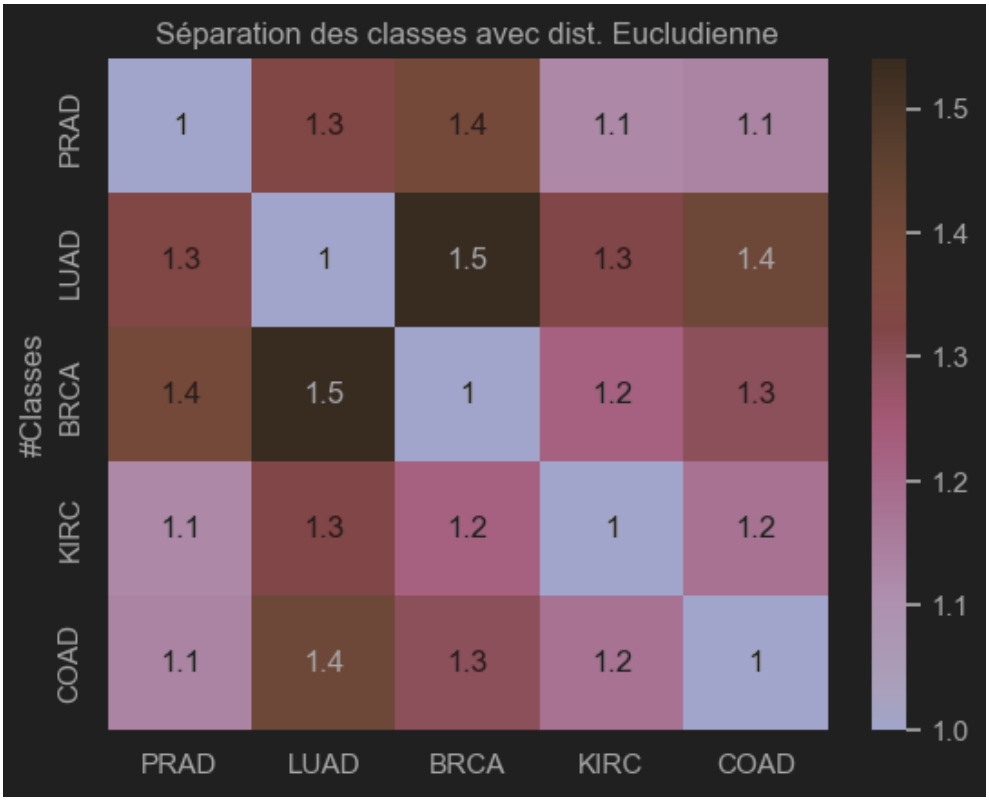
Ne donnant pas assez d'information sur la séparation des classes, un indice de séparation est mis en place pour évaluer ladite séparation : Le **Overlap.**

Dans cette première méthode, le test à faire pour confirmer la séparation entre les deux classes est de regarder à quel point les classes sont distantes entre elles. Overlap () de classes est défini comme suit :

$$Overlap(C_1, C_2) = \frac{dist_{intra}(C_1) + dist_{intra}(C_2)}{2 \times dist_{inter}(C_1, C_2)}$$

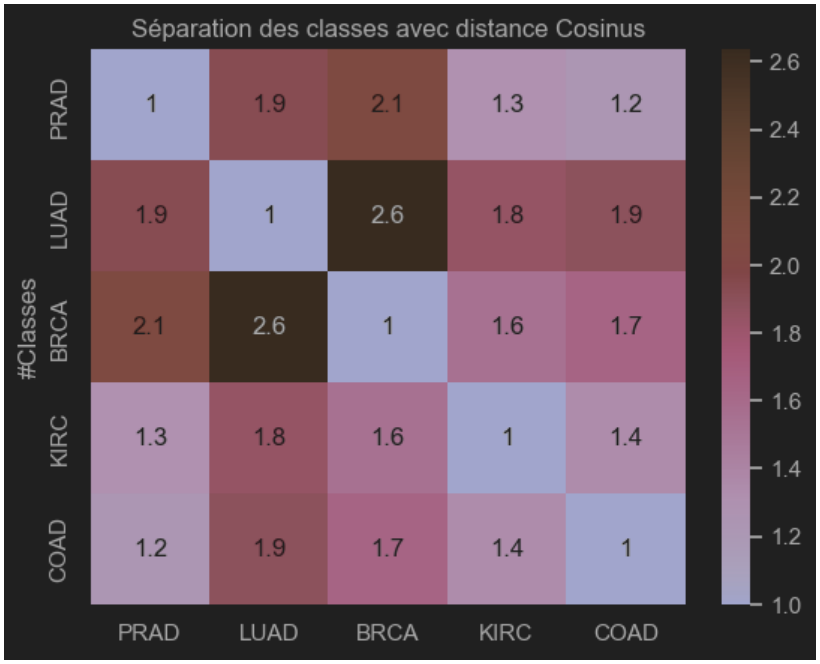
Si Overlap (C_1, C_2) < 1 on pourra dire que les classes C_1 et C_2 sont bien séparées.

- **Distance Euclidienne :**



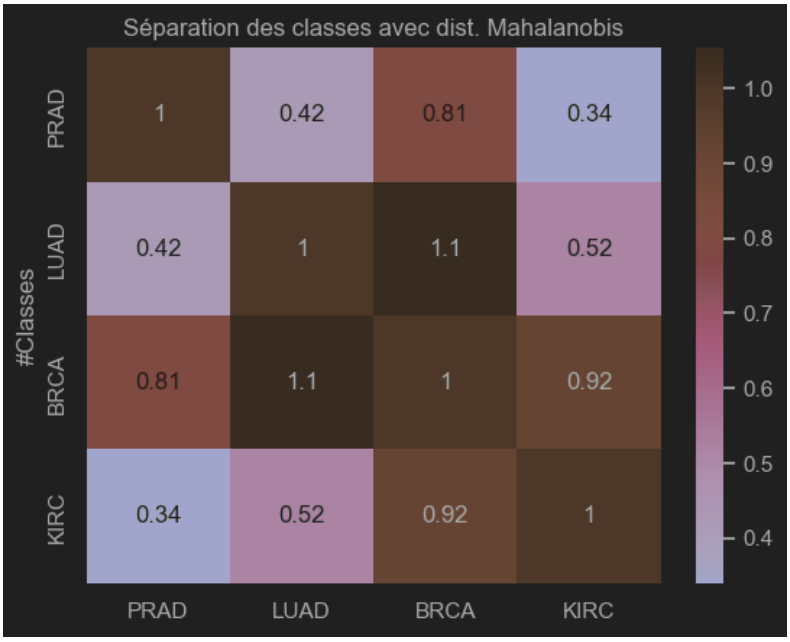
On remarque qu'aucune classe n'a été bien séparée, car toutes les valeurs sont supérieures à 1 pour les distances de type "Euclidienne".

➢ **Distance Cosinus :**



On remarque également qu'aucune classe n'a été bien séparée, car toutes les valeurs sont supérieures à 1 pour les distances de type "Cosinus".

➢ **Distance Mahalanobis :**



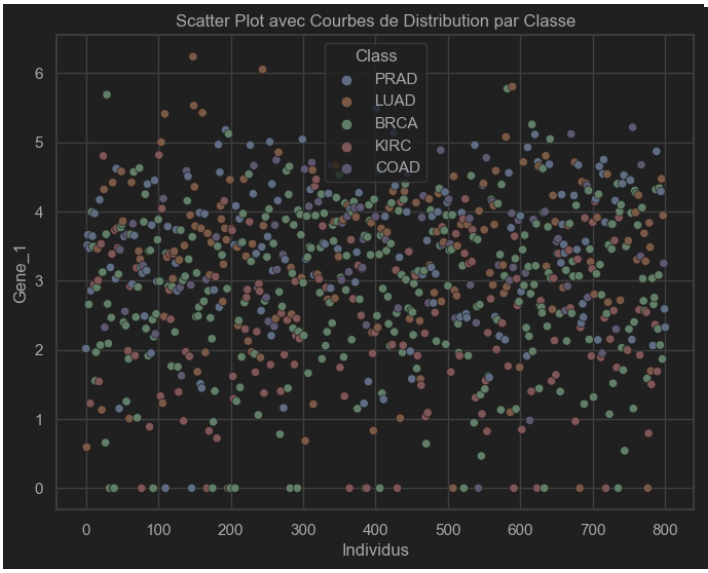
Sauf pour la classe COAD, où l'on a rencontré des singularités (incapacité d'évaluer la distance interclasse), la distance de Mahalanobis a permis une séparation nette de toutes les classes, sauf la séparation entre la classe LUAD avec BRCA. Ce qui est attendu, car ce type de distance, tient compte des déformation (matrice de covariance) de l'espace engendré par les observations.

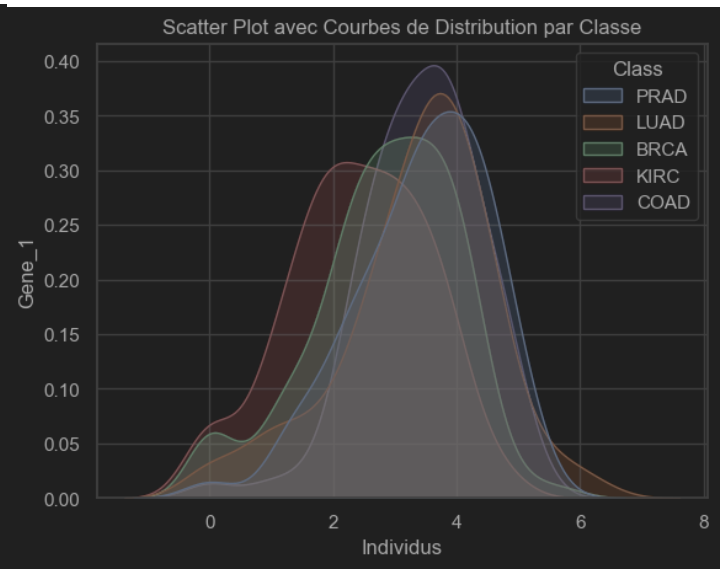
Méthode 2 : Avec visualisation des données (Utilisation des graphe et réduction de dimension)

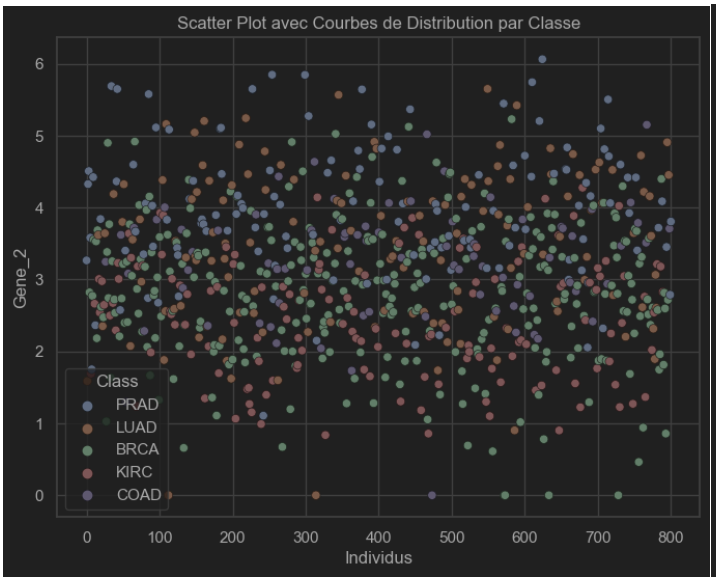
1. Les visualisations

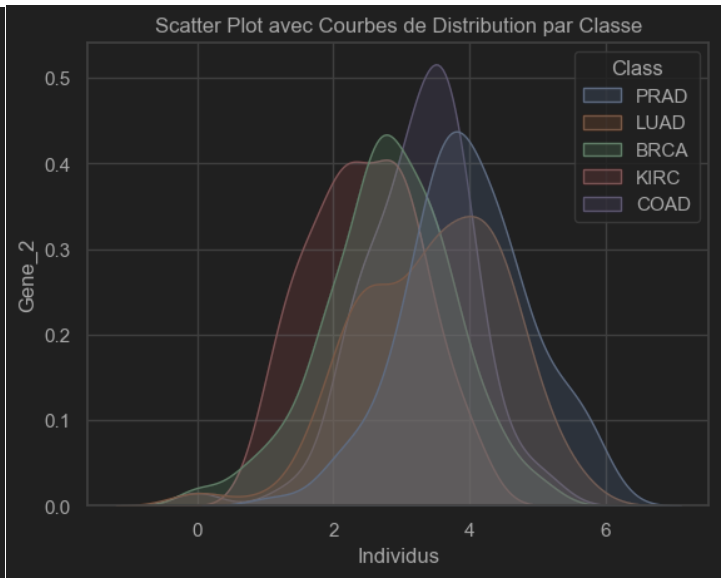
a) Distributions et Nuages des points des différentes classes (coloriés par classe) avec les deux variables choisies (gene_1 et gene_2)

- Une première visualisation, des distributions des classes et des nuages des points, séparément pour gene_1 et gene_2 :



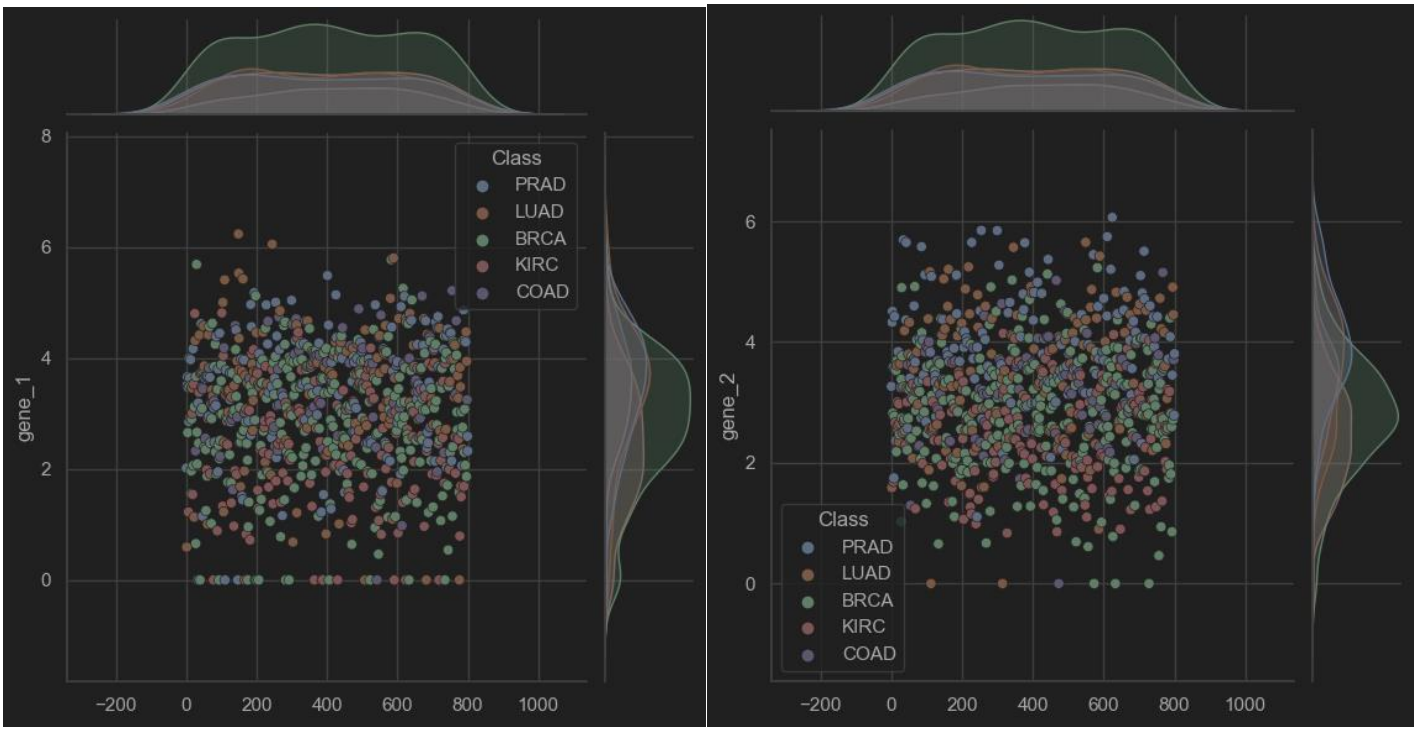




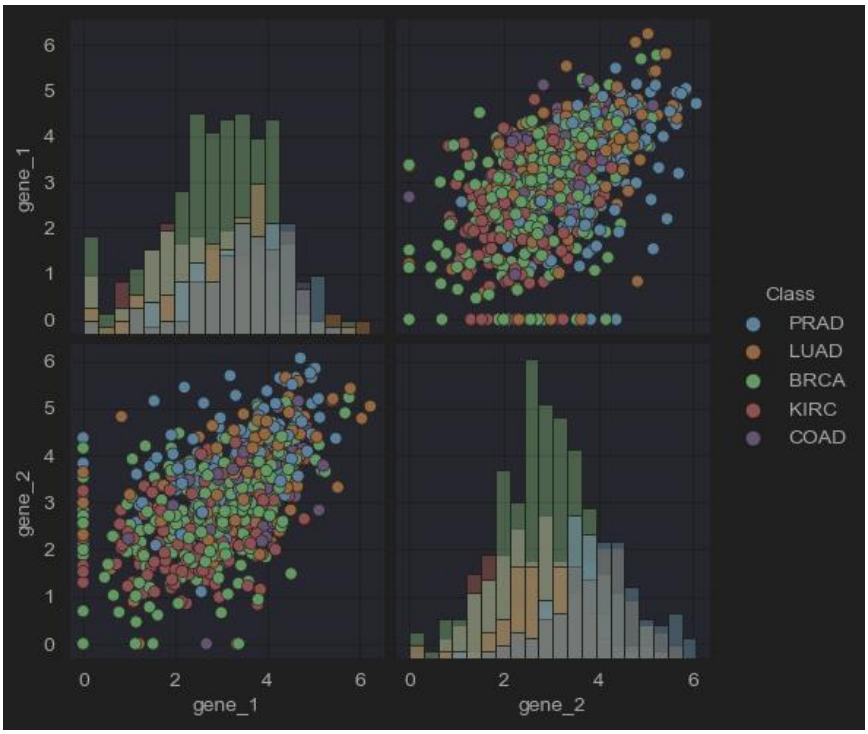


On remarque que gene_2 dissocie au mieux les classes, comparativement à gene_1. Et que certaines classes ont une distribution gaussienne (COAD, PRAD et BRCA), tandis que d'autres non (KIRC et LUAD). Cet aspect, pourrait nous conduire à l'utilisation de l'hypothèse de normalité, sur certaines variables, pour certaines classes principalement.

- Une seconde visualisation, en affichant les nuages et les distributions bilatéralement, en même temps.



- Puis, une troisième visualisation, avec les deux variables conjointement :

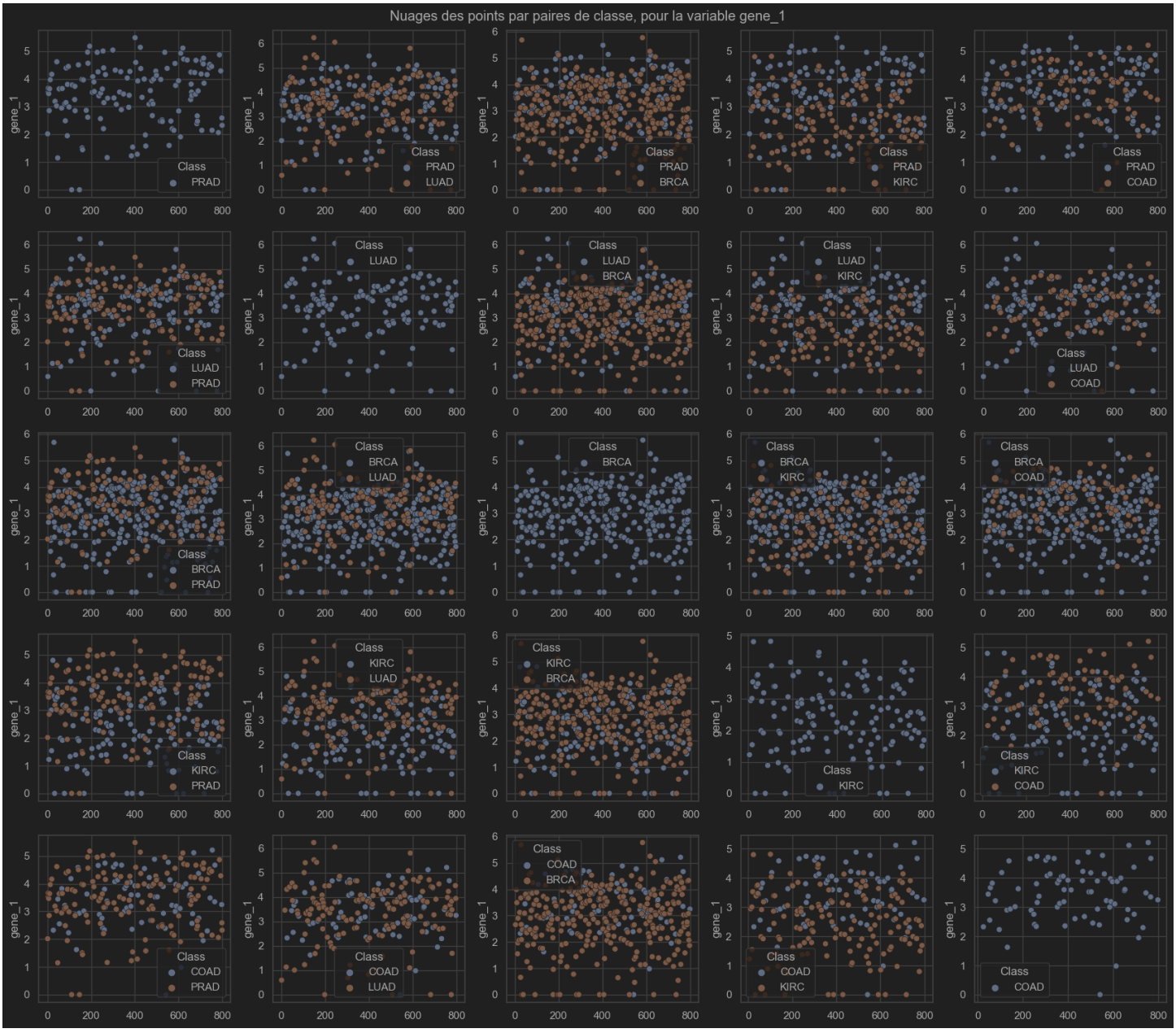


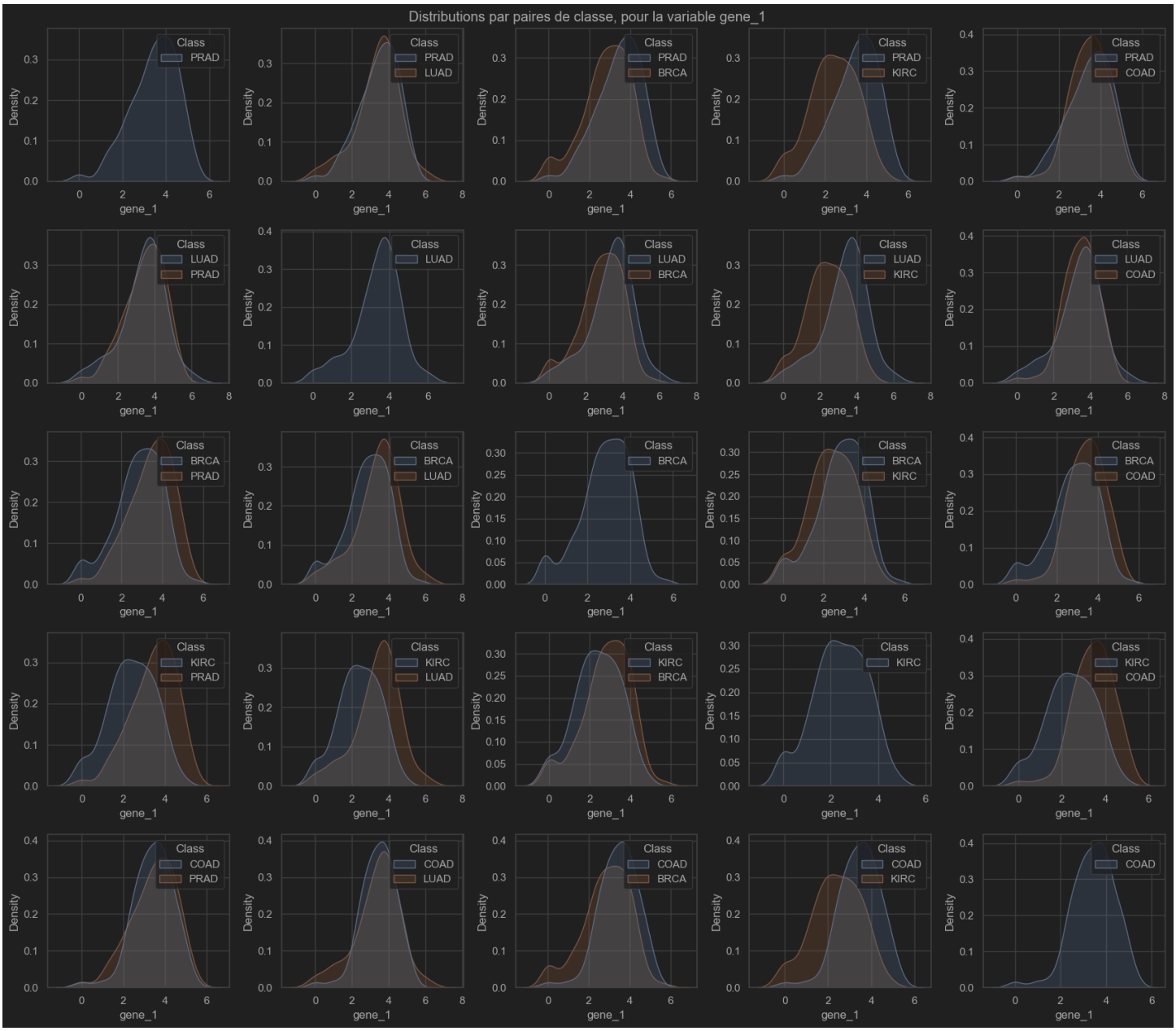
Qui est plus parlant, car il prouve l'existence d'une certaine dépendance linéaire entre gene_1 et gene_2, puis nous confirme l'aspect majoritaire de la classe BRCA, dans le jeu de donnée, ainsi que la normalité de certaines classes.

b) Distribution conjointe des paires de classes pour les deux variables (gene_1 et gene_2)

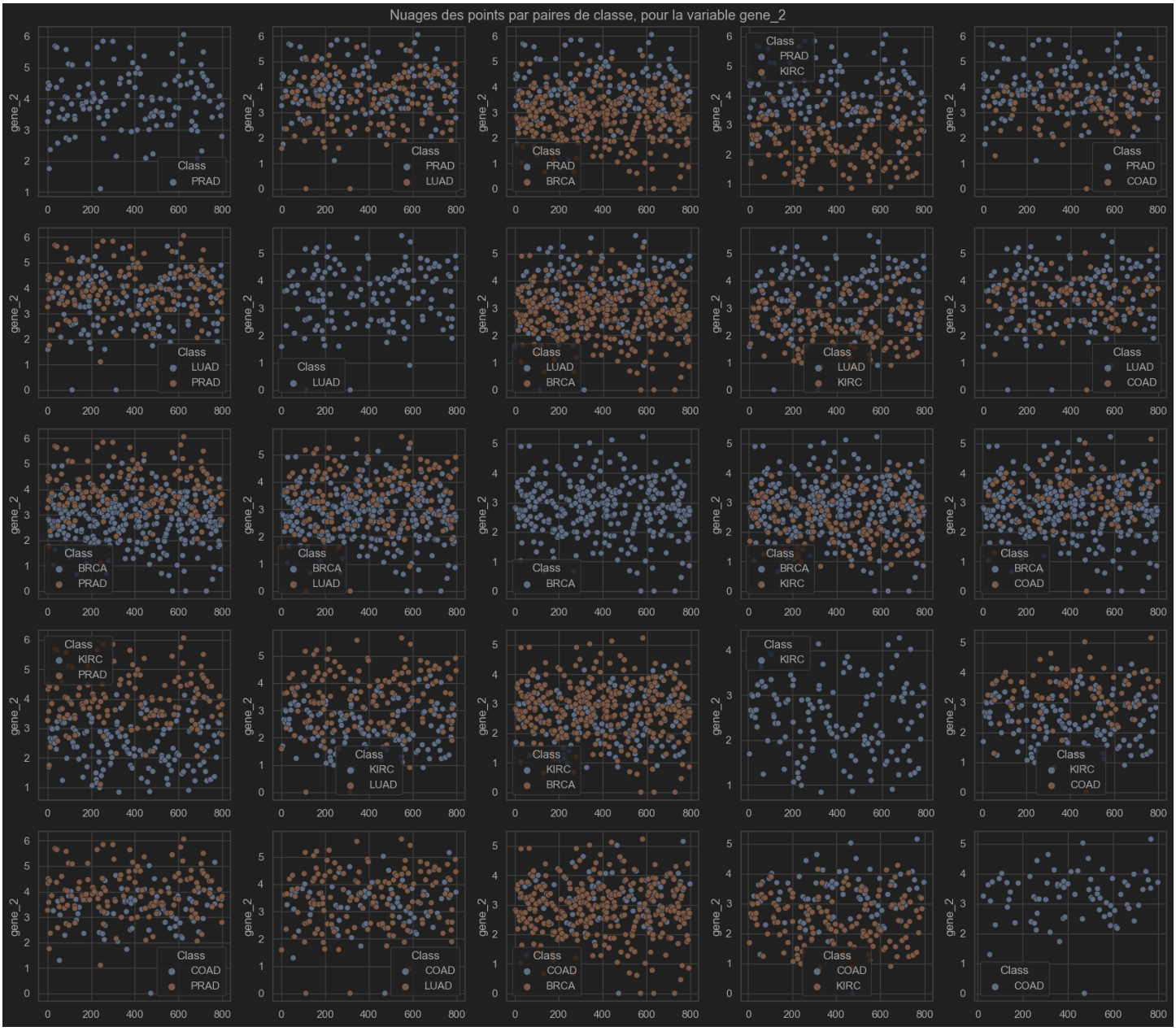
Afin de mieux constater la séparation des classes, il serait judicieux de les voir deux à deux. Ci-bas, les visualisations donnant les nuages des points ainsi que les distributions, pour chacune des deux variables.

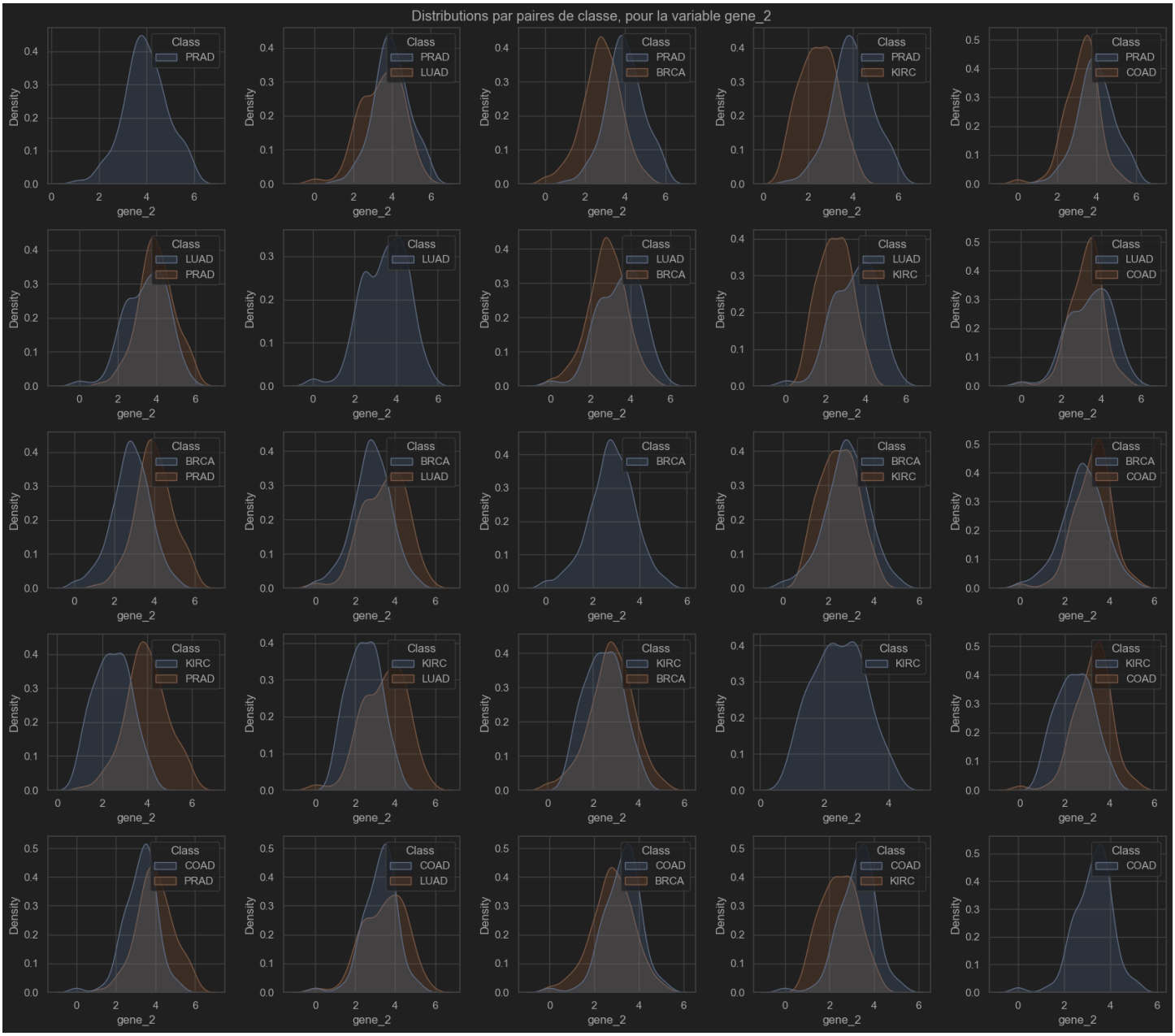
- **Gene_1 :**




Distributions par paires de classe, pour la variable gene_1
Class
PRAD
LUAD
BRCA
KIRC
COAD
Density
gene_1

- **Gene_2 :**


Nuages des points par paires de classe, pour la variable gene_2
gene_2
Class
PRAD
LUAD
BRCA
KIRC
COAD



Les figures précédentes, nous prouvent une séparation, malheureusement pas trop nette, mais quand même existante entre certaines paires de classe. Et surtout, plus visible, sur la variable gene_2.

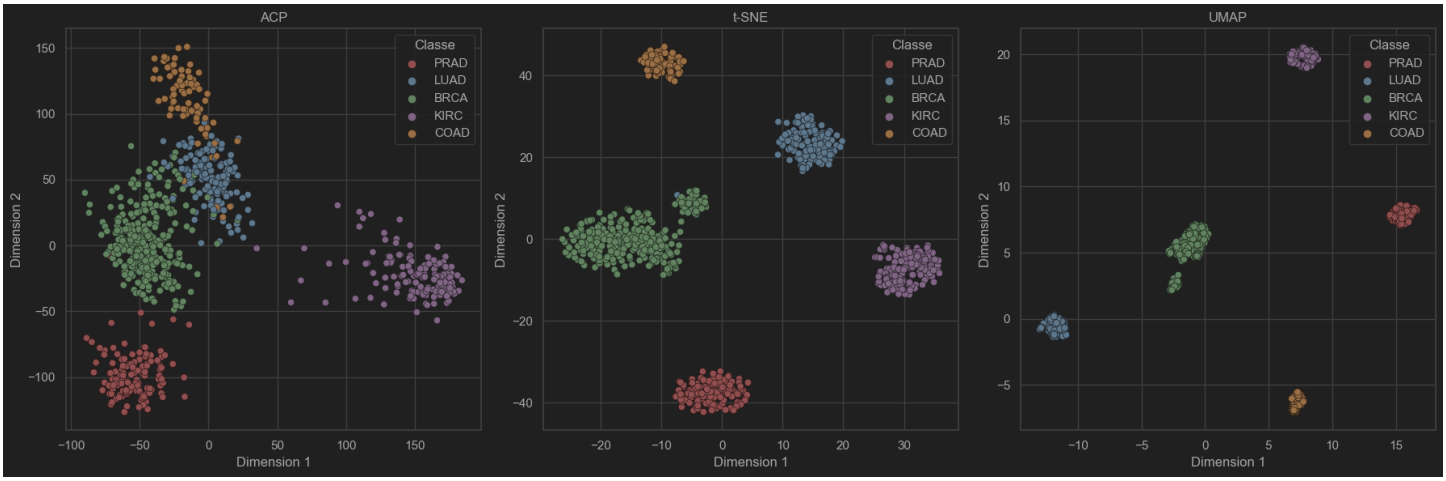
Sur cette variable (gene_2) par exemple, les paires de classe (BRCA, PRAD) et (KIRC, PRAD), ont une distribution conjointe avec une séparation, qui saute aux yeux, bien qu'elle possède une imprécision large (La probabilité d'appartenance aux deux classes en même temps, est signifiante).

Afin de bien pouvoir observer la séparation mutuelles des classes, il serait utile d'effectuer un changement (Une transformation) du repère. Autrement dit, il faudrait reprojeter les données, dans un nouveau repère, où les données sont séparable. C'est l'objet de la réduction de dimensionnalité.

2. Reduction de dimensionnalité (ACP, TSNE et UMAP)

Au cours de cette étape, nous effectuerions une réduction de dimensionnalité. Dans le jeu de données fourni, nous avons beaucoup trop de variables (attributs) qu'il serait fastidieux de trouver une bonne combinaison de deux ou trois variables permettant de bien visualiser la séparation des différents types de cancers. Pour cette raison, la réduction du nombre de dimensions en utilisant des méthodes s'impose. 3 techniques à l'appuis **ACP, TSNE et UMAP,** dont nous nous en passons du fonctionnement mathématique.

On obtient une représentation des individus dans les nouveaux repères, formés des 2 premières composantes principales (Dimension 1 et Dimension 2) et selon les 3 techniques utilisées, dans les graphes ci-dessous :



Force est de constater que UMAP, effectue mieux la séparation de différentes classes, ensuite vient le TSNE, puis l'ACP. Mieux, l'UMAP, permet une séparation linéaire, car avec des droites, on pourra poser une règle de décision (par valeur) sur l'appartenance d'un nouvel individu, à une classe spécifique.

ACP, a quant à elle, a du mal à distinguer les classes BRCA, LUAD et COAD.

Pour ce qui est de TSNE, on peut refaire la même chose, mais en utilisant un ensemble de droite de décision. Selon la source, TSNE est une technique de réduction de dimensionnalité non linéaire qui est souvent utilisée pour la visualisation de données à haute dimension. Elle permet de réduire la dimension de l'espace des caractéristiques tout en préservant les structures de similarité ou de distance entre les données. Contrairement à des méthodes comme l'ACP (Analyse en Composantes Principales), TSNE tente de conserver la structure de similarité intrinsèque entre les données, ce qui en fait une technique efficace pour la visualisation de clusters ou de groupes de données.

Quant à l'UMAP, d'après la source, elle est non linéaire et s'inspire de t-SNE, tout en étant plus efficace en termes de calcul et plus stable pour de grandes quantités de données. UMAP utilise une méthode de gradient stochastique pour optimiser la topologie d'un graphe de voisinage basé sur la distance entre les points de données. Il vise à préserver les

relations de voisinage tout en réduisant la dimensionnalité, ce qui en fait un outil efficace pour la visualisation et la réduction de dimensionnalité.

Conclusion

En sommes, ce projet nous a permis tout d'abord d'analyser les données qui nous ont été fournies afin de déterminer dans quelle mesure séparer chacune des classes. Par suite de celle-ci, une première méthode nous a conduit aux calculs des différentes distances inter-classes et intra-classes avec les distances Euclidienne, Cosinus et Mahalanobis, puis, calculer le Overlap entre nos différentes classes.

La séparation des classes n'étant pas visible avec cette méthode, surtout pour les distances Euclidiennes et Cosinus, nous sommes passés à la méthode de visualisation. Avec cette méthode, nous avons tout d'abord limité notre visualisation à deux variables (gene_1 et gene_2). Ensuite, nous avons utilisé les transformations ACP, TSNE et UMAP pour la réduction des dimensions sur toutes les variables afin de mieux visualiser la séparation entre les différentes classes. De ces différentes transformations, il ressort que seule l'UMAP sépare au mieux distinctement les différentes classes; de ce fait, elle est la transformation efficace pour le traitement d'un grand jeu de données.