Compte-rendu du TP02 de SY09
Classification automatique

NGO Sy-Toan, Elliot BARTHOLME

11 mai 2017

1 Introduction

Comme beaucoup de méthodes d'analyse de données, la classification automatique a pour but d'obtenir une représentation simplifiée des données initiales. Elle consiste a réduire les données en les organisant dans un ensemble de classes homogènes ou naturelles.

2 Visualisation des données

Dans cette partie nous allons étudier les données que l'on utilisera pour les méthodes de classification dans la suite. Nous avons trois jeux de données qui sont les *Iris* de Fisher, les *Crabs* (étudié au TP01) et finalement les *Mutations*, qui représentent des dissimilarités entre espèces.

2.1 Les données Iris

Après avoir effectué une ACP sur les données, on remarque que les deux premiers axes ont une inertie expliquée cumulée de 97.77%, et le premier axe 92.46%, on ne pourrait donc pratiquement se contenter que de celui-ci pour projeter le nuage initial. On affiche donc les données dans le premier plan factoriel, tout d'abord sans tenir compte de l'espèce :

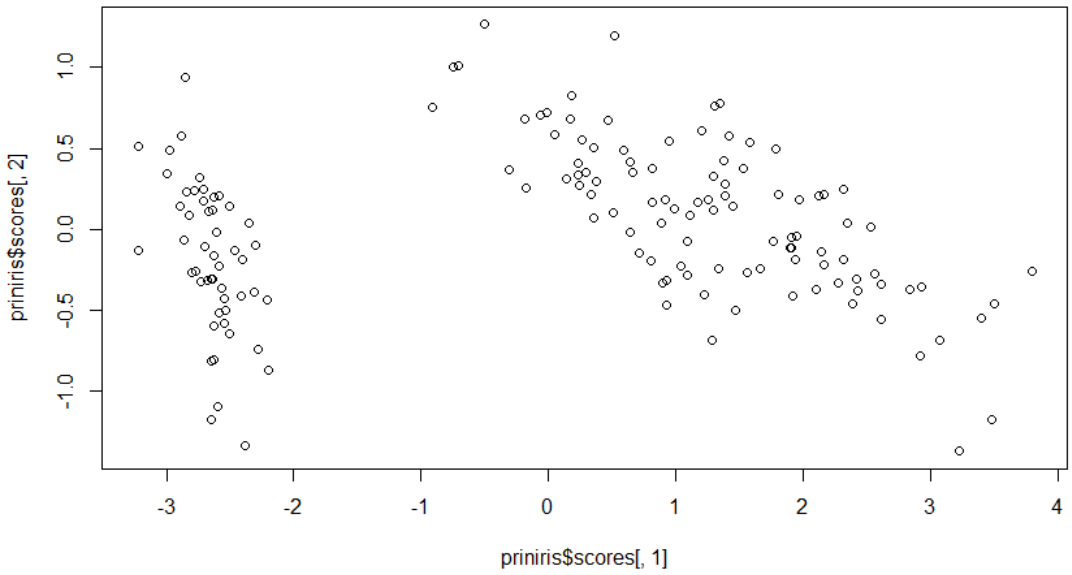

FIGURE 1 – Les données Iris dans le premier plan factoriel

Nous apercevons l'apparition de deux groupes d'individus, l'un a gauche, et l'autre à droite, dont le nuage est plus important. On voit bien aussi que le premier axe distingue ces deux groupes, l'axe 2 n'a pratiquement pas d'influence. Nous affichons maintenant la distinction selon les espèces :

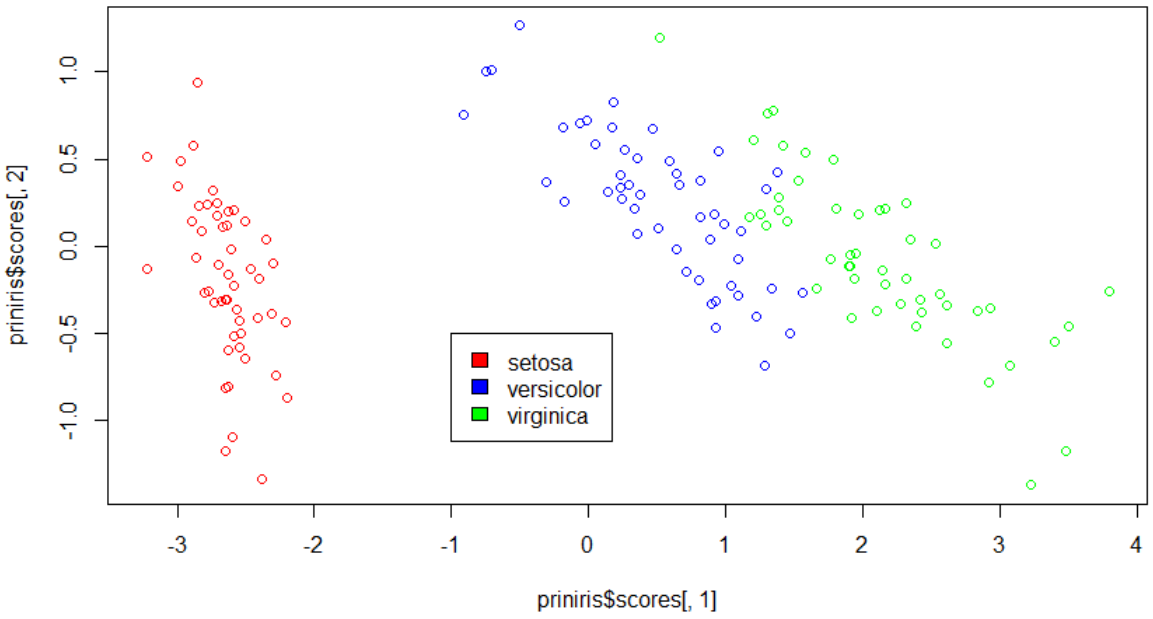

FIGURE 2 – Les données Iris dans le premier plan factoriel avec la couleur sur l'espèce

On constate qu'il y a à présent trois groupes de points. Si on recherche une partition des données, on s'attend a avoir une partition en 3 classes ; les deux classes *versicolor* et *virginica* sont plus rapprochées quand la classe *setosa* est plus éloignée.

Si l'on regarde (avec *biplot* ou numériquement) l'expression de la composante 1 par rapport aux variables initiales, on s'aperçoit qu'elle est égale à $0.36 * Petal.Width + 0.36 * Sepal.Length + 0.86 * Petal.Length$ c'est-à-dire que les espèces se distinguent très majoritairement par leur longueur de pétale.

2.2 Les données Crabs

Nous affichons les données dans le premier plan factoriel de notre analyse en composantes principales, tout d'abord sans tenir compte de l'espèce ou du sexe des crabes :

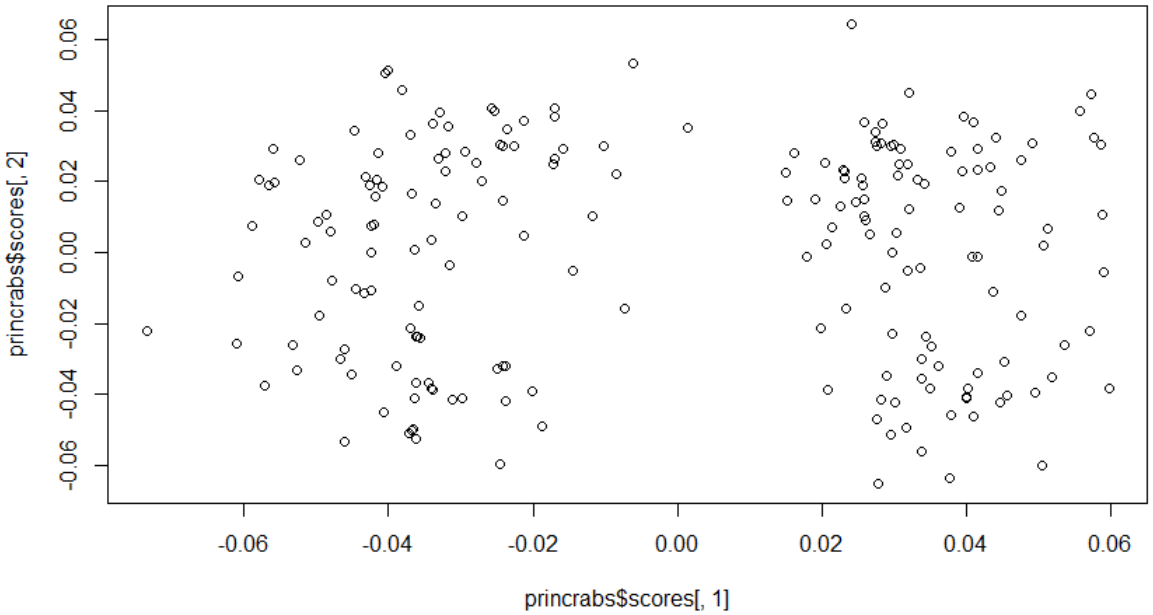

FIGURE 3 – Les données crabes dans le premier plan factoriel

On aperçoit 2 nuages distincts séparés par un espace vertical a la position 0 sur l'axe horizontal. Cette fois nous allons différencier les crabes selon leur espèce puis ensuite selon leur sexe.

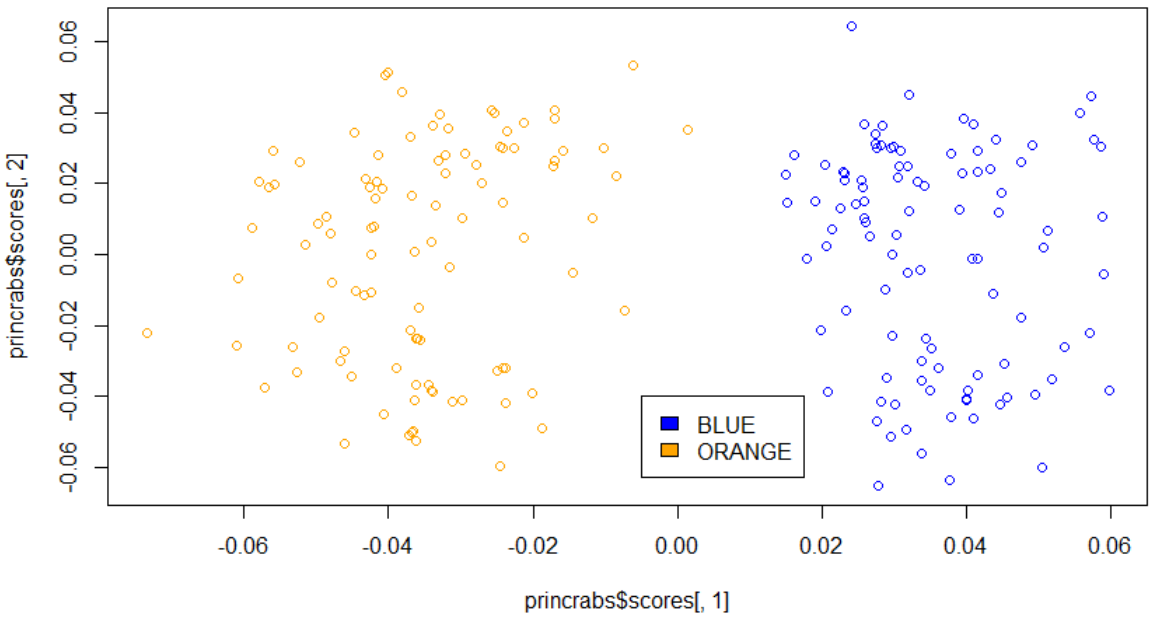

FIGURE 4 – Les données crabes dans le premier plan factoriel avec distinction sur l'espèce

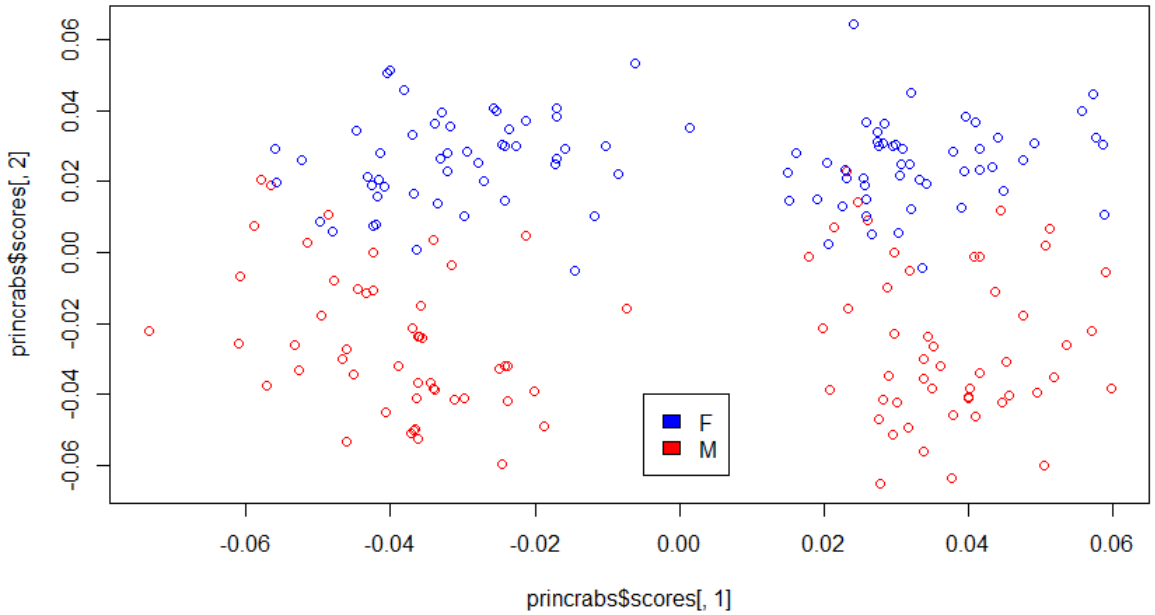

FIGURE 5 – Les données crabes dans le premier plan factoriel avec distinction sur le sexe

Lorsqu'on colore les points selon les espèces, on se rend compte que les deux groupes aperçus sur la figure 3 correspondent aux deux espèces (blue/orange). Cependant, lorsqu'on colorie en fonction du sexe, on voit que la séparation n'est cette fois-ci plus verticale mais horizontale, à la position 0 sur la composante 2. Si l'on examine le *biplot* de notre ACP, on confirme bien les résultats du TP01 :

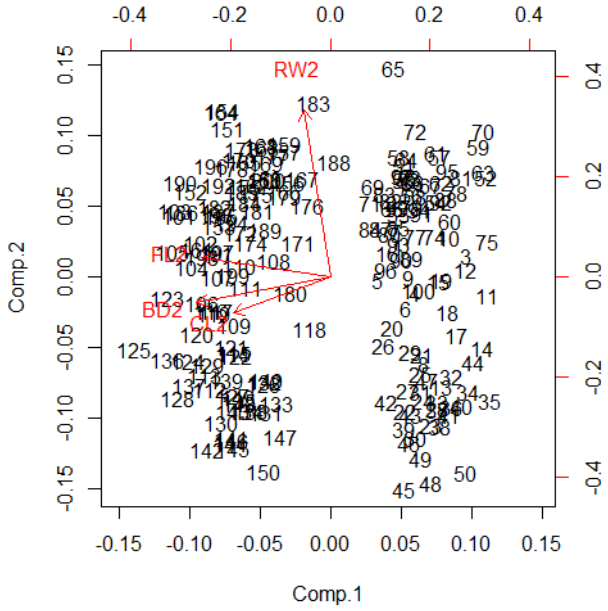

FIGURE 6 – Variables initiales en fonction des deux premières composantes principales

Ainsi, la composante 1 est combinaison des variables initiales caractérisant la taille du crabe (majoritairement *FL2,CL2, BD2*), et ce vers la gauche du graphique. Le nuage de point orange de gauche représente donc bien les crabes oranges qui sont plus gros en moyenne. De même, on voit que l'axe 2 est combinaison majoritairement de *RW2* vers le haut, donc les crabes femelles auront une largeur arrière en moyenne plus grande que les mâles.

Il est important de noter que l'on peut effectuer ces positionnements de nos nuages de points via d'autres méthode que l'analyse en composantes principales : l'analyse factorielle d'un tableau de distances en est une.

Les résultats sont quasi similaires à ceux obtenus via l'ACP, cependant cette méthode est plutôt utile lorsque l'on dispose uniquement d'un tableau de distances ou de dissimilarités entre individus (prochaine section). Voici le graphique obtenu via une AFTD sur les crabes, par espèce, après création d'une matrice de distances entre les individus :

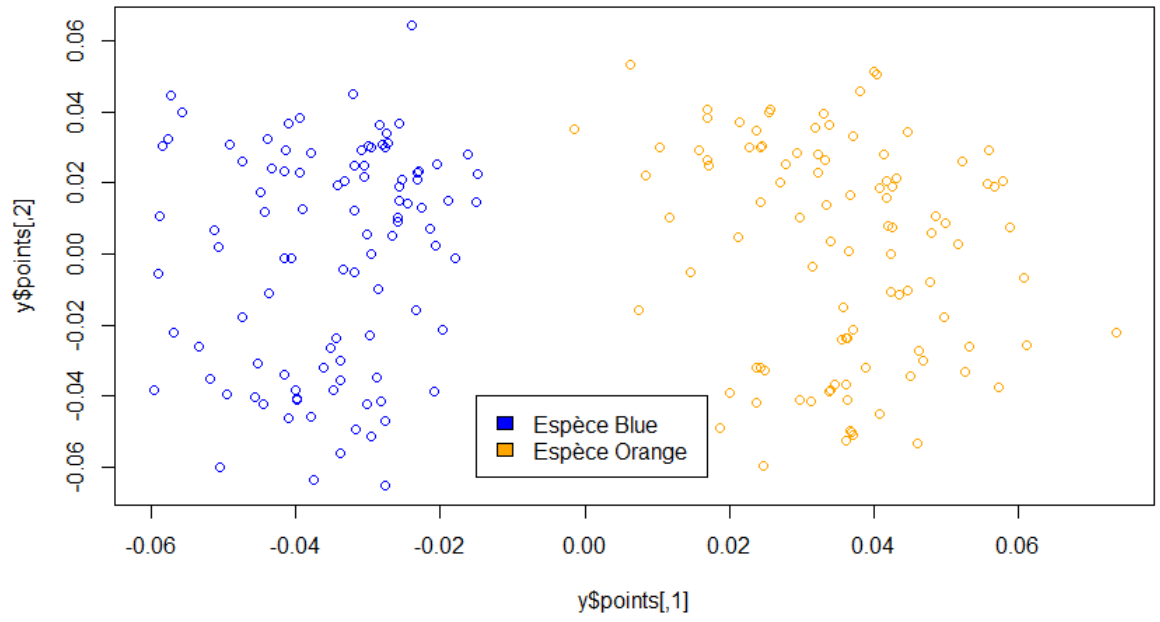

FIGURE 7 – Positionnement multidimensionnel avec un tableau de distances

Le nuage est identique à celui obtenu avec l'ACP, à une symétrie près, et permet bien à nouveau de distinguer les deux groupes d'individus selon les espèces.

Nous verrons systématiquement que les représentations obtenues avec l'AFTD ou l'ACP sont identiques à une isométrie près à chaque fois (rotation, translation, symétrie...). Les composantes principales obtenues avec les deux méthodes sont les mêmes, sauf celles du premier axe obtenu avec l'AFTD qui sont les opposés (valeurs négatives) de celles obtenues avec l'ACP (toujours sur le premier axe). C'est pour cela qu'ici le nuage orange est à droite et non à gauche.

2.3 Les données Mutations

Nous traitons à présent le jeu de données *Mutations*, qui recense les différences d'acides aminés à différentes positions d'une protéine commune à plusieurs espèces d'êtres vivants. Il se présente donc sous la forme d'une matrice où chaque valeur $a_{i,j}$ représente le nombre de positions sur la protéine ou les acides aminés ne sont pas les mêmes, entre les espèces i et j.

Cette fois, nous ne disposons donc que d'un tableau de distances entre les espèces. Ce tableau vérifie les propriétés d'une dissimilarité à savoir la symétrie ($d(x, y) = d(y, x)$) et une diagonale nulle ($d(x, x) = 0$).

L'analyse factorielle d'un tableau de distances (AFTD) qui permet d'obtenir une représentation des données

de dimension $p < n - 1$ fixée est donc la méthode que nous allons utiliser afin d'effectuer notre positionnement multidimensionnel.

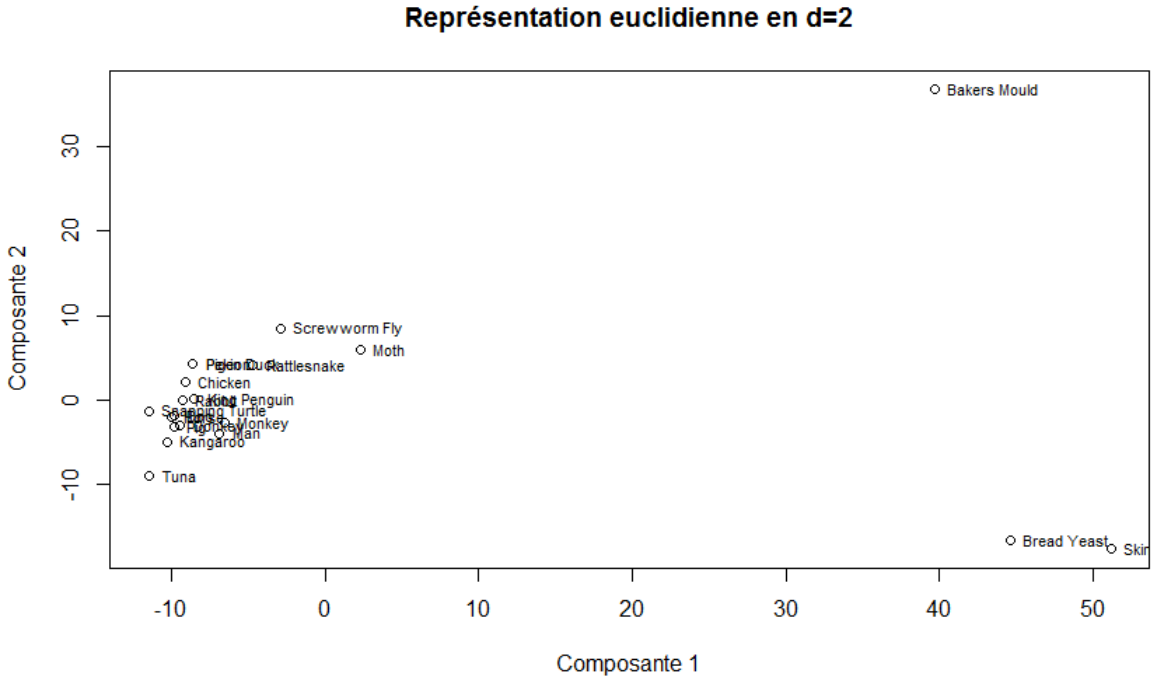

FIGURE 8 – Positionnement des espèces selon leurs distances les uns des autres dans le plan 1-2

On obtient un premier positionnement.
Cependant, la matrice des produits scalaires $W = -\frac{1}{2} Q_n \Delta^2 Q_n$ n'est pas semi-définie positive puisque des valeurs propres sont négatives. La matrice de dissimilarité fournie n'est donc pas euclidienne. Plusieurs stratégies sont alors envisageables :

1. conserver uniquement le sous-espace euclidien associé aux valeurs propres > 0
2. transformer par addition de constantes la matrice en matrice euclidienne
3. utiliser une méthode de positionnement multidimensionnel non-métrique ou non-linéaire telle que celle de Kruskal ou encore la projection de Sammon.

La seconde solution n'est pas intéressante car elle ne produit pas de bon résultats, surtout en raison de l'importante variance (valeur propre) des derniers axes résultants de l'AFTD, ce qui déforme beaucoup les distances entre espèces. Cette méthode a été testée via deux fonctions du package *ade4* de l'université de Lyon :

— *cailliez()* : calcule la plus petite constante positive qui permet de rendre euclidienne la matrice des distances
— *lingoes()* : calcule la plus petite constante positive k rendant la matrice $\sqrt[2]{({d_{ij}}^2 + 2 * k)}$ euclidienne

Conservation du sous-espace euclidien

On conserve pour cela uniquement les axes dont les valeurs propres sont positives, suite à la diagonalisation de la matrice $\frac{1}{n} * W$. Ici cela revient à éliminer les axes 15 à 20 obtenus après l'utilisation de *cmdscale* sur la matrice de distances des mutations.
On obtient alors une qualité de représentation assez faible en $d = 2$ car les deux premiers axes fournissent une inertie expliquée par rapport au nuage initial d'environ 70% seulement. Cela est confirmé par un diagramme de Shepard qui permet de visualiser l'écart entre les dissimilarités initiales et les distances obtenues par les méthodes de positionnement

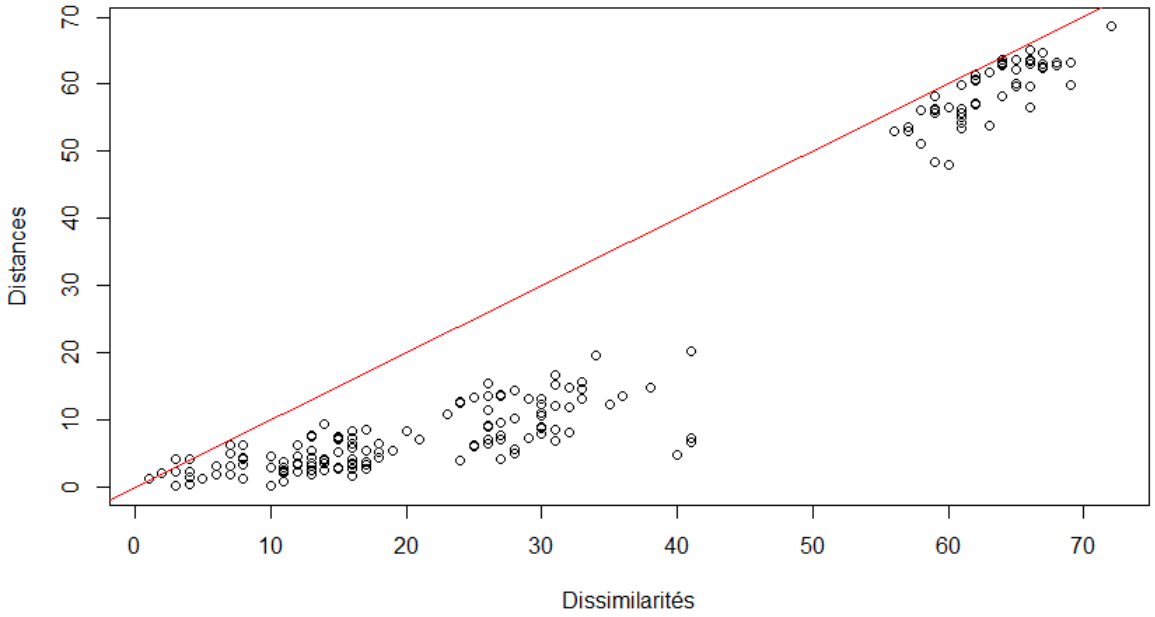

FIGURE 9 – Diagramme de Shepard entre les dissimilarités de mutations et distances obtenues via un AFTD avec les deux premiers axes conservés

On voit bien sur ce graphique que la relation entre les distances obtenues et les dissimilarités données par le jeu de données n'est pas linéaire (droite rouge). La plupart des points sont en dessous de cette droite de proportionnalité, ce qui signifie que les distances ont été largement sous-estimées par rapport aux réelles dissimilarités entre espèces, et que le positionnement obtenu dans la figure 8 n'est pas exact.

Nous recommençons à présent le même processus mais avec des représentations utilisant plus d'axes (jusqu'à 5 variables d de représentation). Nous ne pourrons évidemment en revanche utiliser que la projection dans un plan ($composante_1 - composante_d$) à chaque fois.

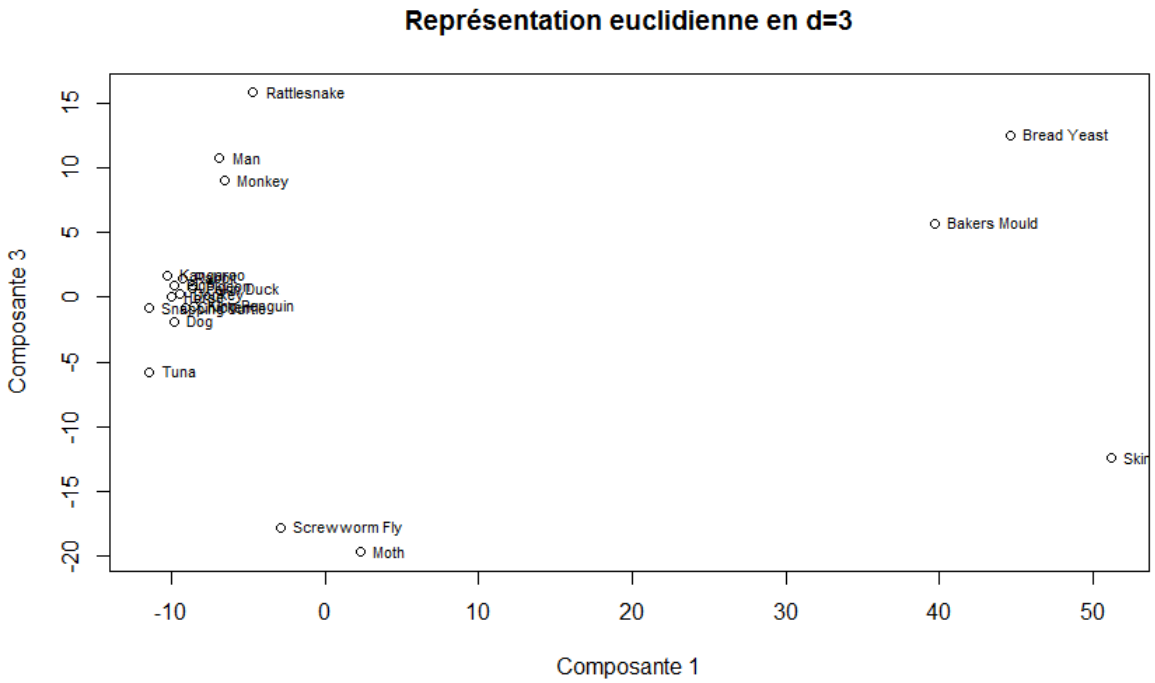

FIGURE 10 – Positionnement des espèces selon leur distances les uns des autres dans le plan 1-3

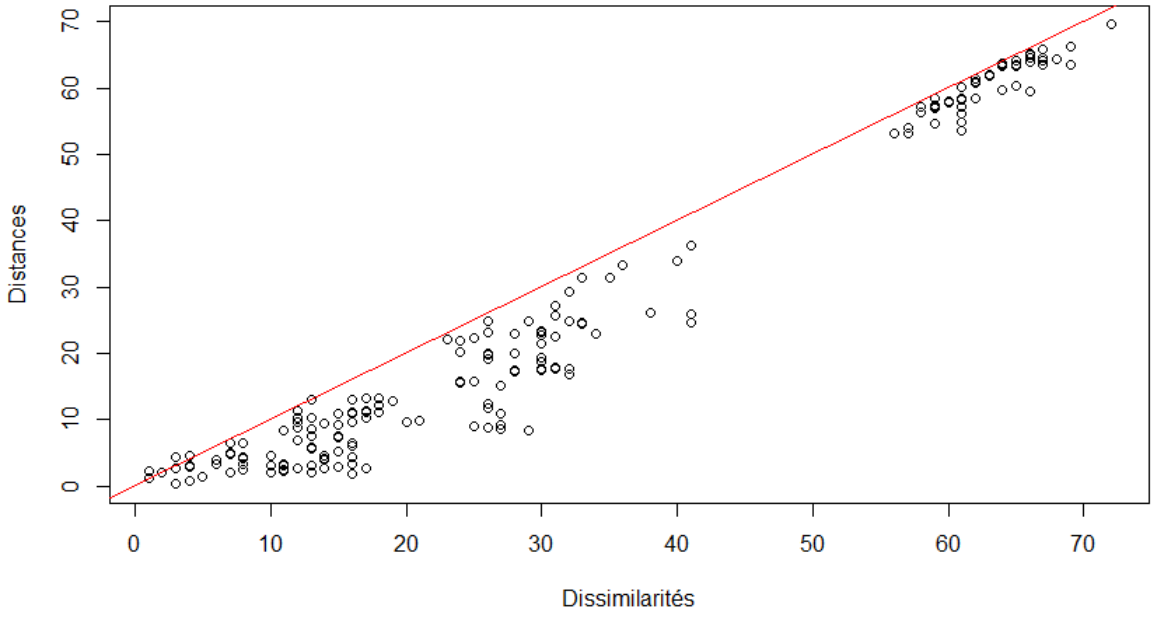

FIGURE 11 – Diagramme de Shepard entre les dissimilarités de mutations et distances obtenues via un AFTD avec les trois premiers axes conservés

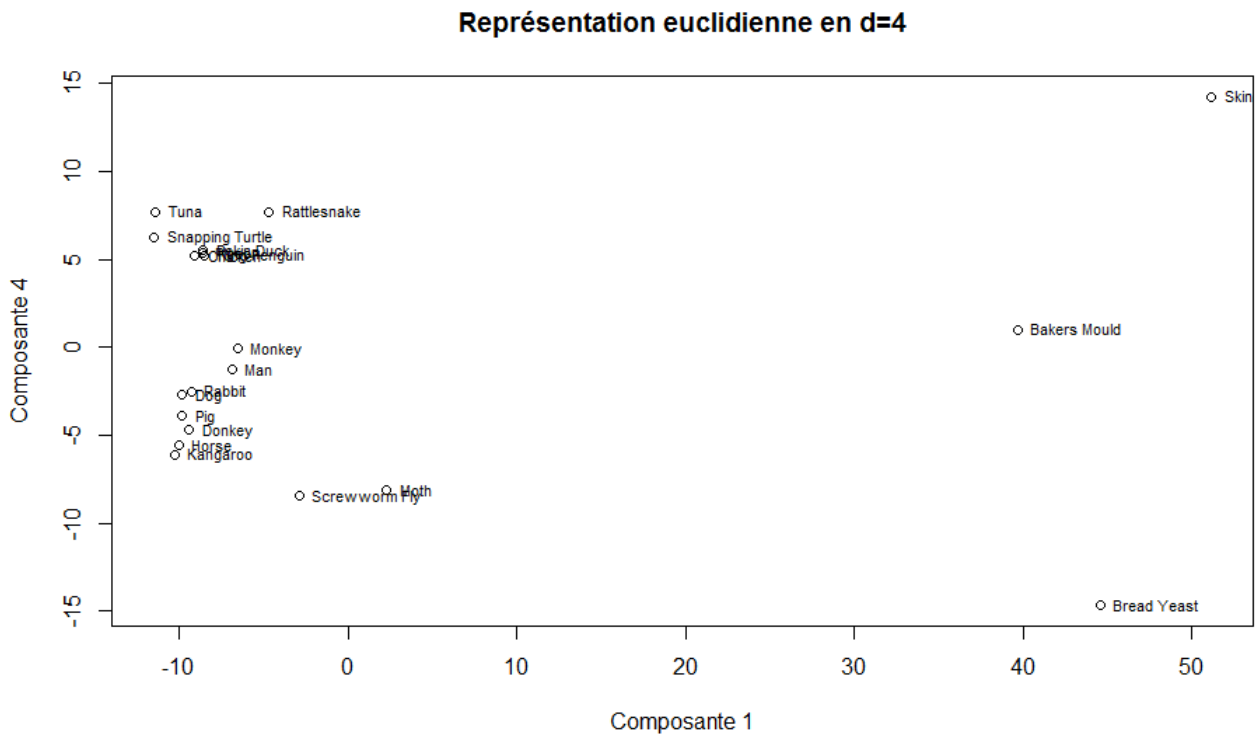

FIGURE 12 – Positionnement des espèces selon leur distances les uns des autres dans le plan 1-4

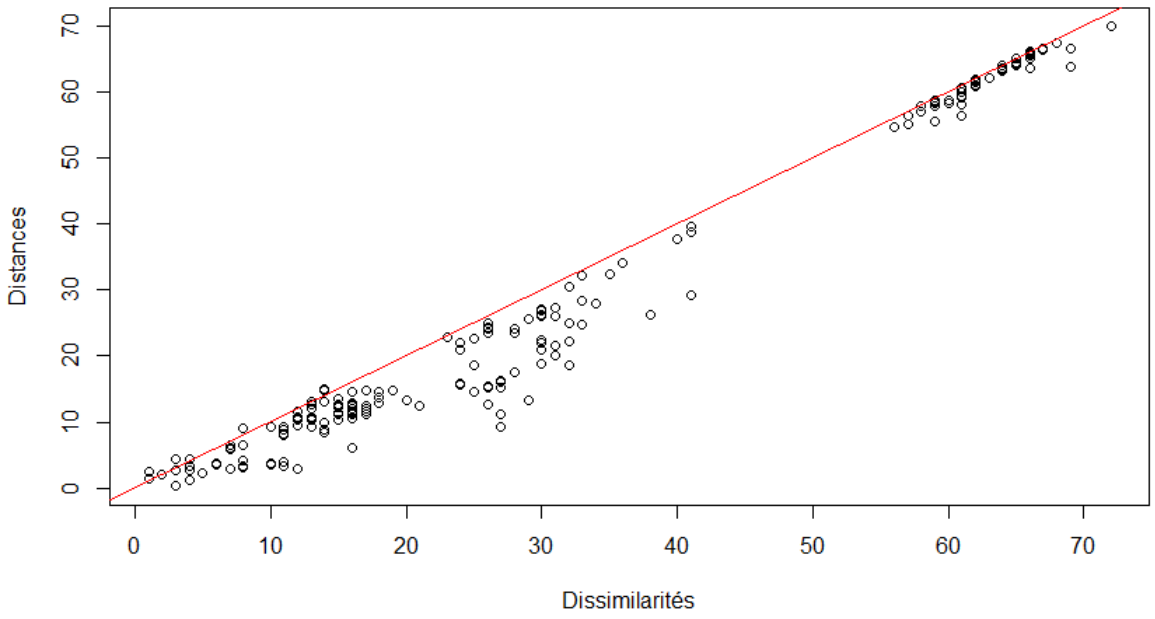

FIGURE 13 – Diagramme de Shepard entre les dissimilarités de mutations et distances obtenues via un AFTD avec les quatre premiers axes conservés

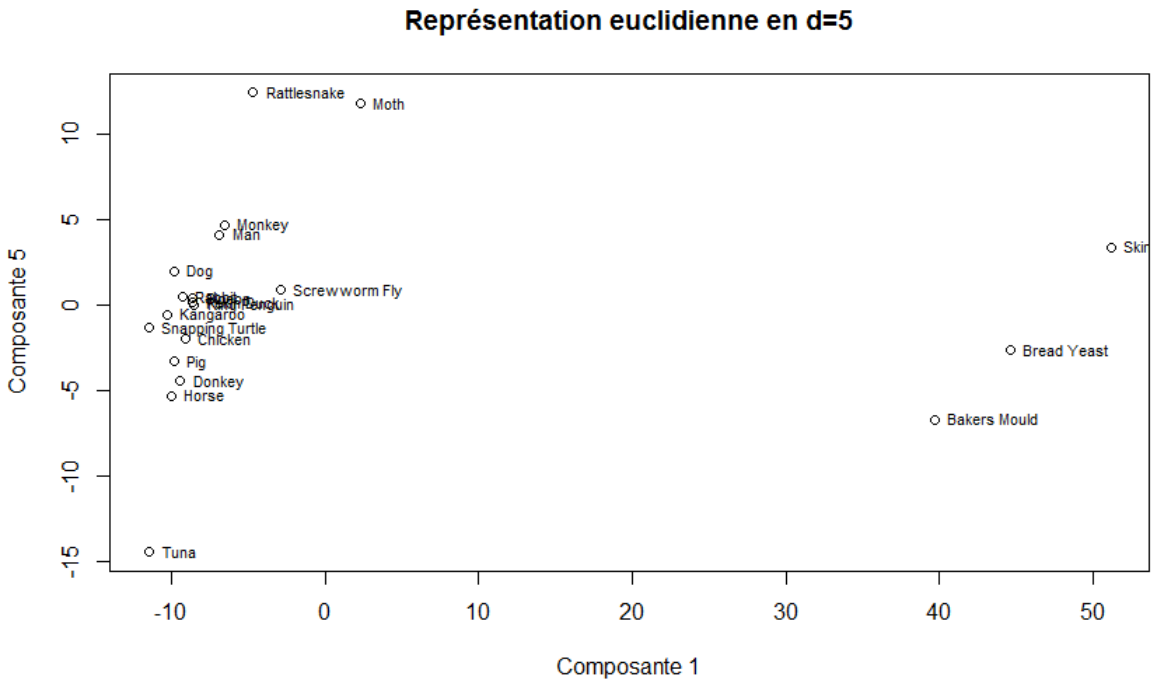

FIGURE 14 – Positionnement des espèces selon leur distances les uns des autres dans le plan 1-5

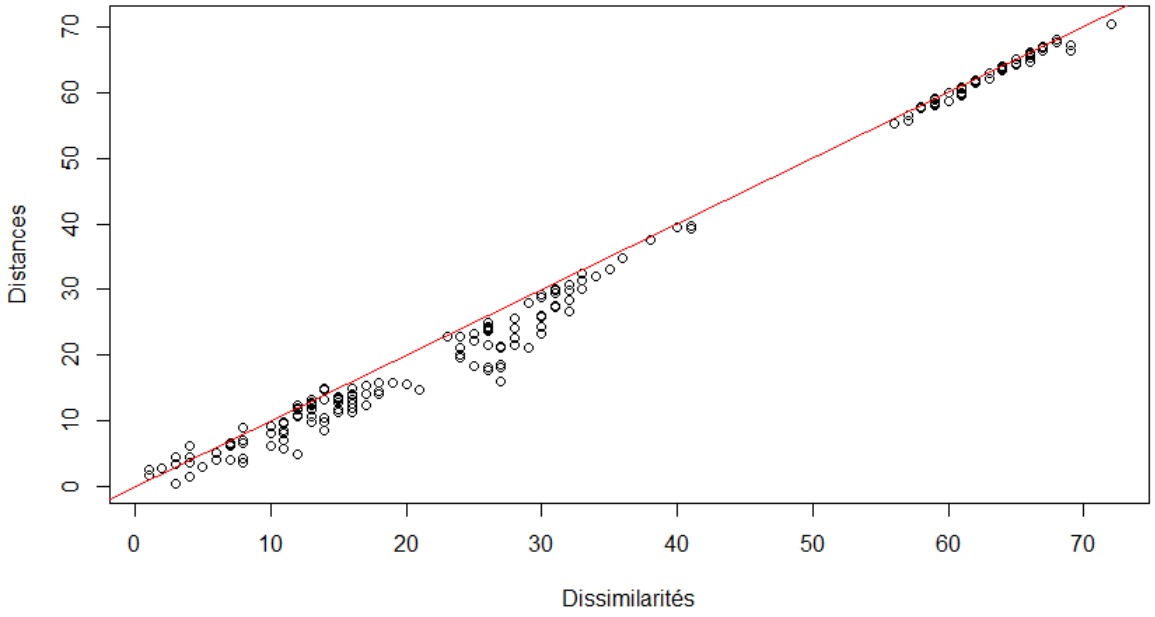

FIGURE 15 – Diagramme de Shepard entre les dissimilarités de mutations et distances obtenues via un AFTD avec les cinq premiers axes conservés

Nous voyons donc que, plus nous augmentons le nombre de dimensions avec lesquelles on représente notre nuage de points, plus le positionnement apparaît exact comparé aux dissimilarités entre espèces présentes dans nos données de base (les points s'approchent de la droite $y = x$). En d'autres termes, le positionnement est d'autant meilleur que nous conservons plus d'axes lors de la diagonalisation de la matrice des produits scalaires $W = -\frac{1}{2}Q_n\Delta^2Q_n = XX^T$. Cela s'explique par un gain d'inertie expliquée par le nuage de points initial (on obtient plus de 90% à partir de $d = 5$).
À titre d'information, voici le diagramme de Shepard pour $d = 14$ (dimension maximale de l'espace euclidien représenté par la matrice de dissimilarités) :

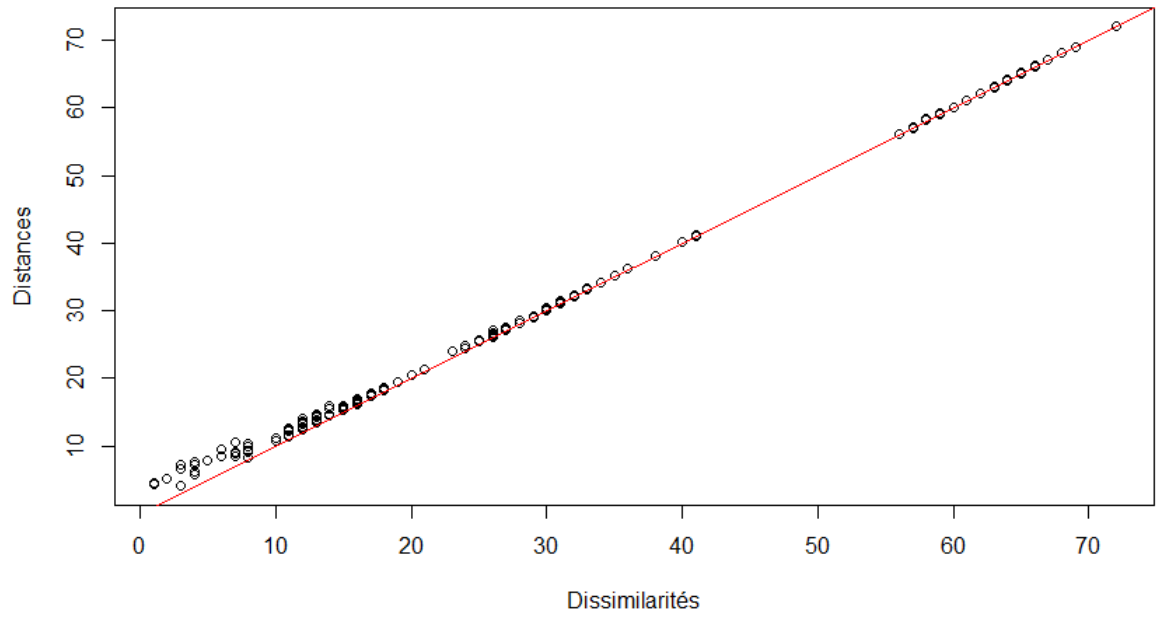

FIGURE 16 – Diagramme de Shepard entre les dissimilarités de mutations et distances obtenues via un AFTD avec les 14 premiers axes conservés

Méthodes non-linéaires et non-métriques : *Sammon* et *Kruskal*

Dans les méthodes non linéaires, on n n'imposera ici plus une projection linéaire des données de base dans un espace euclidien de faible dimension mais on cherchera plutôt à minimiser de manière itérative une fonction d'écart entre Δ (dissimilarités) et D (distances) que l'on appelle fonction *Stress*.

Les méthodes non-métriques de positionnement multidimensionnel consistent elles à étendre le principe précédant en utilisant des procédés et algorithmes itératifs qui se focalisent sur l'ordre des dissimilarités (ou similarités) plutôt que sur leurs valeurs absolues (ou relatives). On représente les différentes données en préservant l'ordre entre proximités en premier plutôt que selon leurs valeurs exactes. On cherche alors à minimiser une fonction *Stress* qui dépend des données X et d'une fonction f à déterminer.
L'acceptation de la représentation alors trouvée dépend des seuils définis mais on s'accorde généralement sur le suivant :

- $Stress > 0.20$: mauvais
- $0.10 < Stress < 0.20$: passable
- $0.10 < Stress < 0.025$: bien
- $Stress < 0.025$: excellent
- $Stress = 0$: représentation parfaite.

Sammon :
Après 40 itérations, pour une configuration en $d = 5$ dimensions, on obtient un stress de 0.00136, ce qui est excellent. La représentation est la suivante :

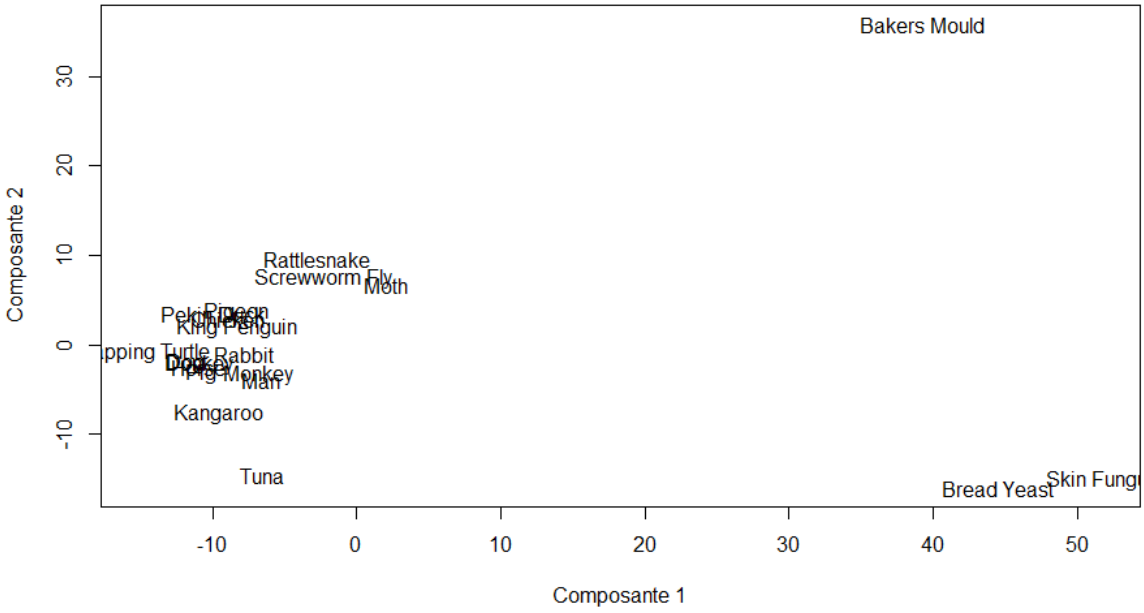

FIGURE 17 – Représentation sur l'axe 1-2 de la projection de Sammon en $d = 5$ dimensions

Voici le diagramme de Shepard associé :

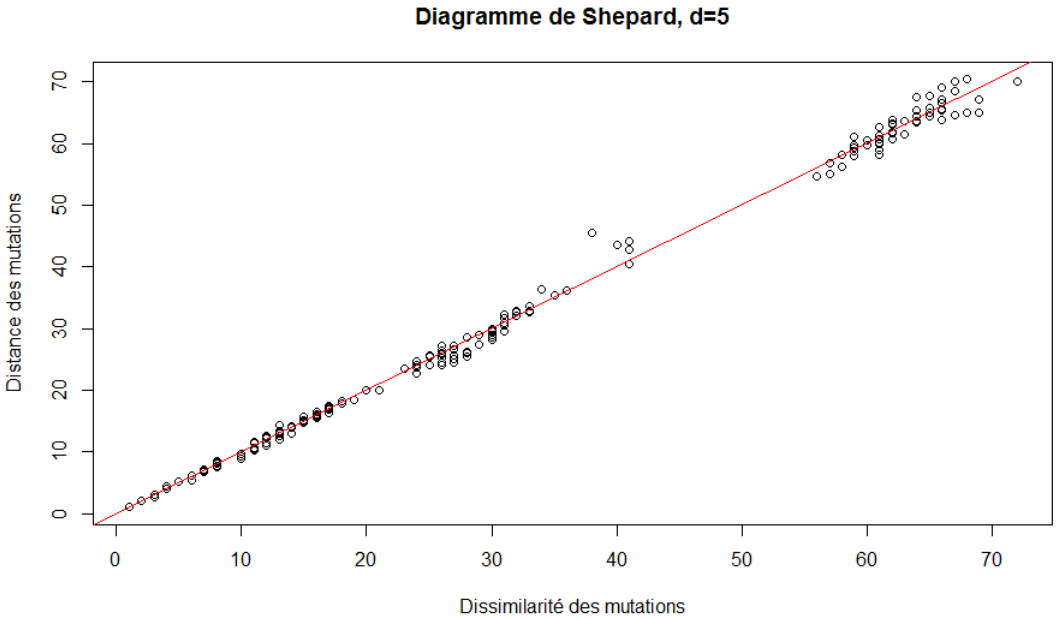

FIGURE 18 – Diagramme de Shepard avec 5 dimensions et la projection de Sammon

Kruskal :

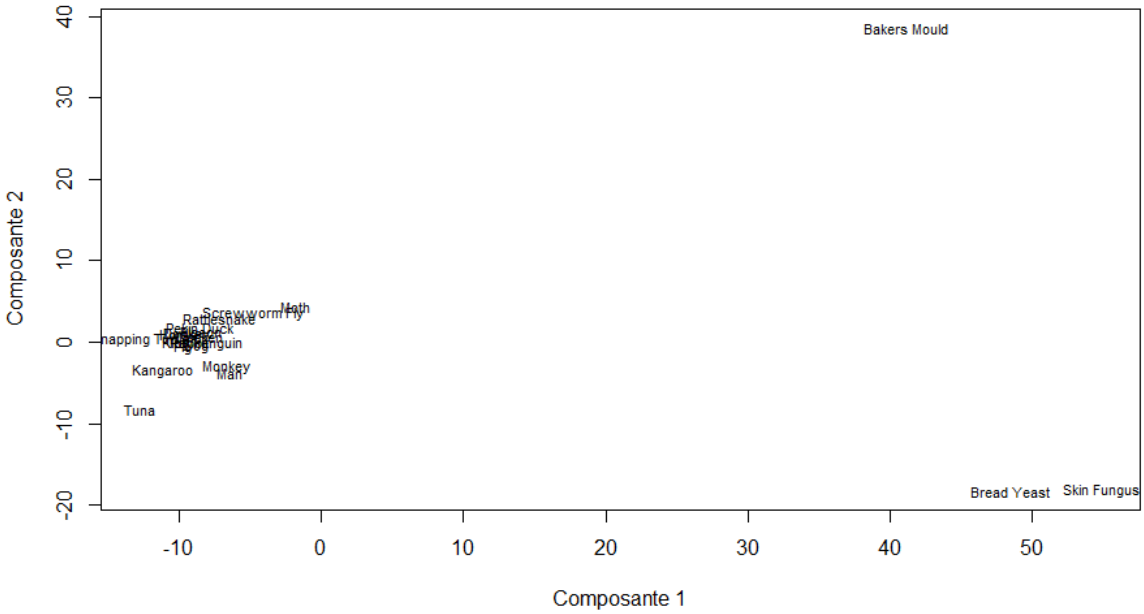

FIGURE 19 – Représentation sur l'axe 1-2 de la projection de Kruskal en $d = 5$ dimensions

Voici le diagramme de Shepard associé :

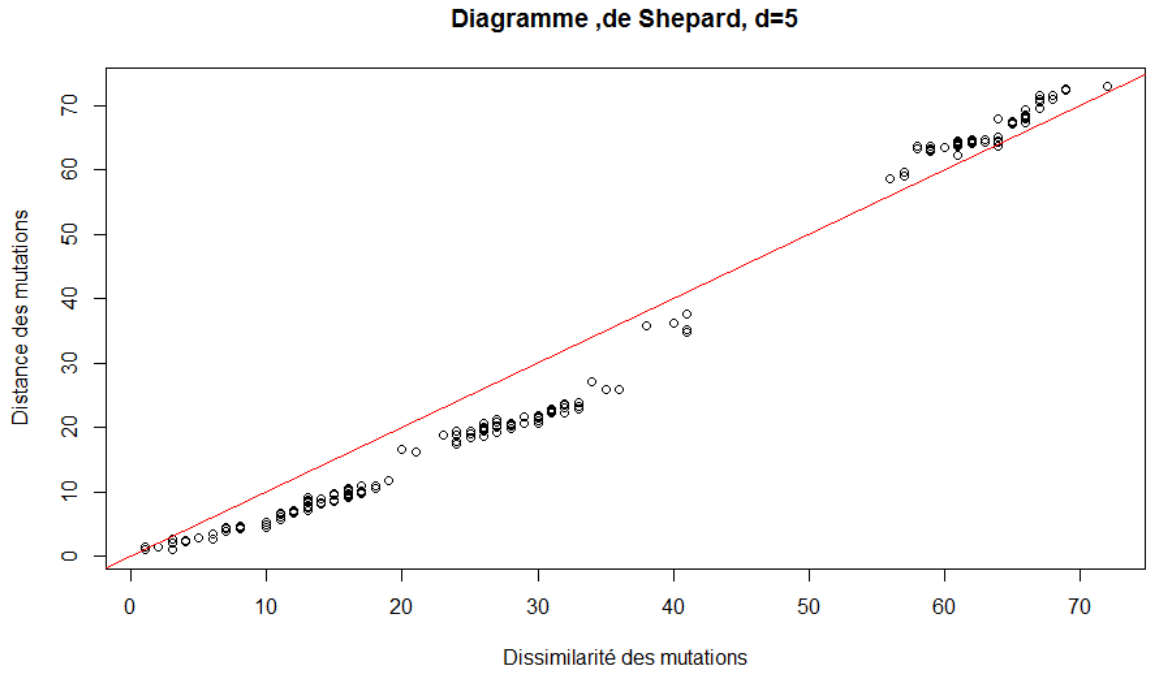

FIGURE 20 – Diagramme de Shepard avec 5 dimensions et la projection de Kruskal

Nous avons donc vu plusieurs méthodes de positionnement multidimensionnel, qui possèdent chacune leurs caractéristiques et des environnements ou types de présentation des données qui leurs correspondent le plus.

L'augmentation de la dimension dans laquelle on représente les données améliore systématiquement la qualité de la représentation, mais en choisir une trop grande compromet l'objectif initial qui est d'obtenir une réduction des données dans des espaces plus petits.

Les représentations obtenues par chacune des méthodes sont plus ou moins similaires ; elles ont en tout cas à chaque fois écarté les espèces *Bakers Mould*, *Bread Yeast* et *Skin Fugus* de toutes les autres, ce qui montre une forte dissimilarité par rapport au reste de l'échantillon, ce que confirme bien le tableau de distances fourni initialement.

3 Classification hiérarchique

La classification hiérarchique est une méthode dont l'objectif est de construire une hiérarchie indicée d'un ensemble possédant une mesure de dissimilarité de telle manière que les classes de plus petit indice contiennent les points les plus proches.

3.1 Données Mutations

Nous effectuons la classification hierarchique ascendante des données de mutations selon plusieurs critères d'agrégation (distance minimale, maximale, moyenne et critère de ward) avec la fonction *hclust*. Voici les dendogrammes de chaque classification associés :

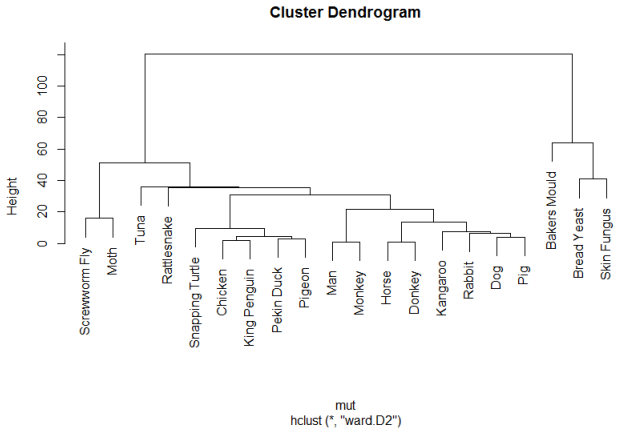

FIGURE 21 – Critère de Ward

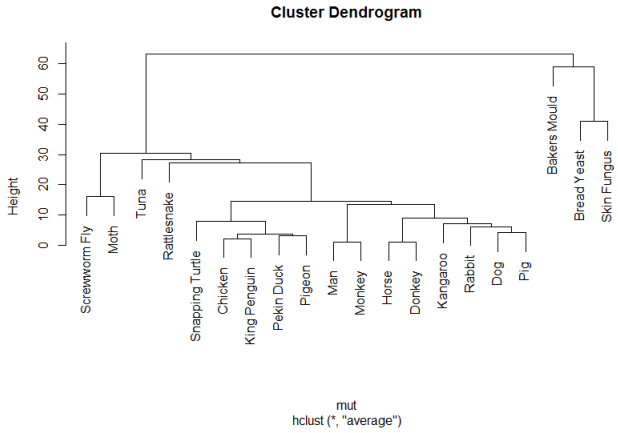

FIGURE 22 – Agrégation en distances moyennes

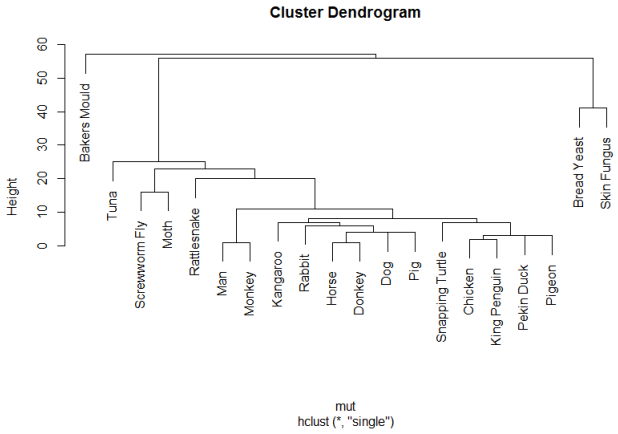

FIGURE 23 – Agrégation en distances minimales

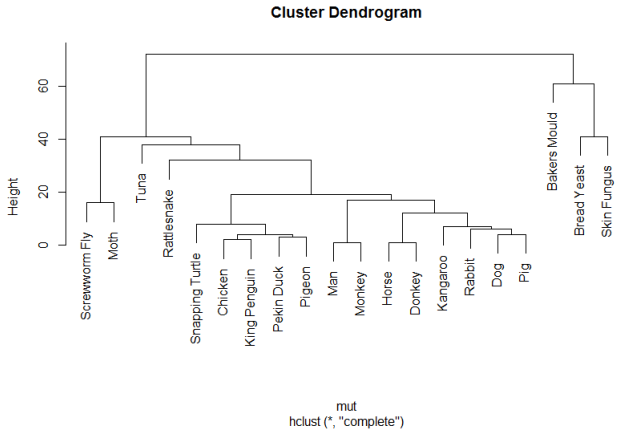

FIGURE 24 – Agrégation en distances maximales

On remarque tout d'abord qu'une agrégation par le critère de Ward (augmentation de l'inertie intra-classe minimale c'est-à-dire augmentation de l'inertie interclasse maximale) donne les mêmes *cluster* qu'une agrégation par distances moyennes, à des distances interclasses (dissimilarités) près.

En revanche, on obtient des résultats différents avec une agrégation en distance minimale ou maximale. Cependant, quoi qu'il arrive, ce que l'on remarque à chaque fois c'est la présence de deux (maximum trois pour les distances minimales) *cluster* principaux, le premier possédant toutes les espèces et l'(les) autre(s) les espèces *Bakers Mould, Bread Yeast, Skin Fungus*, ce qui caractérise un fort éloignement de leur part.

Cela concorde avec les représentations des individus que l'on a eues grâce à l'analyse factorielle d'un tableau de distances effectuée dans la partie 2.3 (figure 8 et suivantes) qui excluaient à chaque fois ces trois espèces du reste.

3.2 Données Iris

À présent, nous allons classifier hiérarchiquement les données Iris.

Classification ascendante

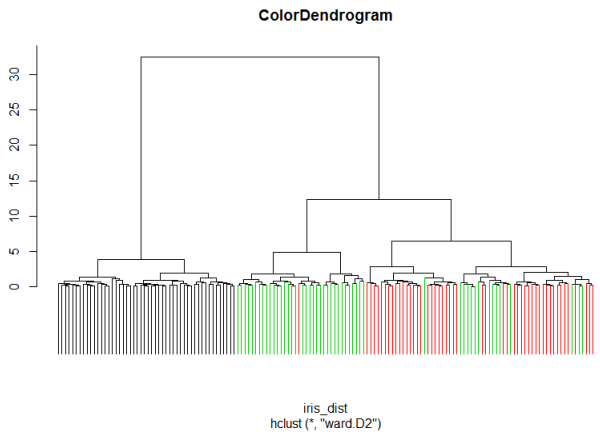

FIGURE 25 – Critère de Ward

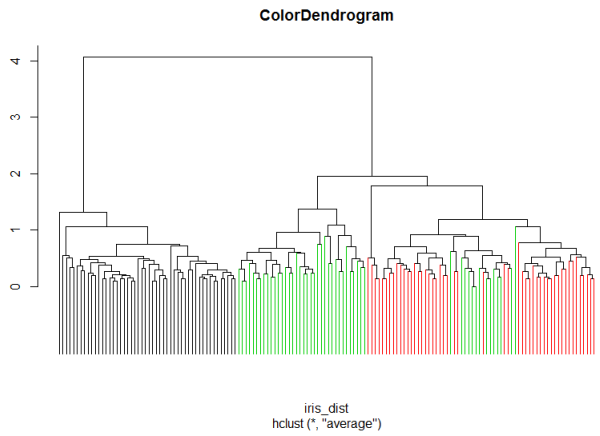

FIGURE 26 – Agrégation en distances moyennes

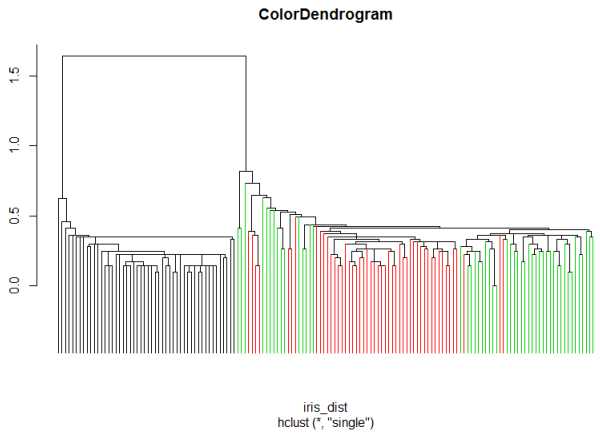

FIGURE 27 – Agrégation en distances minimales

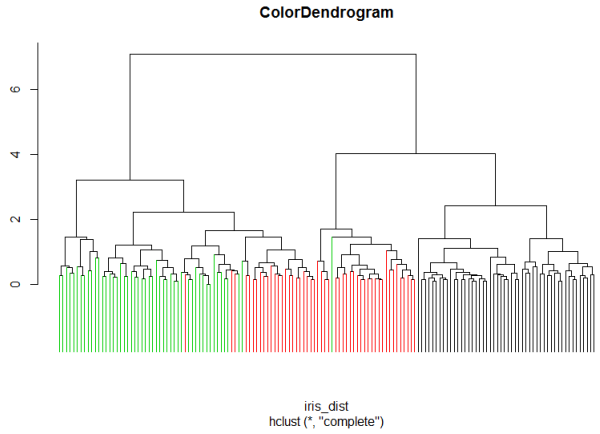

FIGURE 28 – Agrégation en distances maximales

On observe la présence de deux *cluster* principaux pour chaque agrégation. Le deuxième groupe est séparé en deux sous-groupes pour les agrégations *moyenne* et *Ward*. L'agrégation minimale produit un résultat plus éloigné des autres, et l'agrégation maximale divise plus la première classe.

Afin de vérifier la classification effectuée par chacun des critères, il est intéressant de couper l'arbre produit par le dendogramme avec la fonction *cutree*. On choisit de couper en deux ou trois groupes et on fait la correspondance avec le nom des espèces (à noter que les couleurs sur le dendogramme permettent déjà de distinguer les espèces présentes dans chaque cluster).

Groupes avec le critère de Ward :

clustercutward2	setosa	versicolor	virginica
1	50	0	0
2	0	50	50

FIGURE 29 – Coupe en 2 classes

clustercutward3	setosa	versicolor	virginica
1	50	0	0
2	0	49	15
3	0	1	35

FIGURE 30 – Coupe en 3 classes

Groupes avec le critère *moyenne* :

clustercutaverage2	setosa	versicolor	virginica
1	50	0	0
2	0	50	50

FIGURE 31 – Coupe en 2 classes

clustercutaverage3	setosa	versicolor	virginica
1	50	0	0
2	0	50	14
3	0	0	36

FIGURE 32 – Coupe en 3 classes

Groupes avec le critère *minimum* :

clustercutsingle2	setosa	versicolor	virginica
1	50	0	0
2	0	50	50

FIGURE 33 – Coupe en 2 classes

clustercutsingle3	setosa	versicolor	virginica
1	50	0	0
2	0	50	48
3	0	0	2

FIGURE 34 – Coupe en 3 classes

Groupes avec le critère *maximum* :

clustercutcomplete2	setosa	versicolor	virginica
1	50	27	1
2	0	23	49

FIGURE 35 – Coupe en 2 classes

clustercutcomplete3	setosa	versicolor	virginica
1	50	0	0
2	0	23	49
3	0	27	1

FIGURE 36 – Coupe en 3 classes

Les figures ci-dessus permettent de visualiser le contenu (espèces) de chaque classe définie par l'algorithme de classification. À gauche on retrouvera un groupement selon 2 classes, et à droite 3 classes. On voit alors facilement dans quelle classe est mise chaque espèce d'iris.

On observe premièrement (figures de gauche) que systématiquement la classification a regroupé les espèces *setosa* d'un côté, et les espèces *versicolor* et *virginica* de l'autre, excepté pour la *CAH* en critère maximal (ce qui est logique puisque l'on voit sur la figure 28 que le premier groupe est bien plus mixte que sur les autres classifications). Cette première observation concorde avec la première vue que l'on a eue sur les données avec une analyse en composantes principales (figures 1 et 2 qui distinguait deux groupes principaux : le premier composé de *setosa* et le deuxième de *versicolor* et *virginica*.

Deuxièmement (figures de droite), on voit que la séparation des espèces *versicolor* et *virginica* est à chaque fois plus complexe puisque lorsque l'on coupe l'arbre l'arbre en 3 classes (qui correspondraient donc théoriquement à chaque espèce) le contenu de chacune varie beaucoup. La première n'est composée que d'espèces *setosa*, alors que la deuxième et la troisième ne sont pas séparées selon les espèces *versicolor* et *virginica*. Cette deuxième observation concorde aussi avec la vue obtenue sur la figure 2, puisque les espèces en vert et bleu sont bien plus proches et donc moins différentiables. On a un groupe d'individus plus mixte et mélangé.

Au final, la classification ascendante hiérarchique nous permet (selon le critère d'agrégation, Ward étant la méthode la plus courante et souvent la meilleure) d'obtenir avec plus ou moins de précision les mêmes classes d'individus que l'on observe avec les méthodes de positionnement multidimensionnel.

Classification descendante

Nous utilisons cette fois-ci la classification hiérarchique descendante (CDH) des données. Avec R, on la fait grâce à la fonction *diana* de la bibliothèque *cluster*. Voici le résultat obtenu :

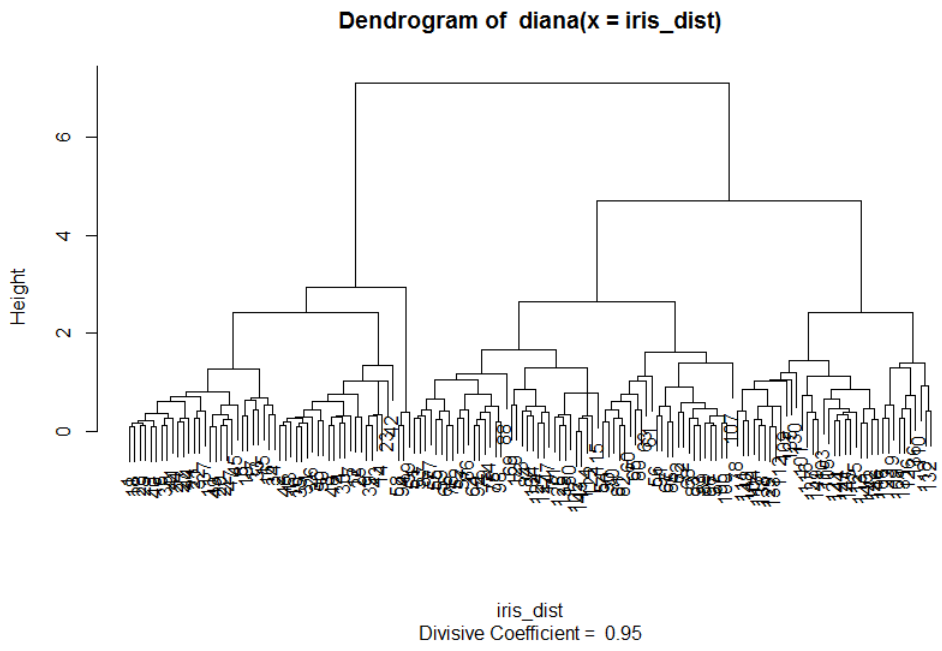

FIGURE 37 – Classification descendante hiérarchique des données Iris

Cette nouvelle classification laisse apparaître une quatrième classe (hauteur 3), qui n'est incluse ni dans *setosa* ni dans *virginica* ou *versicolor*. Cela correspondrait donc à trois individus qui forment une quatrième classe. Cette éventuelle quatrième classe n'est jamais apparue auparavant et ne se distingue jamais sur les positionnements en deux dimensions. Cette classification semble donc légèrement moins performante, du moins exacte, dans notre cas.

4 Méthode des centres mobiles

Dans cette partie nous tenterons d'effectuer des partitions avec la fonction *kmeans* : méthode des centres mobiles (cas particulier des nuées dynamiques).

Cette méthode découle d'un problème d'optimisation combinatoire dont le principe est de regrouper les points en k groupes que l'on appelle classe ou *cluster*, de manière à minimiser une certaine fonction ou un critère. En général le critère utilisé est l'inertie intra-classe.

4.1 Données Iris

Avant de faire le partitionnement, nous appliquons une ACP pour obtenir les informations les plus importantes à partir des données. Cela nous permet de représenter les individus dans 1 plan et d'éviter une représentation avec les 4 variables (6 plans) qui n'est pas compréhensible. Voici les résultats alors obtenus :

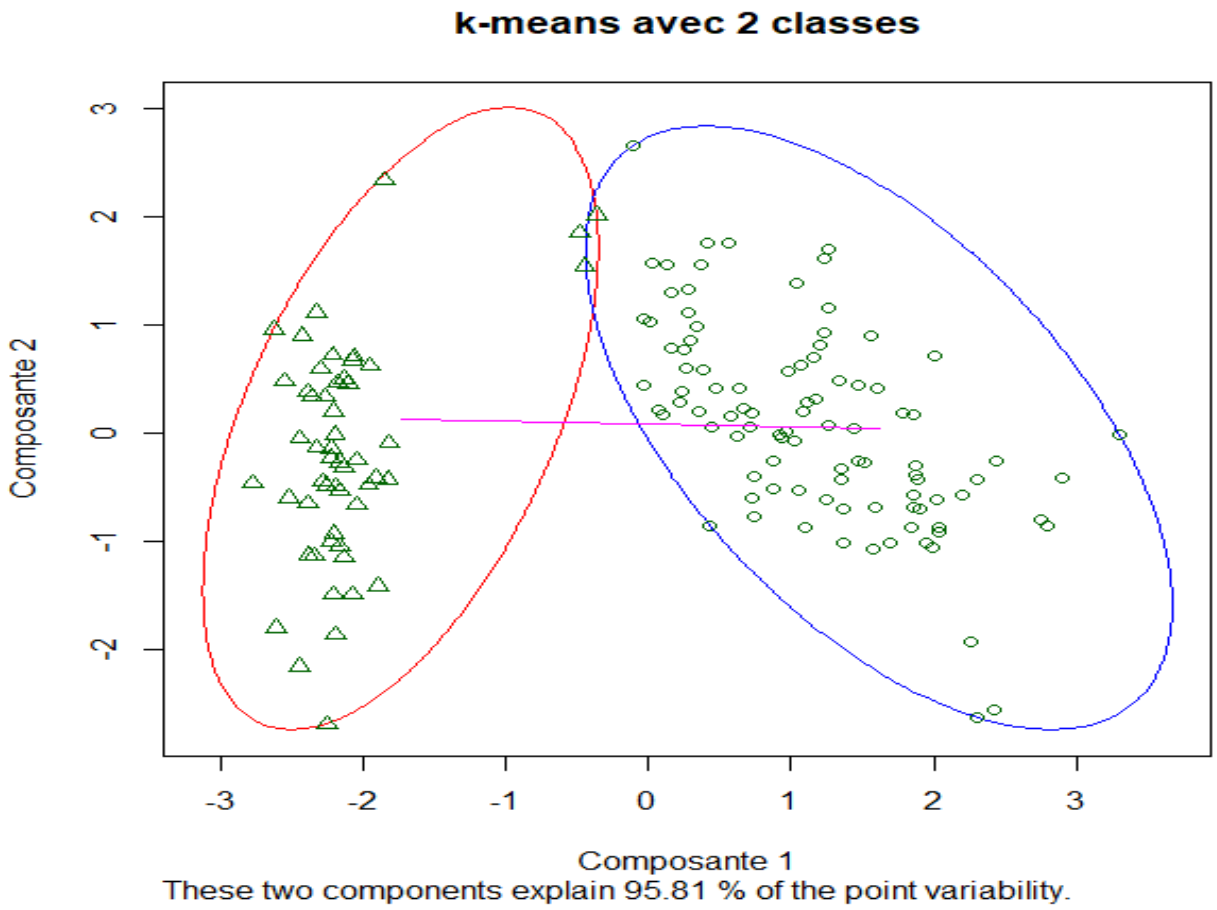

FIGURE 38 – k-means avec 2 classes

	setosa	versicolor	virginica
1	0	47	50
2	50	3	0

FIGURE 39 – Table de résultat de classification pour k=2

Nous savons avant que le jeu de données est composé de 3 espèces différentes. Lors d'une partition en 2 groupes, le résultat est assez simple à interpréter : *versicolor* et *virginica* sont dans un même cercle ; en revanche, il y a bien une séparation entre *setosa* et les autres.

On observe également que trois individus *versicolor* sont présents dans la classe n°2, ce qui est de nature inattendue lorsque l'on attend une classification qui suit les groupes d'espèces, ce sont donc les *erreurs* de la classification. Ces trois individus sont les trois triangles que l'on voit sur le chevauchement des deux ellipses.

Regardons à présent le partitionnement en 3 classes :

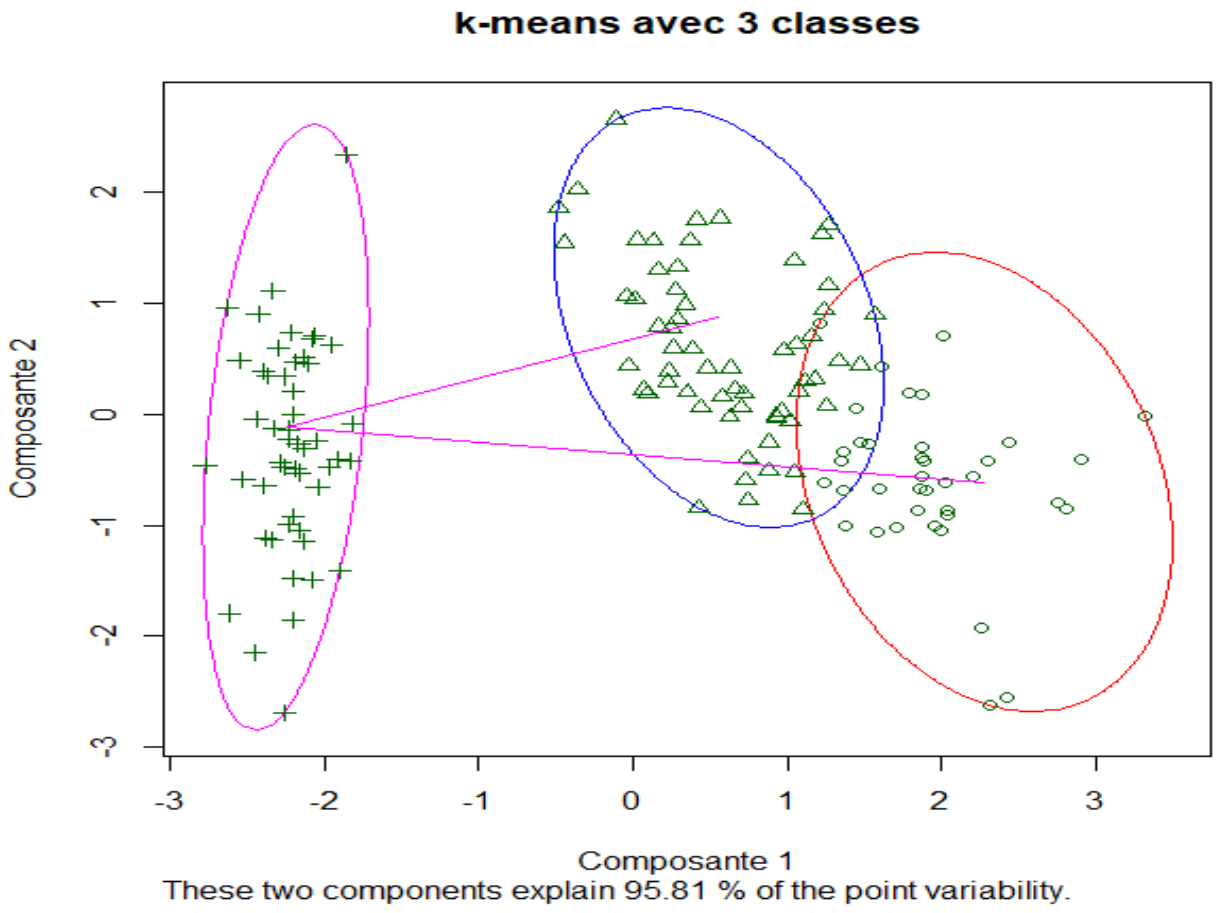

FIGURE 40 – k-means avec 3 classes

	setosa	versicolor	virginica
1	0	2	36
2	0	48	14
3	50	0	0

FIGURE 41 – Table de résultat de classification pour k=3

Comme nous l'avons prévu, avec 3 classes *setosa* est bien à nouveau séparée des autres mais on obtient cette fois-ci une nouvelle partition : *virginica* et *versicolor* sont elles aussi séparées. En revanche la séparation est bien moins claire, il y a un chevauchement entre le premier (bleu) et le deuxième groupe (rouge). De plus la table de classification montre qu'il y a un mélange d'espèces dans les deux premiers groupes (comme on l'a vu dans la CAH), le partitionnement obtenu n'est donc pas 100% adéquat avec le partitionnement par espèce.

Bien que cela nous permette d'obtenir une rapide vue de la classification effectuée par l'algorithme, on ne voit les erreurs que dans les tableaux et pas graphiquement car les symboles (triangle, cercle, croix) n'indiquent que la classe de l'individu suite à la méthode, pas son espèce réelle.

Pour remédier à cela, voici un graphique qui représente via des points à double composantes le *clustering* effectué par l'algorithme : la partie entourant le point (couleurs claires) donne la vraie espèce d'iris, et la partie intérieure (couleurs foncées) donne l'espèce de l'individu selon la classification effectuée par l'algorithme.

Quand les couleurs ne concordent pas on obtient donc les erreurs de l'algorithme : par exemple un point noir dans un cercle rouge signifie que la classification a considéré que l'individu appartenait à la classe 1 (théoriquement les *versicolor*) alors qu'en réalité c'était une espèce *virginica*

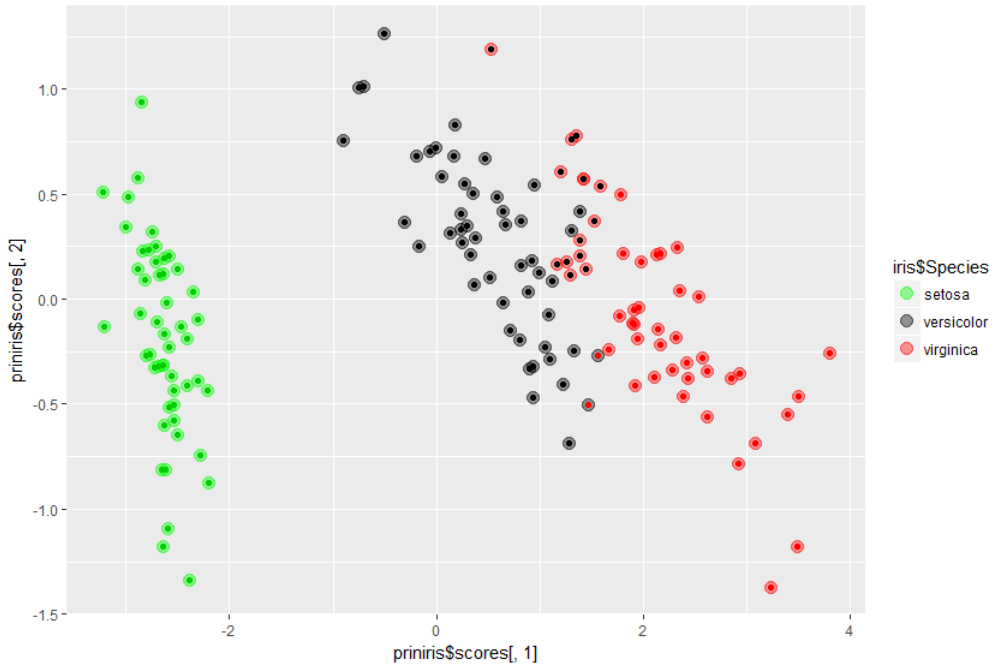

FIGURE 42 – Graphique de la classification en k=3 au regard des espèces d'Iris

Passons maintenant à une classification en 4 classes :

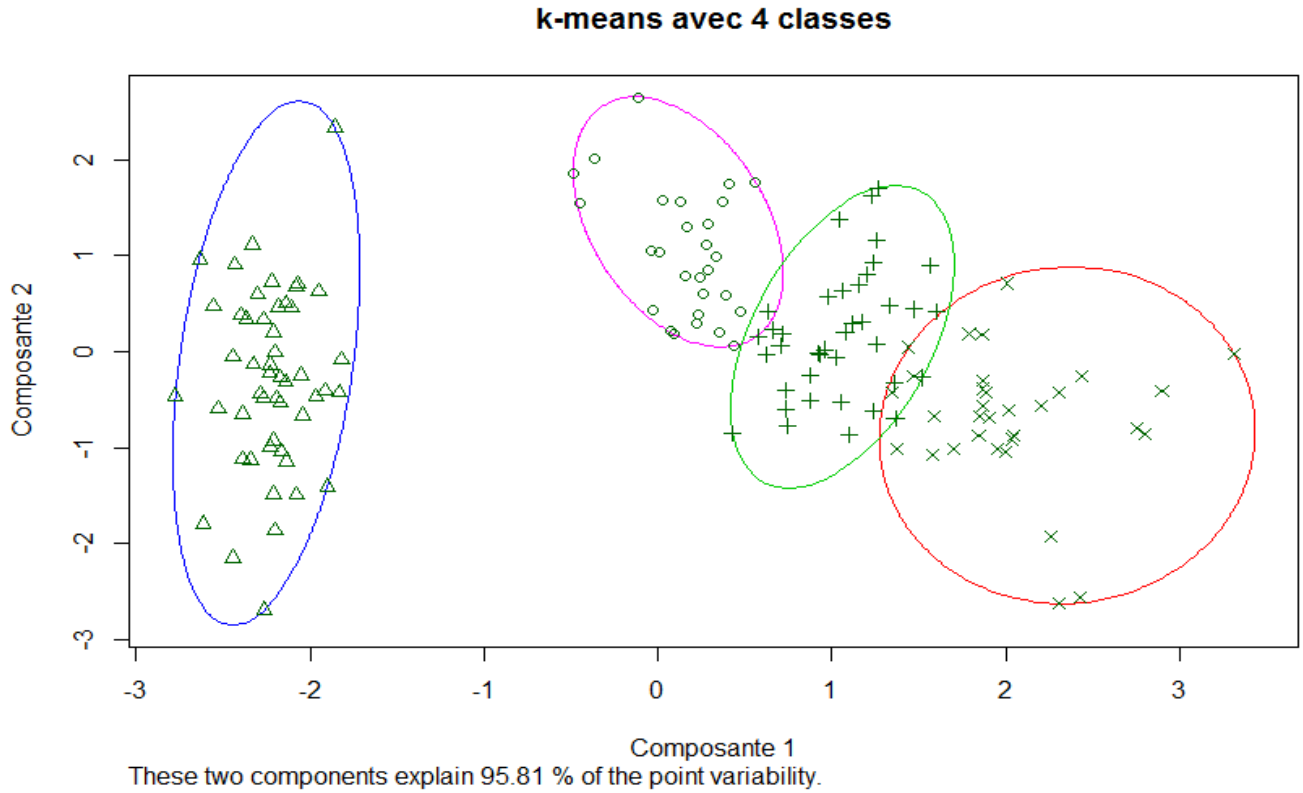

FIGURE 43 – k-means avec 4 classes

	setosa	versicolor	virginica
1	0	27	1
2	50	0	0
3	0	23	17
4	0	0	32

FIGURE 44 – Table de résultat de classification pour k=4

Maintenant, il y a plus de classes que d'espèces, *versicolor* semble alors séparée en 2 sous-classes, comme le confirme le graphique suivant :

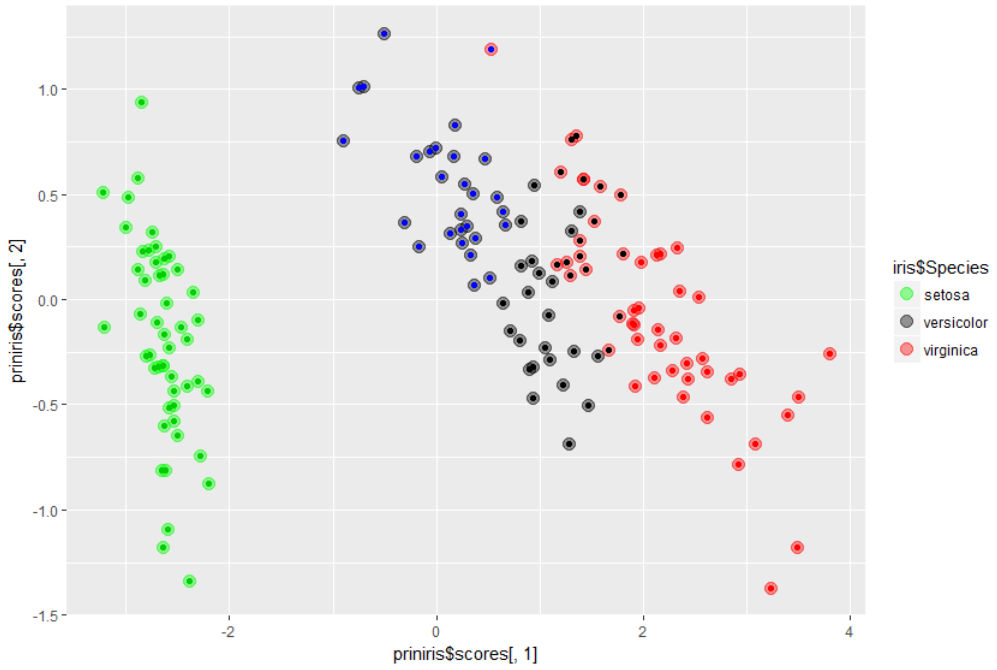

FIGURE 45 – Graphique de la classification en k=4 au regard des espèces d'Iris

Lors de partitions en 4 classes, on remarque deux phénomènes :
— L'espèce *versicolor* est séparée en deux classes
— On obtient un *cluster transverse* : le n°3, qui possède environ 50% d'individus de chaque espèce. On y trouve alors des espèces *virginica* et *versicolor* (points de couleur **noire** dans la zone du milieu qui se confond dans le nuage de points), et dans les *cluster* 1 et 4 on a des espèces bien séparées qui forment les extrémités gauche et droite du nuage de points comprenant *versicolor* et *virginica*.

La stabilité du résultat de la partition

Lors de partitions avec la méthode des *kmeans* (centres mobiles), les résultats ne sont pas toujours les mêmes parce que le choix des noyaux initiaux est aléatoire.

Pour éviter de tomber dans un résultat aléatoire qui serait non-optimal, nous utilisons l'option $nstart = 25$ pour utiliser l'algorithme dans R, qui indique que la fonction va tenter plusieurs configurations initiales (ici 25) et garder celle qui donne le meilleur *clustering*. On remarque que la classification alors obtenue est celle qui revient le plus souvent lorsque l'on répète de nombreuses fois aléatoirement l'algorithme.

Nombre de classes optimal

La méthode des centres mobiles nécessite de connaître le nombre de centres initiaux et donc de classes finales. Lorsque l'on n'a pas d'informations préliminaires sur le nombre de partitions réelles formées par la distribution des individus observés et donc le nombre théorique optimal de classes qui devraient apparaître, il est intéressant d'utiliser la méthode du coude (*elbow method*) qui montre à partir de combien de classes le critère (inertie des

classes par rapport à leur centres respectifs) n'est pas significativement amélioré (il existe une variante similaire avec le pourcentage de variance expliquée selon le nombre de classes).

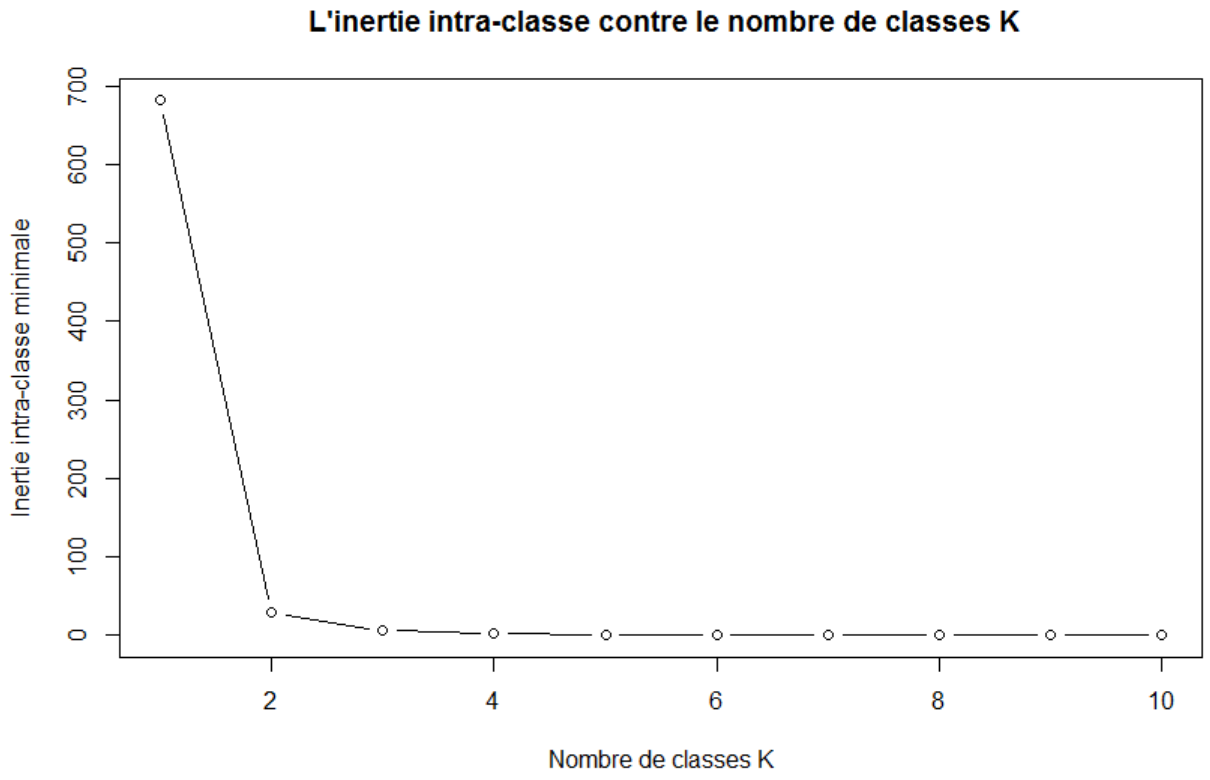

FIGURE 46 – Inertie intra-classe minimale pour 100 répétitions de l'algorithme des kmeans de k = 1 classe jusqu'à k = 10 classes

On observe donc un *coude*, une cassure dans le graphique autout de $k = 2$ et $k = 3$ On peut donc penser que le nombre de classes optimales pour la méthode des centres mobiles sur Iris est 2 ou 3. Le résultat que nous avons obtenu avec 3 classes est le plus proche du jeu de données initial, comme nous le voyons dans la suite.

Comparaison des résultats avec la partition réelle en trois groupes

Nous avons effectué trois analyses de classification non hiérarchique avec les centres mobiles. Il convient de conclure sur la proximité entre les classes obtenues par l'algorithme de partitionnement et celles qui sont *réelles*, c'est-à-dire définies par l'espèce d'Iris : *setosa*, *versicolor* et *virginica*. Pour cela, on s'appuie sur les visualisations graphiques ainsi que sur un indice numérique : l'indice ajusté de Rand, qui mesure le taux d'accord entre deux partitions d'un ensemble en calculant le nombre de paires de points classés identiquement dans les deux partitions.

La première classification, en deux groupes, laisse apparaître deux classes qui contiennent respectivement l'espèce *setosa* et les espèces *versicolor/virginica*.

$$AdjustedRandIndex = 0.54 \tag{1}$$

Le partitionnement est considéré comme *à moitié d'accord* avec la réalité. Sans considération numérique, cette classification n'est pas si mauvaise car comme l'attestent les représentations en deux dimensions des individus, les espèces *versicolor* et *virginica* sont très proches. Un partitionnement en deux groupes au lieu de trois regroupe donc logiquement ces deux espèces d'un côté et *setosa* à part (uniquement trois erreurs dans la table de contingence de la figure 39).

La deuxième classification, en trois groupes, donne trois classes qui contiennent à quelques erreurs près les trois espèces d'Iris. On a l'indice suivant :

$$AdjustedRandIndex = 0.73 \tag{2}$$

Les partitions par les *kmeans* et des espèces réelles sont donc similaires à quasi 75%, ce qu n'est pas mauvais.

La troisième classification, en quatre groupes, donne quatres classes qui séparent à présent les espèces *versicolor* en deux, mais les erreurs sont toujours présentes dans la zone du milieu. L'indice se verra donc diminué par rapport à un partitionnement en $k = 3$ puisqu'il n'y a pas quatre espèces d'Iris :

$$AdjustedRandIndex = 0.65 \tag{3}$$

On se situe donc entre un partitionnement en $k = 2$ et $k = 3$.

Au final, chacune des classifications apporte son lot d'informations utiles à l'analyse, mais celle qui se rapproche le plus de la réalité est logiquement celle en 3 classes puisque l'on a dans la réalité trois espèces que l'on peut distinguer dans un positionnement en deux dimensions.

4.2 Données Crabs

Nous allons effectuer un partitionnement des données en 2 puis en 4 classes. Les résultats n'étant pas tout à fait stables à chaque fois, nous utilisons de nouveau l'option $nstart = 25$ afin d'obtenir le résultat optimal.

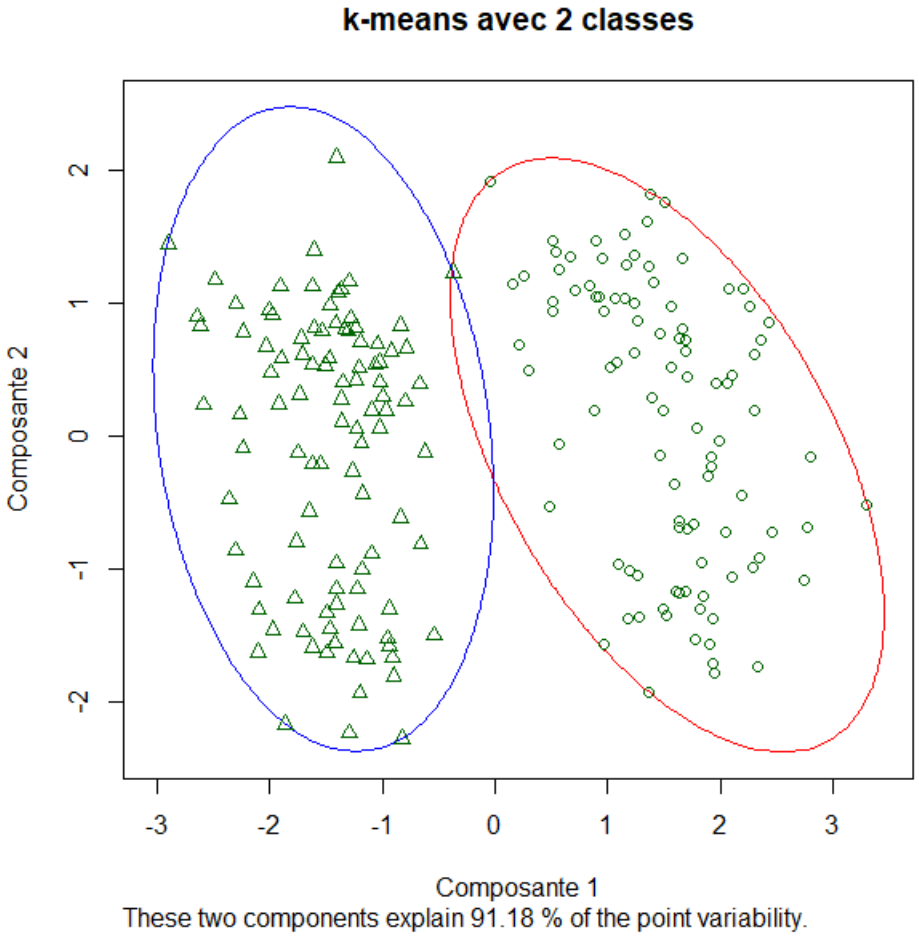

FIGURE 47 – K-means de Crabs avec k=2

	B	O
1	0	99
2	100	1

FIGURE 48 – Tableau de K-means de Crabs avec k=2

On obtient un partitionnement en 2 classes qui représentent les espèces Blue et Orange. Il n'y a qu'un individu, celui du milieu, qui n'est pas bien classé, sa position étant très intermédiaire.

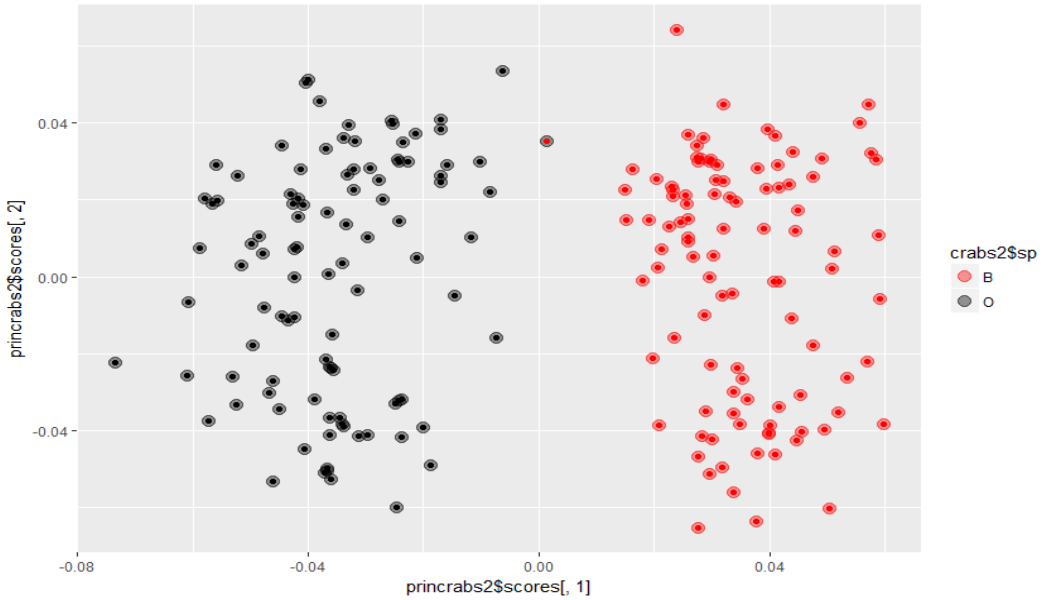

FIGURE 49 – Résultat du partitionnement selon les espèces

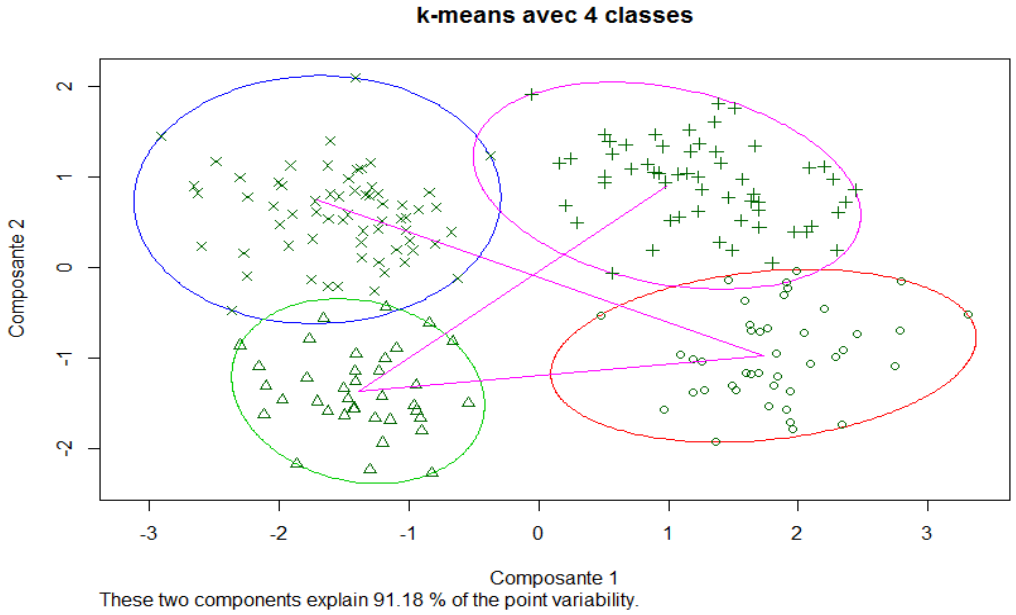

FIGURE 50 – K-means de Crabs avec k=4

Voici un graphique qui résume très bien le partitionnement effectué en 4, avec les individus plus homogènes qui rendent la classification parfois inexacte :

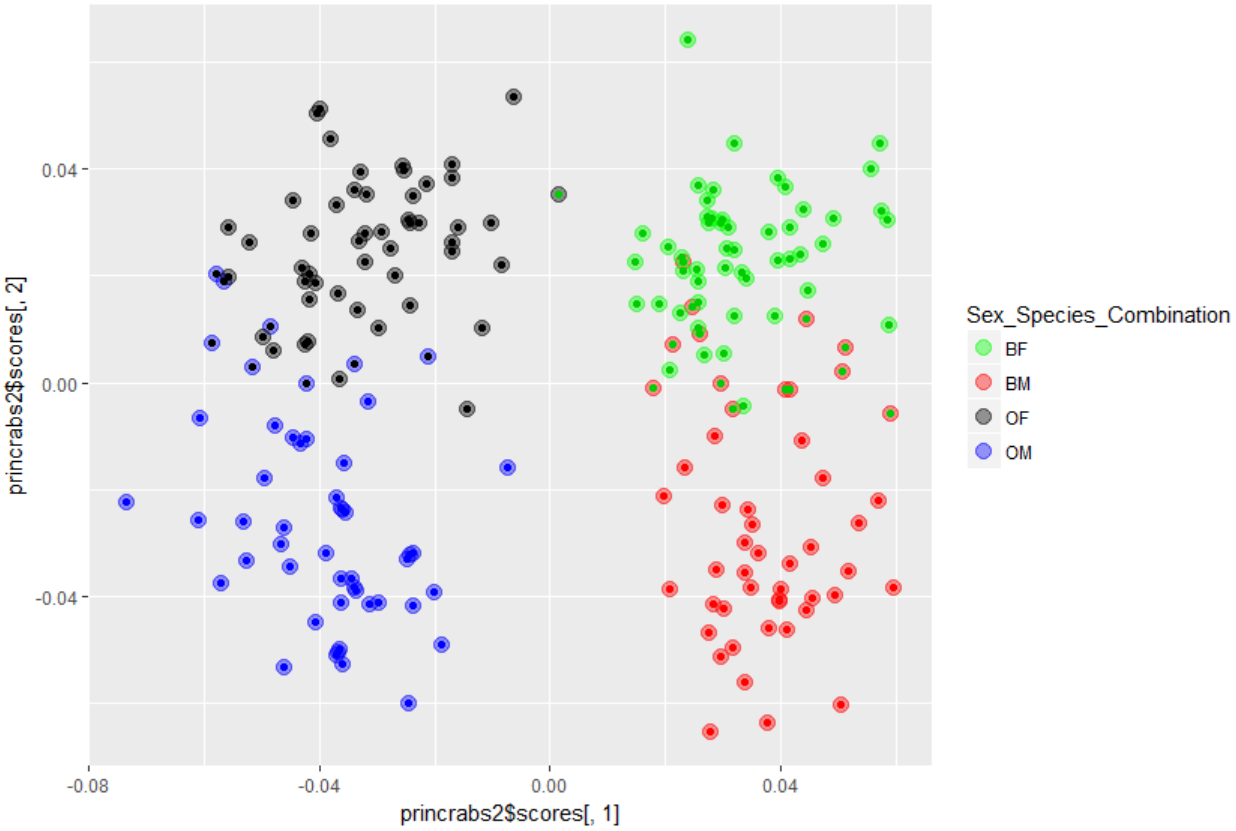

FIGURE 51 – K-means de Crabs avec k=4

On reconnaît l'individu mal classé entre les espèces Blue et Orange, puis on aperçoit à présent les individus considérés comme femelles au lieu de mâle et inversement. Ces observations sont à prendre en compte si l'on considère que l'algorithme effectue un partitionnement selon les quatre classes qui correspondent aux combinaisons de caractéristiques obtenables chez les crabes (voir légende du graphique) :

— Blue - Male
— Blue - Female
— Orange - Male
— Orange - Female

Il est très important de remarquer qu'une classification optimale (*nstart*, classification qui revient le plus) en $k = 2$ ne classera pas les individus selon leur sexe mais bien selon leur espèce, car la séparation entre les deux est bien plus importante (voir *Visualisation des données*). C'est donc la première classification *naturelle* que l'on obtient avec les kmeans.

Il faut obligatoirement passer à une classification en $k = 4$ pour voir la séparation entre les sexes, qui est bien moins évidente à observer au premier coup d'œil. Cette classification permet de bien décomposer les données *Crabs* dans les 4 groupes naturels.

4.3 Données Mutations

Voici un exemple de partitionnement des données de mutations en 3 classes :

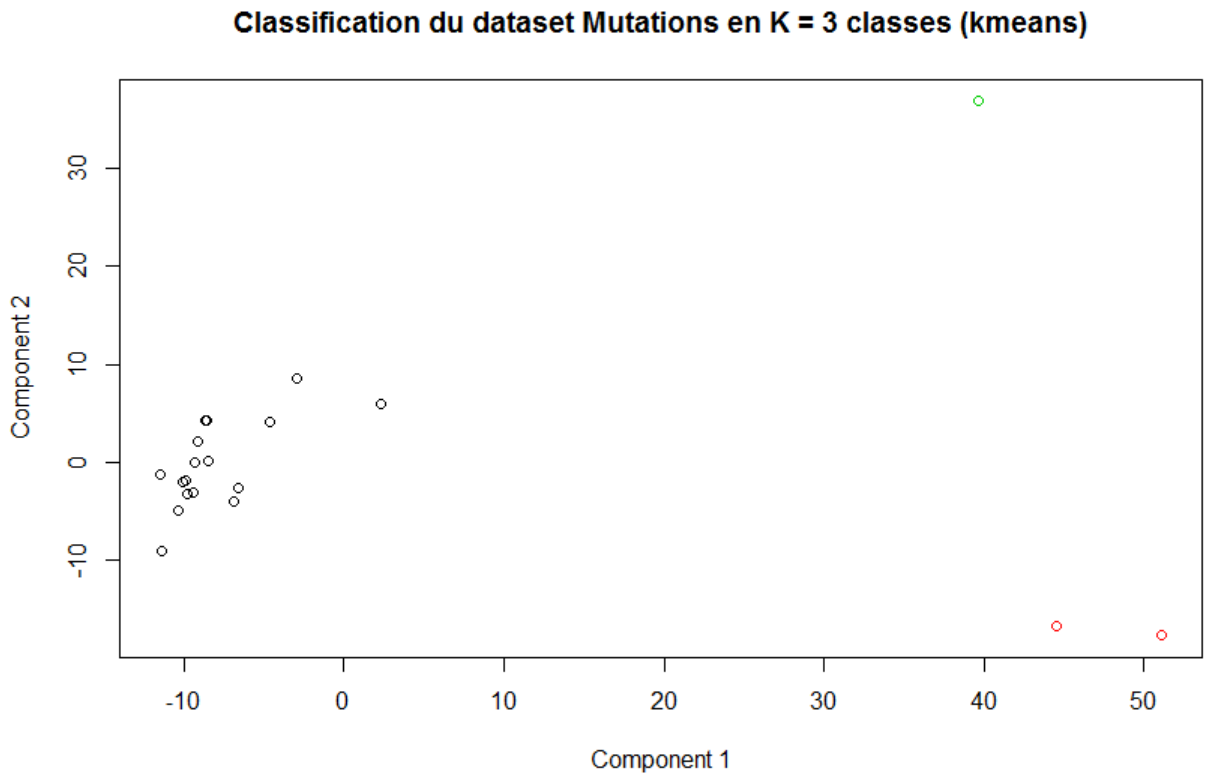

FIGURE 52 – Partition en 3 classes des données de mutations avec les kmeans, représentation dans le premier plan factoriel de l'AFTD

Sur cet exemple la classification est bonne. On obtient une séparation identique à chaque fois avec l'option *nstart*. Le nuage de gauche (points noirs) appartient à une classe, les deux points en bas à droite (*Bread Yeast* et *Skin Fungus* en rouge) à une autre et finalement *Bakers Mould* (en vert) forme une troisième classe.

En revanche sans cette option, les résultats obtenus ne sont pas du tout stables et varient constamment, la division du reste du nuage de points (à gauche) étant très aléatoire. Ce manque de stabilité montre la difficulté de l'algorithme à séparer les autres espèces car elles ne sont pas assez distinctes dans le premier plan factoriel de notre représentation (et aussi dans les données de dissimilarités initiales, qui sont assez linéaires et faibles).

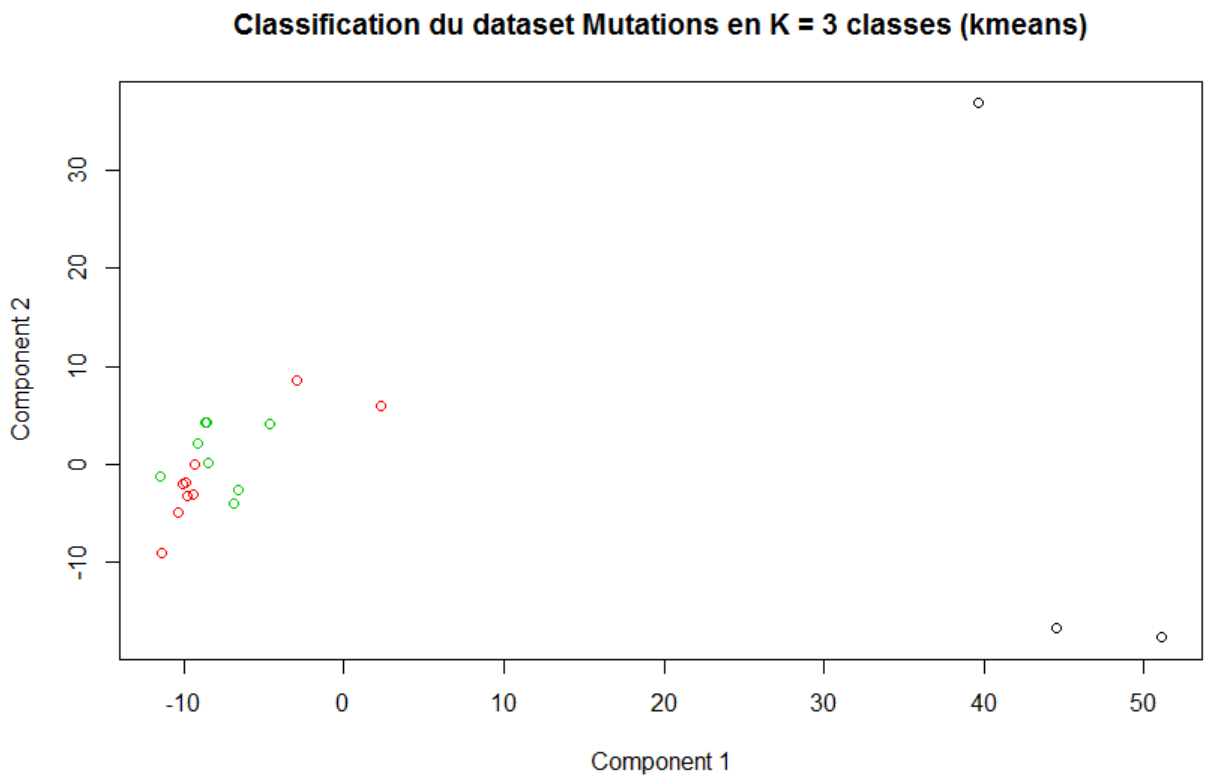

FIGURE 53 – Partition en 3 classes des données de mutations avec les kmeans, représentation dans le premier plan factoriel de l'AFTD

Sur cette représentation par exemple les trois espèces excentrées appartiennent au même *cluster* (en noir) et le reste est divisé en deux (rouge et vert).

Finalement, on remarque malgré tout que les trois espèces excentrées *Skin Fungus*, *Bread Yeast* et *Bakers Mould* n'ont jamais la même couleur que les autres espèces, quelque soit l'exécution de l'algorithme. C'est très important car cela confirme à nouveau leur caractère particulier dans l'échantillon.

5 Conclusion

Nous avons dans ce nouveau travail expérimenté d'autres méthodes de représentation multidimensionnelle de données.

Celles-ci nous ont permis de partir de sources différentes (comme des tableaux de distances et de dissimilarités) afin de voir qu'il est tout à fait possible de condenser l'information pour obtenir rapidement et efficacement des représentations dans des espaces de faible dimension et donc interprétables par l'œil humain.

Les méthodes de classification fournissent une autre approche de ces méthodes et donnent d'autres représentations comme les dendogrammes qui sont utiles pour catégoriser les individus d'un jeu de données.

Table des matières

Table des figures