Laboratoire CRIStAL

Machine Learning pour la détermination d'espèce à partir de spectre MALDI-FT issus de prélèvement archéologiques

Tuteur professionnel : Dr Julie Jacques

Tuteur universitaire : Dr Nadarajen Veerapen

CRIStAL
Centre de Recherche en Informatique, Signal et Automatique de Lille

Equipe Orkad

Présenté par : Emmanuel LAURENT

Université de Lille

Sommaire

Index des illustrations

I. Remerciements

Avant tout propos sur cette expérience, Il me paraît important de commencer ce rapport par des remerciements, à ceux qui m'ont accueilli et encadré durant ce stage.

Je tiens d'abord à remercier le docteur Julie Jacques, Enseignant-Chercheur en informatique à l'Ecole du Numérique (Université Catholique de Lille), mon maître de stage, pour m'avoir accueilli comme stagiaire au sein de l'équipe ORKAD du laboratoire CRIStAL, ainsi que pour ses conseils avisés et la confiance qu'elle m'a accordé tout au long de mon stage.

Merci également au docteur Nadarajen Veerapen, mon tuteur universitaire, chercheur au sein de l'équipe ORKAD, pour son temps et ses conseils apportés au quotidien de mon stage.

Je tiens ensuite à remercier le Professeur Clarisse Dhaenens, Doyenne du laboratoire de recherche CRiStAL, ainsi que le Professeur Laetitia Jourdan, chef d'équipe de l'équipe ORKAD pour l'organisation des réunions et de divers évènements qui m'ont permis de mieux appréhender le monde de la recherche.

Je tiens à remercier spécialement Weerapan SAE-DAN de m'avoir convié à sa soutenance de thèse, un évènement que j'ai trouvé fort intéressant.

Enfin, je remercie Guillaume, stagiaire dans l'équipe, Yousra ingénieure d'étude, Clément, Thomas et Agathe doctorants au sein de l'équipe et l'ensemble de l'équipe pour leurs accueils et les nombreux échanges enrichissant autour de leurs travaux.

Toutes ces personnes ont contribué, par leur disponibilité et leur bonne humeur, à rendre mon stage enrichissant et motivant.

II. Introduction

Présentation générale

Dans le cadre de ma formation, Master 1 système d'information et aide à la décision, j'ai eu à réaliser un stage du 17 mai au 31 août 2022 au sein de l'équipe ORKAD du laboratoire CRIStAL (centre de recherche en informatique signal et automatique de Lille).
Plus précisément j'ai eu à réaliser du Machine Learning sur des spectres MALDI-FT issus de prélèvements archéologiques. Souhaitant m'orienter à l'avenir vers le métier data scientist, curieux de découvrir le milieu de la recherche, j'ai été séduit à l'idée de réaliser ce stage. Celui-ci allait me permettre de travailler sur les outils nécessaires afin de réaliser du Machine Learning notamment les différentes librairies python.

Présentation du laboratoire

CRIStAL est né le 1er janvier 2015 sous la tutelle du CNRS, de l'Université Lille 1 et de l'Ecole Centrale de Lille en partenariat avec l'Université Lille 3, Inria et l'Institut Mines Telecom. L'unité est composée de près de 430 membres (222 permanents et plus de 200 non permanents). Les activités de recherche de CRIStAL concernent les thématiques liées aux grands enjeux scientifiques et sociétaux du moment tels que : BigData, logiciel, image et ses usages, interactions homme-machine, robotique, bio-informatique… avec des applications notamment dans les secteurs de l'industrie du commerce, des technologies pour la santé, des smart grids. Aujourd'hui le laboratoire CRIStAL c'est près de 5M€ de budget et en moyenne 446 publications par an !

L'équipe ORKAD

Créée en 2017, l'équipe ORKAD, dirigée par le Professeur Laetitia Jourdan est composée de 10 membres (7 permanents et 6 non permanents). Son objectif, est d'exploiter simultanément l'expertise en optimisation combinatoire et en extraction de connaissances pour résoudre des problèmes d'optimisation. C'est donc au sein de cette équipe que j'ai effectué ce stage.

L'équipe BONSAI

L'objectif principal de l'équipe BONSAI sous la responsabilité du Professeur Hélène Touzet est de développer des algorithmes et des modèles pour l'analyse des séquences biologiques. Cette équipe travaille notamment en parallèle avec l'équipe ORKAD sur la même problématique que celle associée à mon stage mais avec une approche différente.

Le laboratoire MSAP

Fondée en janvier 2010, le laboratoire MSAP (Miniaturisation pour la Synthèse, l'Analyse et la Protéomique) dirigé par Ahmed Mazzah, est une Unité d'Appui et de Recherche structurée autour de 3 métiers : spectrométrie de masse, techniques séparatives, synthèse et physico-chimie organique. Les données issues de la spectrométrie sur lesquelles la thématique de mon stage est basée proviennent de ce laboratoire par l'intermédiaire de Fabrice Bray.

Objectif et problématique du stage

Comme évoqué précédemment, l'objectif du stage est de réaliser du Machine Learning sur des résultats de spectres MALDI-FT issus de prélèvements archéologiques. Les paléontologues à l'origine du projet ont besoin de pouvoir déterminer, à partir d'un prélèvement osseux, une espèce. Pour le moment ces travaux sont souvent réalisés à partir de constatations structurelles qu'ils font sur les ossements (ex : la longueur du fémur, la structure de certaines articulations, ...) mais ce n'est pas toujours possible lorsque l'on se retrouve avec des fragments, ou encore un objet taillé dans l'os. Certaines espèces proches sont parfois complexes à la différentiation par exemple l'auroch (un ancêtre du bœuf) et le bœuf. Le Machine Learning pourrait s'avérer utile dans le processus de création d'un outil automatisé et fiable dans cette tâche.
Pour rappel le machine Learning désigne le processus par lequel les ordinateurs développent la reconnaissance de schémas ou l'aptitude à apprendre continuellement et à faire des prévisions à partir de données.

Les spectres MALDI-FT sont produits à l'aide d'un spectromètre de masse. La spectrométrie de masse est une technique d'analyse qui permet la détermination des masses moléculaires des composés biologiques analysés, permettant ainsi leur identification et leur quantification. La spectrométrie de masse est particulièrement utilisée en biologie pour l'analyse de molécules comme les protéines

Figure 1: Spectromètre de masse du laboratoire MSAP

Dans notre étude, des tissus osseux issus de prélèvements archéologiques provenant de différentes espèces animales, sont passés au spectromètre de masse et nous permettent d'observer leurs « composition moléculaire ». Dans cette étude préliminaire de faisabilité, l'objectif est de déterminer s'il est possible d'identifier une espèce animale à l'aide des données de spectrométrie et des méthodes de classification en Machine Learning. Une telle possibilité serait d'intérêt pour en informer les paléontologues, puisqu'elle permettrait d'aboutir à la création d'un outil facile d'identification d'espèces animales.

III. Phase exploratoire

A) Découvertes des données et des travaux précédents

Les Données

Les données analysées sont issues de spectres MALDI-FT (cf. fig2). Sur un spectre de masse, chaque pic correspond à la masse moléculaire d'une molécule présente dans l'échantillon analysé. Ainsi, chaque échantillon archéologique analysé permet d'identifier un grand nombre de molécules (300 ou plus). Pour commencer, le spectromètre de masse nous permet d'obtenir un spectre de masse représentant les rapports m/z, où m représente la masse et z la valence des ions détectés ainsi que leur Intensité I.

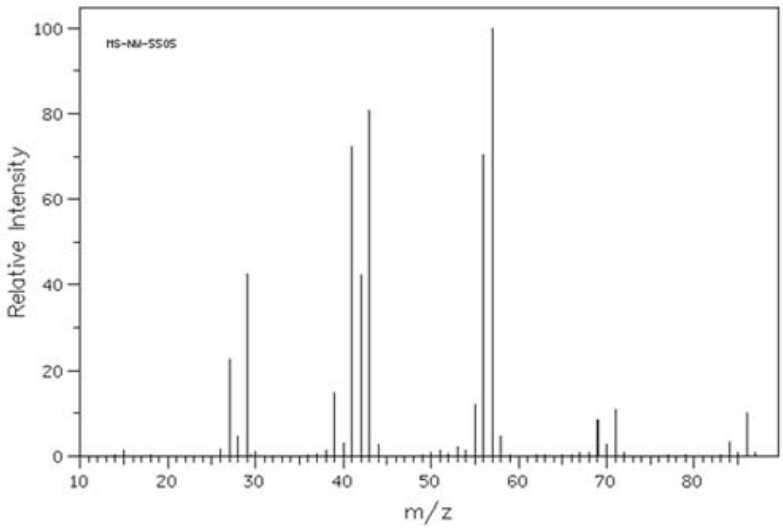


Figure 2: Graphique spectre de masse

Une fois ce spectre obtenu, ces données sont alors traduites dans un fichier csv qui vont alors nous permettre de travailler plus facilement, ce sont les données qui nous sont transmises par le laboratoire MSAP (cf. fig3).

	A	B
1	m/z	I
2	836.42228	537294
3	852.41646	309358
4	898.49309	546337
5	1095.53933	1406327
6	1105.54829	4292108

Figure 3 : Echantillon d'un spectre de masse traduit en un fichier csv

A mon arrivée, 2 datasets contenant alors plusieurs de ces fichiers avaient déjà été transmis :

- Le premier contient plusieurs spectres concernant 2 espèces distinctes : le bœuf et l'Auroch (espèce disparue de bœuf sauvage apparue en Inde il y a 1,8 million d'années, c'est l'ancêtre de notre bœuf domestique). Chaque échantillon a été digéré 6 fois et analysé 2 fois ce qui donne 12 fichiers. L'objectif premier de ce datasets était donc de déterminer les possibles différences de ces 2 espèces.
- Le second contient pléthore d'espèces, 29 exactement allant des mammifères aux poissons en passant par les oiseaux pour un total de 196 fichiers.

Les travaux précédents

Avant mon arrivée, un travail préliminaire avait déjà été réalisé à l'aide de ces données. De ce travail on trouve des analyses réalisées à l'aide de méthodes d'apprentissage non supervisé (voir le paragraphe dédié). 2 outils piliers de mon stage ont été utilisé : Scikit Learn une librairie python ainsi que Metaboanalyst un outil spécialisé dans l'analyse spectrométrique. De ce travail, il reste une analyse comparative de ces outils ainsi que des bases de données constituée de l'ensemble des spectres présentés précédemment des différents spectres présentés précédemment. Ces bases de données permettent de travailler à l'aide de scikit Learn.

	label	703.36253	715.39917	735.33054	785.37665	836.42224
autruche_s1	birds	0	0	0	0	793801
autruche_s2	birds	0	0	0	0	824694
autruche_s3	birds	0	0	0	0	860992
oiseau_os_act16_s1	birds	0	0	0	0	442070
oiseau_os_act16_s2	birds	0	0	0	0	440775
oiseau_os_act16_s3	birds	0	0	0	0	0
pigeon_os_act21_s1	birds	0	0	0	0	729520
pigeon_os_act21_s2	birds	0	0	0	0	576213
pigeon_os_act21_s3	birds	0	0	0	0	556123

Figure 4 : échantillon d'une base de données de spectre concaténées

Les analyses déjà effectuées sur les bases de données ont consisté à faire du clustering hiérarchique à l'aide d'apprentissage non supervisé. Elles ont résulté en la présentation de graphiques nommés dendrogrammes (cf. fig5).

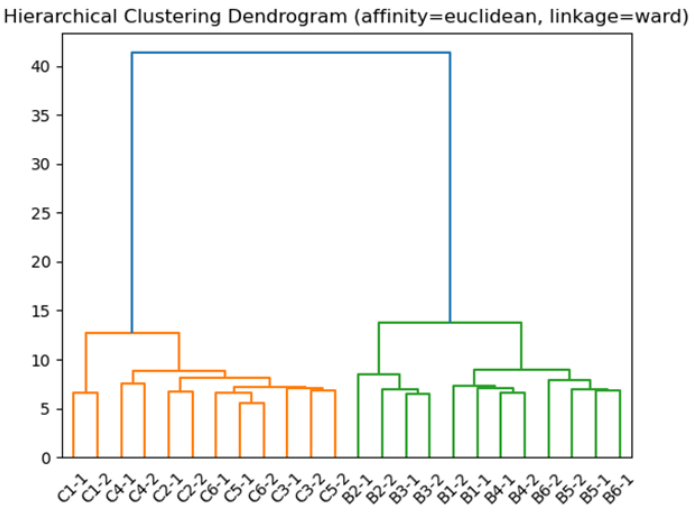


Figure 5 : Dendrogramme scikit learn

Ma première mission fut alors de reproduire ce travail afin de mieux comprendre les outils, les bases de données de travail et émettre mes propres hypothèses.

B) Prise en main de Metaboanalyst

Qu'est-ce que Metaboanalyst ?

Metaboanalyst, est un outil spécialisé dans l'analyse spectrométrique conçu sur une base R. Celui-ci propose une palette de ses fonctionnalités sur un site internet interactif. Sur ce site nous pouvons uploader nos fichiers afin de réaliser différentes analyses d'apprentissage supervisé ou non supervisé.

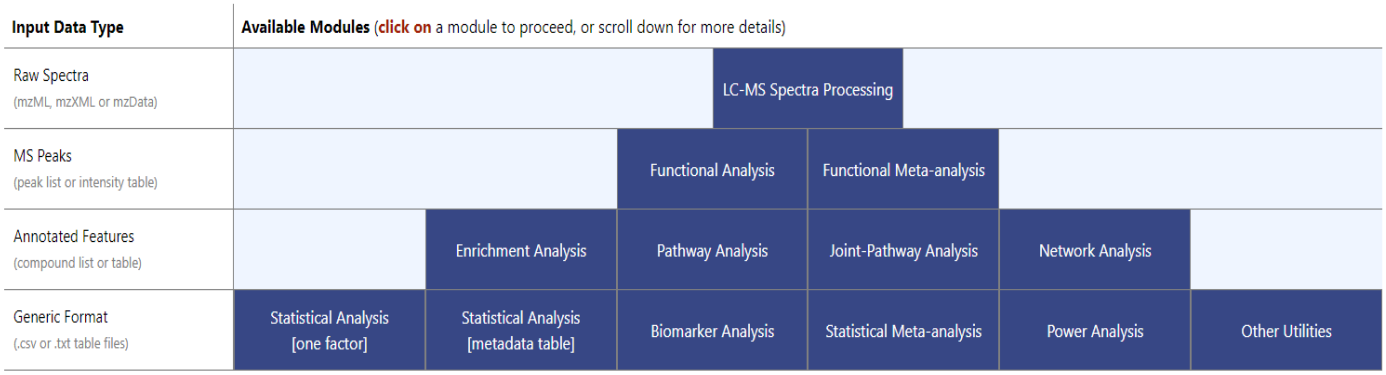


Figure 6: Différents modules d'analyses de Metaboanalyst

Le module qui nous intéresse dans le cadre de notre travail, est Statiscal Analysis. Comme nous pouvons le voir ci-dessous (fig 7), ce module permet d'utiliser facilement plusieurs types d'analyses et notamment l'analyse de cluster avec des dendrogrammes ou encore le Kmeans. Plus de précision sont fourni en **annexe 1**.

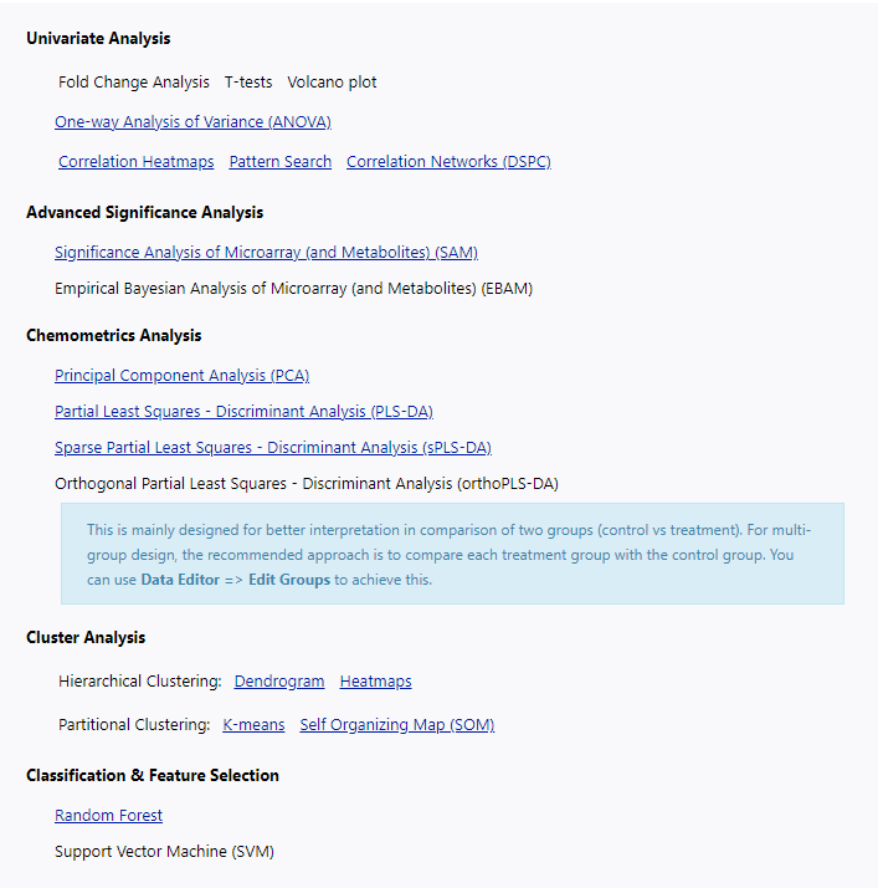


Figure 7 : Les différents outils d'analyse proposer par le statistical analysis

Inconvénients de Metaboanalyst

Metaboanalyst comporte plusieurs inconvénients comparés à une manipulation des données non guidés :

- Prend forcément en compte les intensités des pics (à moins de les supprimer des données)
- Applique certains pré-traitements de manière opaque (certains échantillons sont éliminés)

Cela peut fausser les résultats si on compare les échantillons anciens et récents, d'où l'idée de travailler sur des données normalisées ainsi que sur scikit learn.

C) Exécution de Machine Learning sur les données

1) Apprentissage non supervisé

En machine Learning, la technique de l'apprentissage non supervisé consiste à entraîner des modèles, sans réaliser d'étiquetage des données au préalable. Les algorithmes regroupent les données en fonction de leur similitude, détectent des données ou individus présentant des caractéristiques ou des structures similaires.

Les modèles d'apprentissage non supervisé sont notamment utilisés pour :

- Le classement des données
- Le calcul approximatif de la densité de distribution
- La réduction des dimensions

Dans mon travail d'exploration des travaux précédents, il fut intéressant de reproduire ce qui avait déjà été réalisé, soit du clustering à l'aide dendrogramme.

J'ai commencé à travailler avec la base de données du second datasets contenant 29 espèces avec une normalisation 1 et 0, c'est-à-dire que chaque intensité des pics exprimés par le spectre est traduite par un 1 et ceux non exprimés par un 0.

Scikit-learn

Scikit-Learn est une librairie python conçue afin de faire du Machine Learning. Open-source et fort d'une communauté active, Scikit-Learn s'est imposé comme le meilleur outil disponible pour le travail des data Scientists. Il permet notamment de répondre à des problématiques de classification, régression, regroupement, réduction de dimensionnalité, de sélection de modèle ou encore de prétraitement.

Le clustering

Le clustering a pour objectif de séparer des données en groupes homogènes ayant des caractéristiques communes. Cela peut donc nous donner une première idée des différentes caractéristiques de nos données.

Le Dendrogramme est donc le type de diagramme en arborescence que l'on utilise pour présenter le clustering hiérarchique. Chaque nœud de l'arbre représente un cluster.

Lors de la construction d'un dendrogramme on a le choix entre plusieurs algorithmes de clustering et de mesure de distance, comme nous pouvons le voir avec les options disponibles pour Metaboanalyst en annexe 1 (exemple de résultat). Celles offertes par scikit learn sont sensiblement les mêmes.

On peut également mesurer la qualité du dendrogramme à l'aide du score de silhouette, ce qui permet de faire un choix sur les métriques à utiliser. La valeur du score Silhouette varie de -1 à 1. Si le score est de 1, le cluster est bien séparé des autres clusters. Une valeur proche de 0 représente des clusters qui se chevauchent avec des échantillons très proches de la frontière de décision des clusters voisins. Une note négative [-1, 0] indique que les échantillons ont peut-être été affectés aux mauvais clusters.

Après avoir essayé toutes les combinaisons possibles on peut alors, dresser le tableau suivant (cf.tb1) affichant les différents scores de silhouette en fonction des métriques utilisées :

Tableau 1: Comparaison des scores de silhouette

	Euclienne	Manhattan
complete	0,172	0,307
average	0,282	**0,465**
Single	0,225	0,379
ward	0,096	

Comme nous pouvons le voir ici, les meilleurs scores obtenus apparaissent lorsqu'on utilise la distance de Manhattan, ce qui est vraisemblablement lié au fait que l'on travaille sur une base de données normalisée en binaire et notamment avec la méthode average. Les résultats les plus satisfaisants sont obtenus lors de l'utilisation de la métrique de « Manhattan » ainsi que de la méthode « complete » (plus simple d'interprétation).

Nous observons sur ce dendrogramme (annexe 3) 4 clusters ; un grand cluster qui semble contenir les échantillons de mammifères, un autre cluster qui contient uniquement les échantillons de poissons (plus particulièrement les différentes espèces thon) ainsi que 2 autres clusters qui semblent être proches les échantillons de poulets et d'autruches qui pourraient s'apparenter aux espèces d'oiseaux.

Il apparait visuellement une corrélation dans les clusters en fonction du type d'échantillon extrait des différentes espèces. Dans le cluster « mammifère » la majorité des échantillons provient d'os : de crane et d'os dont la source n'est pas toujours définie.
Lorsqu'on observe les échantillons de thon, on s'aperçoit que les extraits de vertèbre sont réunis en un cluster (cluster poisson) or 3 échantillons extraits d'écailles sont classés dans le cluster « mammifère ». On se demande alors si le dendrogramme a créé ses classes en identifiants des similitudes sur les types d'os et non pas les types d'espèces. Ce constat pourrait expliquer pourquoi les échantillons d'écailles de thon et d'os de pigeon (dont la provenance de la source osseuse est non spécifiée) se situent dans le cluster des mammifères et pas dans le cluster correspondant à leur type d'espèce.
Nous pouvons ainsi distinguer différents types d'os : os plat, os court, os long et irrégulier. Donc, la question se pose s'il existe pour chacun de ses os des spécificités que partagent l'ensemble du vivant qui pourrait alors compliquer l'analyse. Cette spécificité pourrait se manifester en fonction du rôle de celui-ci dans le squelette (fonction de soutien, de protection ou de mouvement par exemple). **La question se pose s'il n'est pas nécessaire de prélever des échantillons uniquement sur les mêmes types d'os**.

Cette hypothèse fut exposée lors d'une réunion en présence des paléontologues, qui l'ont prise en considération et ont également ajouté que les pics de spectromètres pouvaient varier en fonction de l'endroit où a été prélevé l'échantillon sur un même os.

Réduction de dimensionnalité

La réduction de dimensionnalité consiste à réduire le nombre de variables. Si les données sont représentées dans un tableau, la réduction de dimensionnalité passera par une diminution du nombre de colonnes. La réduction de dimensionnalité va nous permettre de mieux décrire les données dont on dispose.

Parmi ces méthodes, on trouve notamment l'analyse en composantes principales (ACP), une des plus rependu, c'est avec cette technique que nous avons obtenu nos premiers résultats.

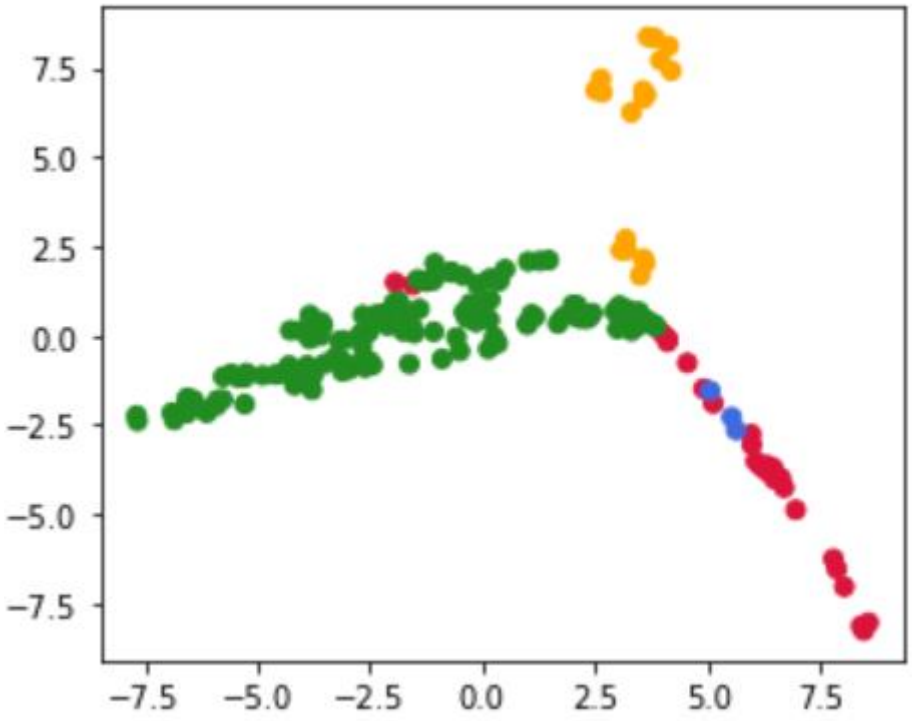


Figure 8: Réduction de dimensionnalité avec l'ACP

Sur cette figure (cf. fig8) on peut observer des points de différentes couleurs. Les poissons sont figurés en rouge, en vert les mammifères et en jaunes les oiseaux, les 3 points bleus correspondent à 3 observations dont on ne connait pas l'espèce.
D'après l'ACP ainsi que le dendrogramme présenté précédemment, ces observations correspondent à des échantillons de poissons. Cette ACP, met en évidence que de nombreuses espèces partagent parfois des caractéristiques similaires et qu'il est difficile de distinguer des clusters distincts pour le type d'espèce.
À la suite de la réalisation de cette ACP, il m'a été conseillé 2 autres méthodes de réduction de dimensionnalité : TSNE et UMAP.

UMAP (Uniform Manifold Approximation and Projection) est une technique de réduction de dimension qui peut être utilisée pour la visualisation de la même manière que t-SNE (T-distributed Stochastic Neighbor

Embedding). Contrairement à l'ACP considéré comme un outil de compression linéaire, Umap et t-SNE sont des outils de réduction de dimensionnalité non linéaires.

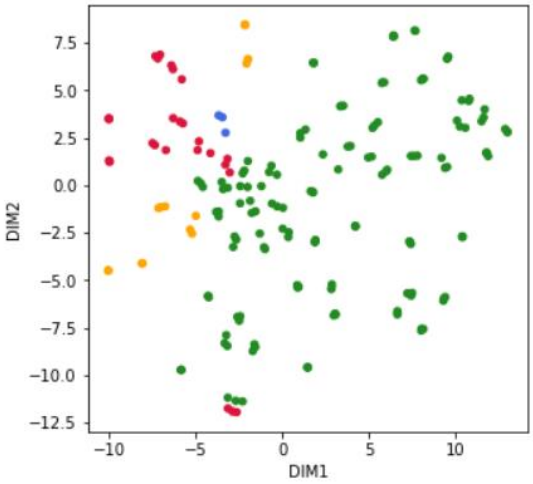


Figure 10: Réduction de dimensionnalité avec TSNE

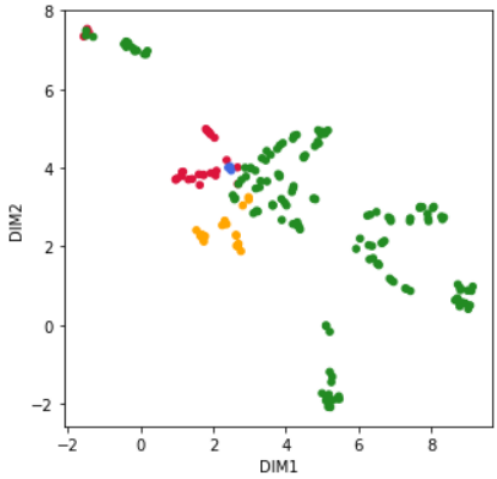


Figure 9: Réduction de dimensionnalité avec UMAP

Comme nous pouvons le voir ci-dessus ces 2 méthodes ne se sont pas avérés fructueuses ; l'interprétation n'est pas aisée, les clusters ne sont pas nettement visibles sur 2 dimensions. Effectivement, on observe un enchevêtrement entres les points de différentes couleurs qui devraient dans le cas d'une bonne clusterisation former des groupes distincts de mêmes couleurs.

2) Apprentissage supervisé

Contrairement à l'apprentissage non supervisé, dans le cadre de l'apprentissage supervisé, l'ordinateur connaît déjà les réponses qu'on attend de lui. On travaille alors à l'aide de données étiquetées. Les algorithmes vont être entrainés à l'aide des bases de données. Dans notre cas, ces algorithmes vont comprendre quelles sont les caractéristiques permettant de classer les spectres dans leurs espèces correspondantes. Donc, au fur et à mesure de l'addition de nouvelles données, l'algorithme pourra l'identifier et émettre une probabilité de correspondance à une classe.

Mesurer la performance d'un modèle

Afin de mesurer la performance d'un modèle de Machine Learning. On divise ainsi le Dataset aléatoirement en deux parties avec un rapport 80/20 :

• **Train set** (80%), qui permet à la machine d'entraîner un modèle.

• **Test set** (20%), qui permet d'évaluer la performance du modèle.

Cette étape fait suite au pre-processing (suppression d'anomalies dans les données, normalisation, etc…) des données. Pour créer un Train set et Test set à partir de nos bases de données, on utilise la fonction train test split de Sklearn.

Le train test split divise des tableaux ou des matrices en sous-ensembles d'entraînement et de test aléatoires. Ce caractère aléatoire crée un problème dans l'entrainement d'algorithmes, par exemple le test set pourrait se voir attribuer aucune observation d'une certaine classe. Dans notre cas, nous avons peu d'observations, il est indispensable de stratifier l'échantillonnage pour s'assurer que les fréquences relatives des classes sont approximativement préservées. On peut simplement ajouter le paramètre **stratify** en indiquant notre table de cibles dans la fonction train test split de Sklearn pour y parvenir.

Choisir le bon modèle

La partie la plus difficile de la résolution d'un problème de Machine Learning est souvent de trouver le bon estimateur. Pour rappel, un estimateur est un objet qui correspond à un modèle basé sur certaines données d'apprentissage et est capable de déduire certaines propriétés sur de nouvelles données. Pour commencer, nous pouvons nous référer au guide proposé par scikit-learn. S'y trouve une carte qui peut nous donner une idée du type d'algorithme à choisir :

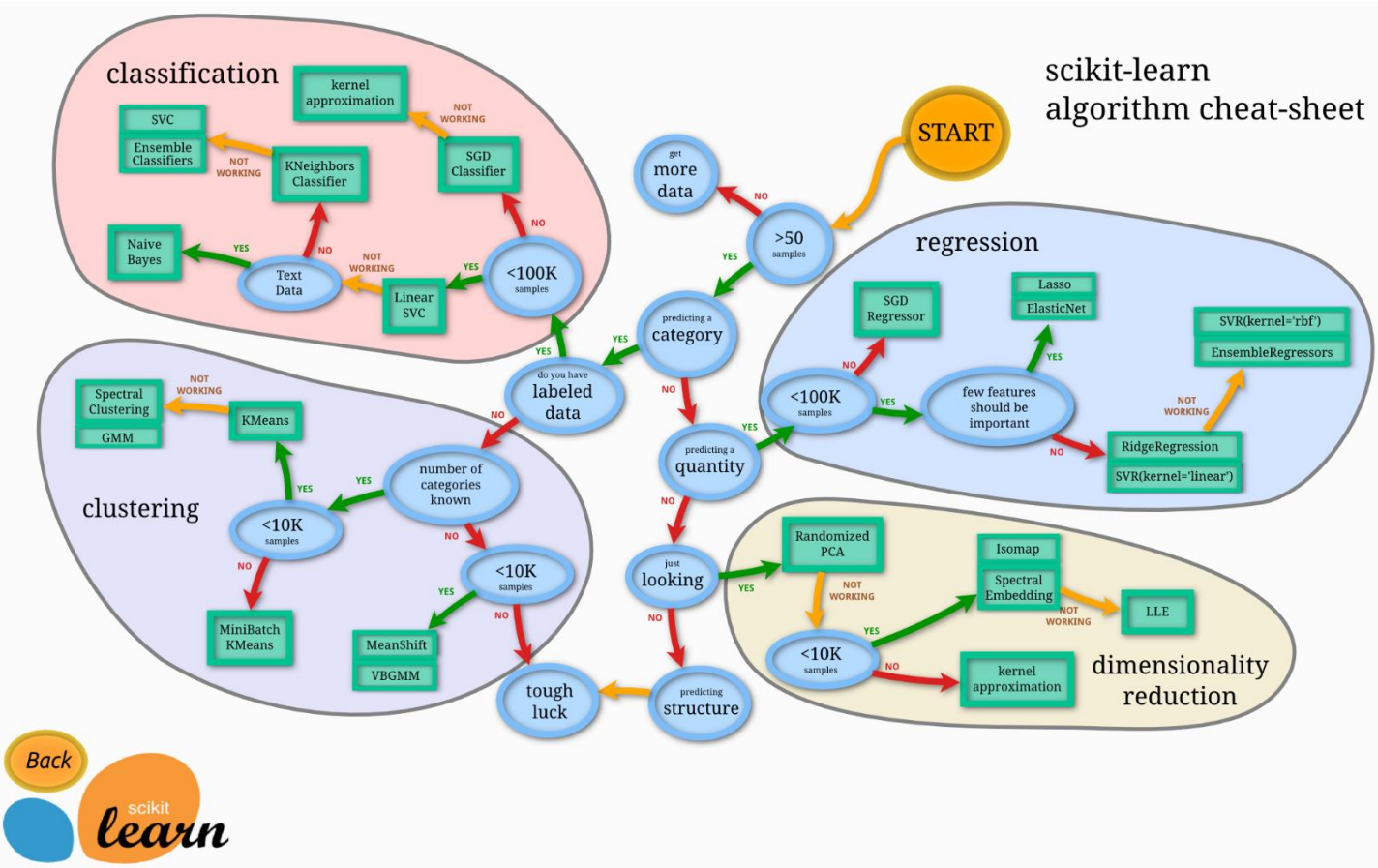


Figure 11: scikit-learn algorithm cheat-sheet

Nous cherchons à réaliser un travail de classification, nous avons moins de 100 000 échantillons et nos données ne sont pas textuelles. Si l'on s'en tient à cette carte, on devrait commencer par résoudre notre problème à l'aide d'un estimateur de type **support vector machine**, en cas d'échec avec **KNeighbors Classifier** ou encore des estimateurs de type **Ensemble Classifiers** comme le **randomforest**.

Afin d'analyser les performances de nos modèles nous utilisons différentes métriques basées sur les concepts des vrais positifs et faux négatifs :

- Le ***recall*** (rappel), qui permet de savoir le pourcentage de positifs bien prédi par notre modèle. Il est calculé en divisant le nombre de vrai positif par le nombre de vrai positif et de **faux négatif**. Quand le recall est élevée (proche de 1), cela veut dire que le modèle classe bien les positifs, mais cela n'indique pas sa qualité de prédiction sur les négatifs.
- La ***precision*** (la précision) permet de connaître le pourcentage de prédictions de positifs correcte. Elle est calculée en divisant le nombre de vrai positif par le nombre de vrai positif et de **faux positif.** Quand la précision est élevée (proche de 1), cela veut dire que le modèle minimise les faux positifs et donc que les prédictions positives sont bien prédites.

- **Le F1-score** est la combinaison du recall et de la précision, il permet de réaliser une évaluation fiable de la performance de notre modèle. Pris indépendamment, le recall et la precision ne sont pas utiles. Il est calculé en multipliant par 2 le recall fois la précision le tout divisé par le recall plus la précision. Donc, plus le F1 score est proche de 1 plus notre modèle est performant.
- **L'accuracy** permet de décrire la performance du modèle en mesurant le taux de prédictions correctes sur l'ensemble des échantillons.

Tableau 2 : Comparaison des métriques associées à 3 modèles

SVC				
	precision	recall	f1-score	support
birds	1.00	1.00	1.00	4
fishs	1.00	0.83	0.91	6
mammals	0.97	1.00	0.98	29
accuracy			0.97	39
macro avg	0.99	0.94	0.96	39
weightedavg	0.98	0.97	0.97	39
KNN				
	precision	recall	f1-score	support
birds	1.00	1.00	1.00	4
fishs	1.00	0.83	0.91	6
mammals	0.97	1.00	0.98	29
accuracy			0.97	39
macro avg	0.99	0.94	0.96	39
weighted avg	0.98	0.97	0.97	39
Randomforest				
	precision	recall	f1-score	support
birds	1.00	1.00	1.00	4
fishs	1.00	1.00	1.00	6
mammals	1.00	1.00	1.00	29
accuracy			1.00	39
macro avg	1.00	1.00	1.00	39
weighted avg	1.00	1.00	1.00	39

Le tableau ci-dessus (cf. tab2) est une comparaison de 3 modèles de classification ayant pour objectif de prédire le type d'espèce sur les données issues du second datasets normalisé par 0 et 1 (avec les 3 échantillons idvertebreXXX considérés comme des poissons). Les données sont splittées en respectant une stratification en un **train set** correspondant à 80% de l'effectif total et un **test set** de 20%. On observe que le Randomforest réussi à classer parfaitement tous les échantillons en fonction de leur type (mammifères, oiseau et poisson). Contrairement aux essais de clusterings précédents les techniques d'apprentissages supervisés permettent de classer bien plus efficacement les échantillons en fonction de leurs types. Si on réitère le processus mais cette fois avec l'objectif de classer les échantillons directement par espèces, ça se

complique. Le nombre de classe étant bien plus élevée les algorithmes vont avoir beaucoup plus de difficultés à réaliser les bonnes prédictions :

Tableau 3 : Comparaison des métriques associées à 3 modèles sur la classification des espèces

	Precision	Recall	F1-score	Accuracy
SVC	0,53	0,63	0,55	0,63
KNN	0,4	0,43	0,39	0,43
Randomforest	0,81	0,88	0,83	0,88

Le tableau ci-dessus (cf.tab3) est une comparaison de 3 modèles de classification ayant pour objectif de prédire l'espèce (parmi 29) sur les données issues du second datasets normalisé par 0 et 1. On observe des scores inférieurs à l'analyse précédente. Le Randomforest reste toujours plus performant que le SVC et le KNN mais cette fois-ci avec 88% d'accuracy et un F1-score de 0,83 donc bien inférieur que précédemment. Notre objectif est de maximiser ces métriques, pour cela nous pouvons tester de nouveaux algorithmes et modifier leurs paramètres (exemple, changer le nombre de branches d'un RandomForest).

Avec les résultats du Randomforest, nous sommes amenées à penser qu'il est possible d'utiliser les spectres pour déterminer l'espèce d'un ossement.
On doit maintenant augmenter la précision des estimateurs à travers la maximisation des métriques vu précédemment. Pour cela nous pouvons tester de nouveaux algorithmes et modifier leurs paramètres (exemple, changer le nombre de branches d'un RandomForest).

Pipeline et cross validation

Avant tous propos sur l'outil qui va nous permettre de trouver les meilleurs algorithmes, il est important de comprendre la notion du pipeline et de Cross-validation en Machine Learning et notamment sur Scikit-Learn :
La cross-validation, consiste à entrainer et valider nos modèles sur plusieurs découpes de notre **train set**. Cela nous permet d'éviter les cas d'overfitting (le modèle s'est trop perfectionné sur les données du train set). La fonction GridserachCV de Scikit-Learn, permet de trouver les meilleurs paramètres d'un estimateur en comparant les performances de chaque combinaison à l'aide de la technique de cross-validation.

Le pipeline est une méthode qui permet d'éviter les séquences fixes d'étapes dans un projet de machine Learning par exemple la normalisation ou le choix d'un estimateur. La fonction make_pipeline de Scikit-Learn permet de créer des pipelines et comporte plusieurs avantages : simple à utiliser, plus sécurisé (évite les problèmes de mauvaise transformation de données avec des oversampleurs ou normalisateurs) et d'effectuer des **cross-validations** sur tout le pipeline.

Teste préliminaires de TPOT et amélioration de l’accuracy

TPOT est une librairie python construit sur Scikit-Learn qui permet de réaliser de l’auto-ML. C’est-à-dire, automatiser la création de modèle de machine Learning. TPOT explore intelligemment des milliers de pipelines possibles pour trouver le meilleur pour nos données. Une fois que TPOT a fini de chercher (ou alors que celui s’est arrêté sous notre ordre), il nous fournit le code Python du meilleur pipeline qu'il a trouvé.
Par exemple, si l’on exécute TPOT sur notre base de données pendant un temps donné ou non, il va nous retourner le code suivant (voir annexe 4 TPOT). A l’aide de l’estimateur MLPclassifier et des paramètres indiqués par TPOT, l’accuracy atteinte après cross-validation est de 0,94 ce qui est déjà nettement supérieur à ce qui a pu être observé précédemment.

IV. Détermination automatique des espèces à partir d'un prélèvement osseux

A) Recentrage de l'objectif

L'objectif initial du stage, est de réaliser une étude de faisabilité sur la détermination d'espèce à l'aide de Machine Learning et de données spectrométriques sur les 29 espèces transmises. Or, à la suite de la réunion du 3 juin, nous a été communiqué des précisions sur l'objectif des paléontologues. Celui-ci est de déterminer s'il est possible d'identifier précisément une espèce au sein d'un groupe, parmi les 4 groupes suivants :

- Le groupe bos contenant ; le bœuf, l'Auroch, le bison
- Le groupe cervus contenant ; le Mega Ceros, le daim, le cerf
- Le groupe ovis contenant ; la chèvre et le mouton
- Le groupe sus contenant ; le porc et le sanglier

D'après les paléontologues il est déjà possible de différencier une chèvre d'un mouton, il est cependant beaucoup plus complexe de différencier les espèces des autres groupes. Par exemple, les archéologues ont identifié le groupe bos, mais ne vont pas savoir s'il s'agit d'un Boeuf, d'un Auroch ou d'un Bison.
Ces travaux de recherches répondent notamment à la difficulté d'identifier certains types d'espèces notamment sur des échantillons d'os de petites tailles.
Il sera également important d'identifier quelles sont les pics permettant la différenciation de ces différentes espèces. Pour cela, il faudra apporter une grande importance à utiliser des modèles permettant une interprétation.

B) Organisation et préparation des données

Durant les 2 premières semaines de juin, des données nous sont parvenus au compte-gouttes afin de pouvoir mieux répondre aux nouveaux objectifs. Parmi ces données beaucoup furent non exploitable, impossibilité d'identifier l'espèce correspondant au spectre.
Après avoir organisé les différents datasets, un recensement des différentes espèces utiles fut effectué.

Tableau 4: recensement du nombre d'échantillon par espèce

Bos		Cervus	
Auroch	22	Cerf	10
Bison	2	Daim	12
Bœuf	40	Mégacéros	7
Sus		**Ovis**	
Porc	8	Mouton	9
Sanglier	12	Chèvre	2

Comme nous pouvons le voir ci-dessus (cf. tab4), certaines espèces sont sous représentés ce qui va rendre l'analyse complexe. Nous travaillerons alors uniquement sur les espèces ayant un effectif suffisant c'est-à-dire :

- L'auroch et le bœuf (sans le bison car sous représentés)
- Le cerf, le daim et le mégacéros
- Le porc et le sanglier

Nous ne pourrons pas répondre à l'objectif visant différencier le mouton et la chèvre étant donné le sous-effectif d'échantillon de chèvres.

Créer une base de données fonctionnelle

Jusqu'à présent le travail a été effectué à l'aide de bases de données fonctionnelles transmises. Avec l'apport de nouvelles de données j'ai été amené à les construire. Les spectres étant transmis dans plusieurs fichiers il fut nécessaire de les concaténer.
De nouveaux fichiers ont donc été réalisé ; dans chaque colonne, un rapport m/z ainsi que son intensité associée pour chaque échantillon en ligne (NA = non-présence de ce rapport m/z dans l'échantillon). La difficulté de ces nouvelles réalisations consistait dans la sélection des variables M/Z. En effet, pour chaque fichier reçu correspondant ou non à la même espèce, certains rapports m/z sensés décrire le même pic ne seront pas égaux mais très proches. Par exemple on va retrouver le rapport m/z 758.3665 pour un échantillon A et 758.3682 pour un échantillon B. En réalité ces 2 rapports traduisant un même pic, il faut trouver le moyen de réunir ces différents rapports m/z en un seul.
Après avoir éprouvé des difficultés à réunir les différents pics sur python, il s'est avéré qu'il était possible d'effectuer ce travail plus facilement avec Metaboanalyst. Si on uploade des fichiers sur le site comme expliqué en annexe, une étape de traitement des pics contenant 2 paramètres devient possible.
Le paramètre de Mass tolérance devient intéressant ; c'est ce qui va permettre de définir la largeur des tranches m/z qui se chevauchent à utiliser pour regrouper les pics sur les échantillons. De base la Mass tolérance est réglé a 0,025 donc toute les variables (rapports m/z) ayant comme écart ce nombre vont être appariés. Après cela on peut télécharger la base donnée et travailler dessus avec d'autres outils (Scikit-Learn).

C'est après quelques jours de travail avec ces bases de données qu'un nouveau problème est apparu. Le nombre de pics de ces bases de données étaient parfois faible (parfois une centaine alors que l'on a concaténé des dizaines d'espèces ayant tous entre 250 et 350 pics). Comme évoqué, précédemment Metaboanalyst comporte plusieurs inconvénients et notamment certains pré-traitements non modifiables.
Ainsi, lors de l'utilisation de la fonction GroupPeakList qui permet de définir le temps de rétention, la mass tolerence, on s'aperçoit qu'un autre paramètre non modifiable intervient. Celui-ci a pour effet d'exclure les pics pour lesquelles aucune classe (espèce définie par un dossier dans le fichier zippé uploader) n'a au moins la moitié de ses échantillons représentés.

Afin de régler ce problème et donc utiliser tous les paramètres que propose cette fonction nous avons décidé d'utiliser le paramètre minfrac de la fonction GroupPeakList. Pour se faire, il faut utiliser MetaboAnalyst R afin de pouvoir coder soit même. De nombreux problèmes apparaissent lors de l'installation de MetaboAnalyst : packages non à jour, etc...

Après relecture des travaux déjà effectués, une solution s'est offerte à nous. Effectivement, la personne ayant déjà travaillé sur ce problème a créé une image docker permettant de créer une table de features à partir d'une archive zip.

Docker

Docker est une plateforme, permettant de lancer certaines applications dans des « conteneurs logiciels », ce qui permet d'empaqueter des applications ainsi que leurs dépendances. On peut alors grâce à Docker installer des dépendances facilement et cela sans problème de compatibilité.
Afin de pouvoir utiliser les fonctionnalités citées précédemment, nous allons faire appel au Docker iguigon/metaboanalystr_feature_table. Plus de détail sur comment faire fonctionner le docker sur Ubuntu en annexe 2.

Contenu des bases de données

3 bases de données seront alors créées à l'aide de l'image docker afin de répondre aux 3 objectifs de différenciation. Chaque base de données se verra attribuer plusieurs classes, une classe par espèces différentes et une classe autre.
La classe « **Autre** » permettra de détecter les échantillons qui ne correspondent pas aux espèces de l'objectif.

C) Machine Learning

OPLS DA

Un premier algorithme a attiré notre attention au début de nos analyses. Cet algorithme c'est l'OPLS DA (Analyse discriminante orthogonale des moindres carrés partiels), disponible nativement dans les outils proposés par Metaboanalyst, il permet de discriminer 2 classes à l'aide de données multivariées et nous dira quelles sont les variables ayant le plus grand pouvoir discriminant.

Comme nous pouvons le voir dans la figure suivante (cf. fig12), OPLS DA permet l'identification des espèces très efficacement, les échantillons sont bien répartis dans 2 clusters distincts.

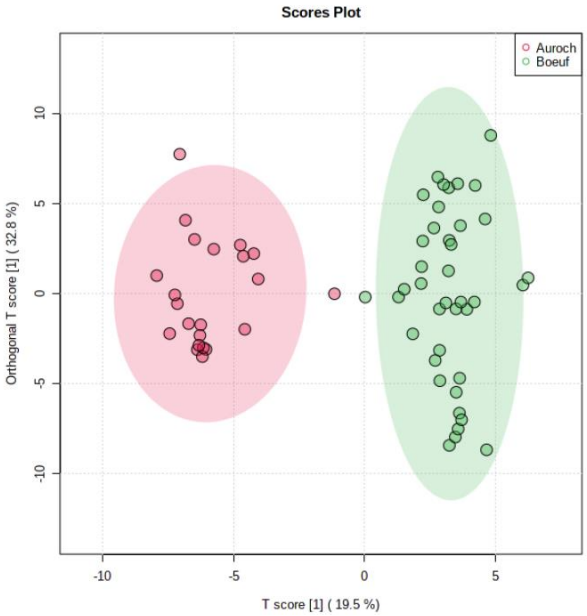


Figure 12: OPLS DA sur les boeufs et les aurochs

Après des tests sur différentes espèces ainsi que sur scikit-learn, les résultats se sont toujours avérés satisfaisant. Surpris par cette étonnante efficacité comparée aux autres algorithmes comme la PCA (cf. fig 13), il fut indispensable de plus se renseigner sur celui-ci.

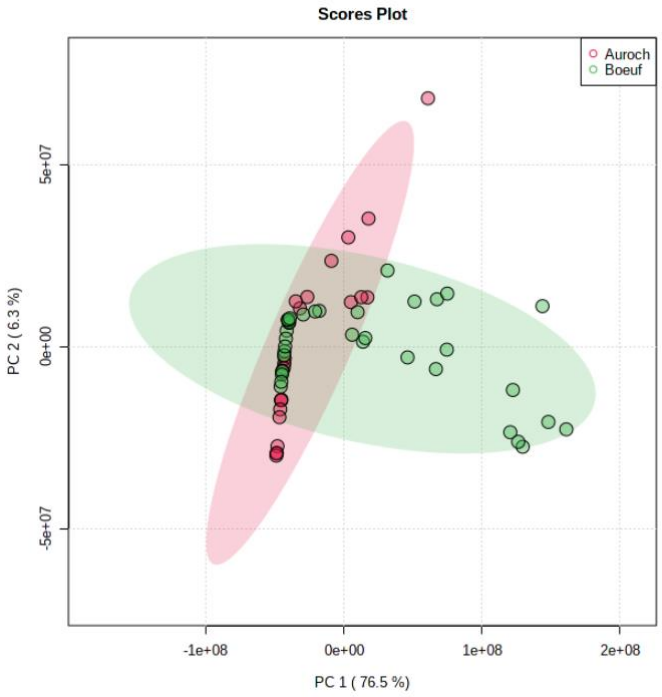


Figure 13: PCA sur les boeufs et les aurochs

Lors de la lecture de la publication « PCA as a practical indicator of OPLS-DA model reliability » publié part Bradley Worley et Robert Powers on apprend que : « Les séparations de groupes observées et la signification du modèle dans (O)PLS est une source d'erreur en métabolomique qui a souvent conduit à la publication de résultats peu fiables. Nos résultats illustrent que si un modèle PCA ne parvient pas à atteindre une séparation nette des groupes, un modèle OPLS-DA ultérieur, malgré toute apparence de séparation de groupe, est souvent peu fiable ou invalide (paragraphe traduit de l'anglais). »

Si on se réfère à cette publication, en effet les résultats de nos OPLS-DA sont invalides étant donnée le résultat des PCA (cf. fig13).

Comparaison de modèles sur des bases normalisés et non normalisés

Avec le nouveau jeu de données, on cherche à trouver le meilleur estimateur et comparer les performances sur des données normalisées.
Il est important de préciser qu'il est préférable de travailler sur une base de données normalisée afin de mettre tous les échantillons à la même échelle. En effet, d'après les travaux réalisés en amont, l'intensité des pics peut diminuer en fonction de l'âge de l'échantillon et même de la protéine que caractérise le pic, donc biaiser l'interprétation des estimateurs. Il sera important de vérifier que la perte d'information engendrée par la normalisation a un faible impact. Nous travaillerons sur 2 types de normalisation :

- Une normalisation « **1 et 0** », c'est-à-dire que chaque intensité des pics exprimés par le spectre est traduite par un 1 et ceux non exprimés par un 0.
- Une normalisation « **Max(x)/x** », c'est-à-dire que pour chaque échantillon chaque valeur d'intensité a été divisée par la valeur de l'intensité maximale de l'échantillon, de sorte que pour chaque échantillon, le pic majoritaire soit toujours d'une valeur de 1.

Pour commencer, on peut simplement comparer les performances d'un Randomforest :

Tableau 5 :

	Non normalisée	**Normalisée 1-0**	**Normalisée Max(x)/x**
Accuracy	**0,92**	**0,91**	**0,9**
F1_score	**0,887**	**0,874**	**0,869**

Dans le tableau ci-dessus (cf.tab5) on observe la comparaison des performances d'un Randomforest sur la base de données Cervus. On observe une faible baisse des performances du modèle lorsqu'on passe sur des données normalisées. La normalisation par 1 et 0 est plus performante d'1% d'accuracy comparé à la Max(x)/x. Sera-t-il plus intéressant de travailler sur ce type de normalisation à l'avenir ?
Nous pouvons également utiliser TPOT pour trouver des pipelines plus performants. Afin d'avoir une comparaison valable, il est nécessaire de lancer TPOT sur les bases de données normalisées et non normalisées. Cela implique 3 exécutions de TPOT multiplié par le nombre de base de données qui est de 3. On a donc 9 exécutions de TPOT à réaliser. Les calculs peuvent durer plusieurs heures ou jours, ce qui n'a pu être réalisé. À titre indicatif, pour exprimer les performances atteignables avec un pipeline optimiser par TPOT, 0,98 d'accuracy a été atteint lors de l'exécution de celui-ci sur les espèces de type Cervus avec une normalisation de type Max(x)/x.

Evaluation de la performance des pics marqueurs

Début juillet, **l'équipe BONZAI** nous a transmis des listes de pics marqueurs pour certaines espèces. D'après eux, lorsqu'ils sont présents ces pics permettraient d'identifier l'espèce auquel ils correspondent. Effectivement, certains chercheurs sont capables d'identifier une espèce avec une simple inspection de certains pics. Il faut donc réaliser une version des différentes bases de données filtrés sur ces pics (donc sur quelques colonnes) afin d'effectuer nos tests.

Dans les listes, 3 animaux nous intéressent :

- Le Bos Taurus (bovin)
- Capra Hircus (chèvre)
- Sus cofra (sanglier)

Seul la liste de pics sur les bovins sera exploitée (pas suffisamment d'échantillons sur les chèvres, liste non complète pour les sangliers).

Premièrement on effectue du clustering sur les données filtrées.

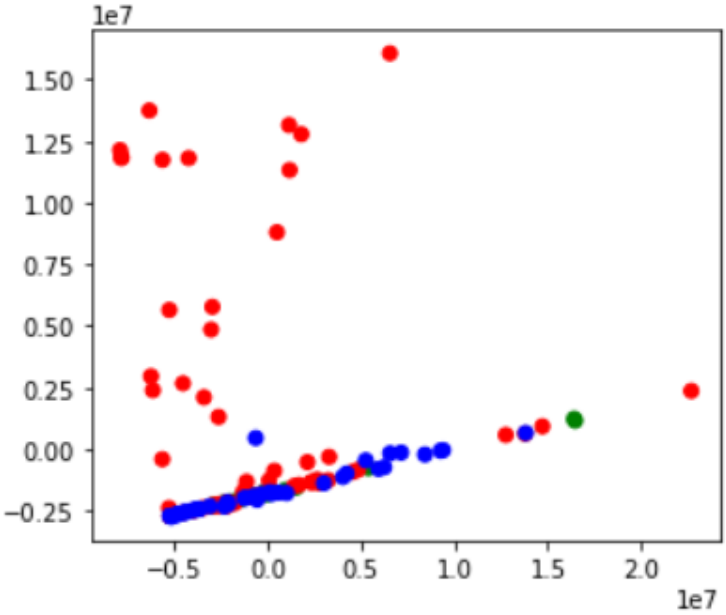


Figure 14 : PCA sur données filtrées non normalisés

Sur cette figure on peut observer une PCA. Les bœufs sont figurés en bleu, en vert les aurochs et en rouge les autres espèces (classe **Autre**). Nous observons les échantillons de bœufs et aurochs sont souvent concentré aux même endroit (beaucoup de points verts sous les points bleus). Donc d'après cette analyse les pics marqueurs des bœufs ne permettent pas de différencier les bœufs des aurochs. On peut l'observer plus facilement si l'on normalise la base de données et que l'on compare l'ACP avec la base de données non filtrées.

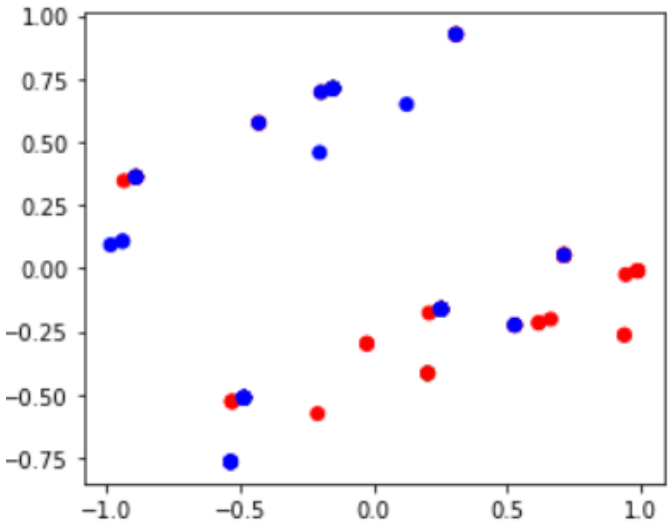


Figure 15 : PCA sur données filtrées normalisées 1 et 0

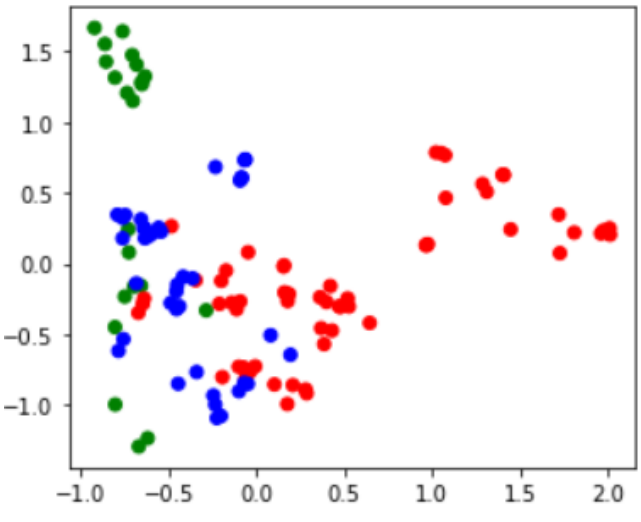


Figure 16 : PCA sur données non filtrées normalisées 1 et 0

La perte d'information du au filtrage rend impossible la différenciation des aurochs et des bœufs en comparaison avec la base de données non filtrées. On arrive au même constant lorsqu'on construit un dendrogramme.

Interprétation des modèles

L'étape la plus importante de notre activité de recherche est surement de comprendre le fonctionnement de nos modèles. Dans le monde de la science des données on nomme ce domaine **L'explainable AI**.
Comprendre le fonctionnement des modèles pourrait permettre d'identifier les pics permettant de différencier les espèces un peu de la manière dont le laboratoire **BONZAI** nous a transmis les pics marqueurs.
Pour cela, on peut utiliser différents outils :
Pour commencer on a le paramètre **features importances** Scikit-Learn, disponible sur certains modèles comme le Randomforest par exemple. Il permet d'attribuer un score aux caractéristiques en fonction de leur utilité pour prédire une variable.

Tableau 6: 10 plus fortes features importances sur bos à l'aide de Randomforest

Features	Importances
2883.42332	0,061816
2899.42206	0,0307
2487.20913	0,030616
2853.40783	0,029301
2767.32014	0,02153
1180.64381	0,020658
3084.41417	0,017546
1196.63859	0,01663
3100.41204	0,016502
1208.67479	0,016257

Le tableau ci-dessus (cf. tab6) nous donne les 10 meilleurs features importances lors de l'utilisation d'un Randomforest pour la base de données bos (normalisé en divisant l'intensité par la valeur de l'intensité maximale de chaque l'échantillon). Le problème c'est qu'on ne sait pas sur quelle classe influe cette features importances. Nous sommes en droit de nous demander en quoi ces pics pris individuellement orientent la prise de décision du modèle. Il faudrait réussir à réaliser une features importances pour chaque classe du modèle. 2 librairies pythons peuvent nous aider pour cela LIME et SHAP.

Lime permet d'expliquer n'importe quel classificateur même lesdites boîtes noires (modèle normalement difficile d'interprétation) tant qu'il accepte des tableaux Numpy (librairie python de gestion de tableaux de nombres) et génère une probabilité pour chaque classe. Néanmoins celui semble pouvoir expliquer le fonctionnement uniquement sur un échantillon à la fois. C'est-à-dire qu'il va expliquer sur un échantillon de test comment va fonctionner le processus de prédiction de la classe.

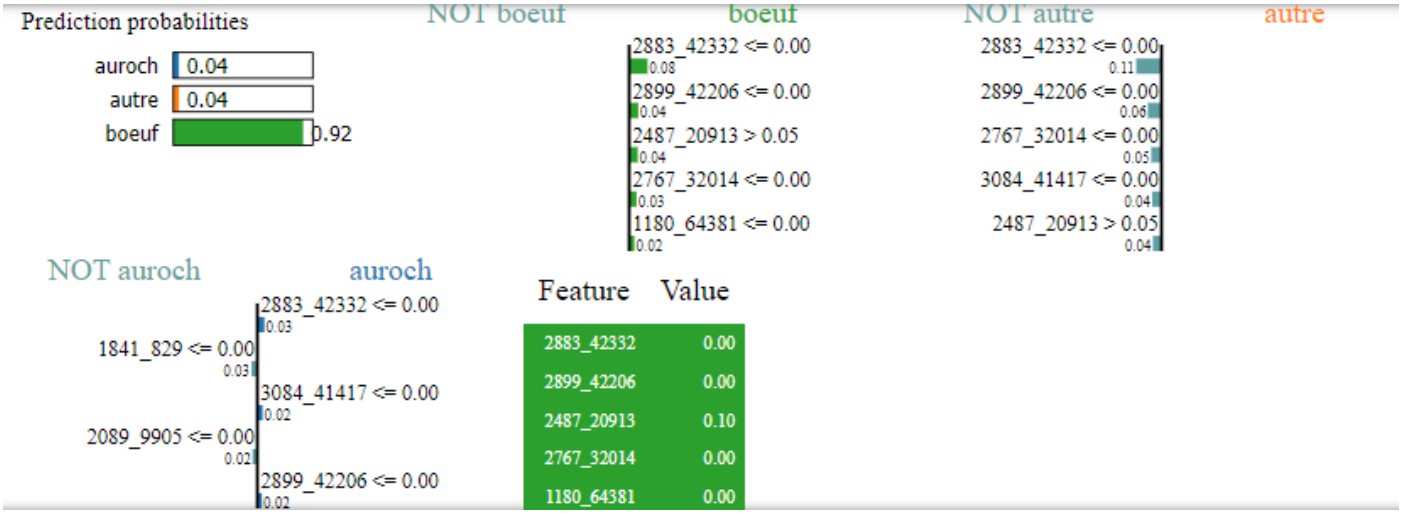


Figure 17 : Exemple de résultat Lime

Lorsqu'on utilise Lime avec un modèle Randomforest sur les espèces de type bos en Max(x)/x, on obtient les résultats ci-dessus (cf. fig17). On observe que l'échantillon analysé (choisi aléatoirement dans le test set) est identifié comme un bœuf avec 92% de précision : il sera donc classé bœuf d'après l'algorithme. On peut observer les features importances (on peut faire varier le nombre de features à afficher dans Lime) de chaque classe. Nous y trouvons les pics déjà observés précédemment avec features importances de sklearn. Dans le cas de cet échantillon la présence du pic 2487.20913 avec une intensité (normalisé) supérieure à 0,05 permet à l'algorithme d'orienter sa classification en faveur d'un bœuf. Les autres pics indiqués nous informent que leur absence oriente également le choix de l'algorithme en faveur de la classification vers un bœuf. **Peut-on considérer les pics comme le 2487.20913 comme des pics marqueurs du bœuf ?**

Shap semble être un outil plus complet, d'après sa documentation il pourrait en comparaison de Lime expliquer le fonctionnement des algorithmes sur l'ensemble des échantillons et faire la distinction du features importances pour chaque classe. On pourrait à l'aide de celui-ci savoir quelles sont les pics marqueurs de chaque espèce et les comparer à ceux transmis par le laboratoire **BONZAI.**
Pour l'instant, l'outil n'a pas encore été maitrisé, des difficultés à faire fonctionner la librairie sur des problématiques multi class sont observés.

Un autre outil (découvert la veille du rendu de ce rapport) **ExplainerDashboard**, qui est une librairie disponible sur python construit sur la base de Shap et qui est une alternative no code. En une seule ligne de code, nous avons accès à un Dashboard proposant plusieurs types d'analyses pertinentes pour notre travail notamment la features_importantces de chaque classe. Cette librairie pourrait être un très bon atout pour la suite du projet de recherche cependant sa véracité reste à prouver.

Conclusions et recommandations

En définitive :

- L'analyse préliminaire des données a permis de mettre en évidence que le type d'os et l'endroit du prélèvement peut exercer une grande influence sur les résultats, ce qui a été confirmé par les paléontologues
- Il n'est pas encore possible de travailler correctement sur les Bœufs, Aurochs et bisons ainsi que sur les chèvres et les moutons. Le nombre d'échantillon de bison et de chèvre est trop faible. Un jeu de données fut tout de même élaboré pour les groupes ayant des effectifs suffisants (3/4 sans les bisons)
- L'analyse des performances des pics marqueurs n'a pas pu être correctement effectué, le nombre d'espèce dont les pics marqueurs sont connus est pour l'instant trop faible. Il faudrait attendre que les groupes d'espèces correspondants aux objectifs (par exemple Bœuf, Auroch et bison) se voit attribué des pics marqueurs.
- L'interprétation des modèles n'a pas pu être mené à bien pour le moment. L'utilisation d'une machine bien inférieure en termes de puissance de calcul peut expliquer le temps qui fut nécessaire pour obtenir des résultats avec **TPOT** (résultat qui plus est peu satisfaisant). Un travail supplémentaire semble alors nécessaire à l'aide de celui-ci. On peut ajouter à cela une difficulté à comprendre les outils **LIME** et **SHAPE** dans des problématiques d'interprétation de modèles de classifications contenant plus de 2 classes. **ExplainerDashboard** pourrait être une méthode plus simple pour réaliser ce travail.

V. Les apports du stage

A) Compétence

Durant les 3 mois et demi de stage au sein du laboratoire CRIStAL et de l'équipe ORKAD, j'ai beaucoup appris. J'ai pu développer des compétences qui me seront indispensables dans mon futur professionnel, notamment :
Un approfondissement du savoir sur python, une meilleure maitrise des packages sklearn ou encore pandas. La découverte de nouveaux packages comme TPOT pour l'auto-ML. Également, toute la partie du travail sur Linux m'a permis de prendre conscience des possibilités que peut offrir ce système d'exploitation, notamment l'outil docker très simple d'utilisation sur linux. Pour finir, ce stage a grandement amélioré ma capacité à rechercher des informations ciblées, que ce soit du code ou recherche dans la littérature (thèse, publication) sur un sujet précis.

B) Apports généraux

Durant ce stage, j'ai pu également observer le fonctionnement d'une équipe de recherche en laboratoire. À travers les différentes réunions d'équipe ainsi que le GT optima (une réunion entre différentes équipes travaillant sur des problèmes d'optimisation), j'ai pu découvrir le travail des doctorants. De plus, j'ai eu la chance d'assister à une soutenance de thèse, celle de Weerapan SAE-DAN un doctorant de l'équipe qui présentait la configuration automatique d'algorithme pour l'optimisation combinatoire.

VI. Conclusion

Tout au long de ces trois mois et demi de stage, j'ai pu au sein du laboratoire CRIStAL et de l'équipe ORKAD, travailler sur un projet de recherche stimulant, permettant de me former pour les métiers de la data science. J'ai eu l'occasion de travailler sur des problématiques d'apprentissages supervisés et non supervisés, gestion et optimisation de bases de données et de prise en main d'outils spécialisés.

Au travers des différents travaux de recherches que j'ai pu effectuer, j'ai pu réaliser une esquisse de réflexion sur le projet qui pourra je l'espère aider à la réussite des objectifs que je n'ai pas pu atteindre. Parmi les travaux qui pourraient être utiles on compte notamment le travail de création de bases de données fonctionnelles et de recommandations.

Durant toute la période de ce stage j'ai bénéficié d'un très bon encadrement et d'une ambiance de travail stimulant mon intellect et ma curiosité.

Cependant il reste toujours beaucoup de chose à apprendre au sein du milieu de la recherche, j'aurais eu à cœur de continuer à travailler au sein du laboratoire CRIStAL et de l'équipe ORKAD sur les objectifs que nous nous étions fixés.

VII. Annexe

Annexe 1 : Fonctionnement de Metaboanalyst

Pour commencer, il faut se rendre à l'adresse suivante https://www.metaboanalyst.ca/home.xhtml. Ensuite, cliquez sur le bouton au centre de l'écran « Click here to start ».
Une fois rendu sur la page décrivant les différents modules que propose métaboanlyst, nous nous intéresserons au module Statistical Analysis. C'est alors que Metaboanalyst nous suggères d'uploader des fichiers (les spectres). La méthode la plus simple est par fichier zippé, on procède avec la méthode suivante : Dans l'onglet « A compressed file (.zip) » on sélectionne MS peak list, puis on uploade le fichier organisé de la manière suivante : Un dossier correspondant à chaque espèce ou catégorie d'espèce ainsi que leurs échantillons correspondants (cf. fig13).

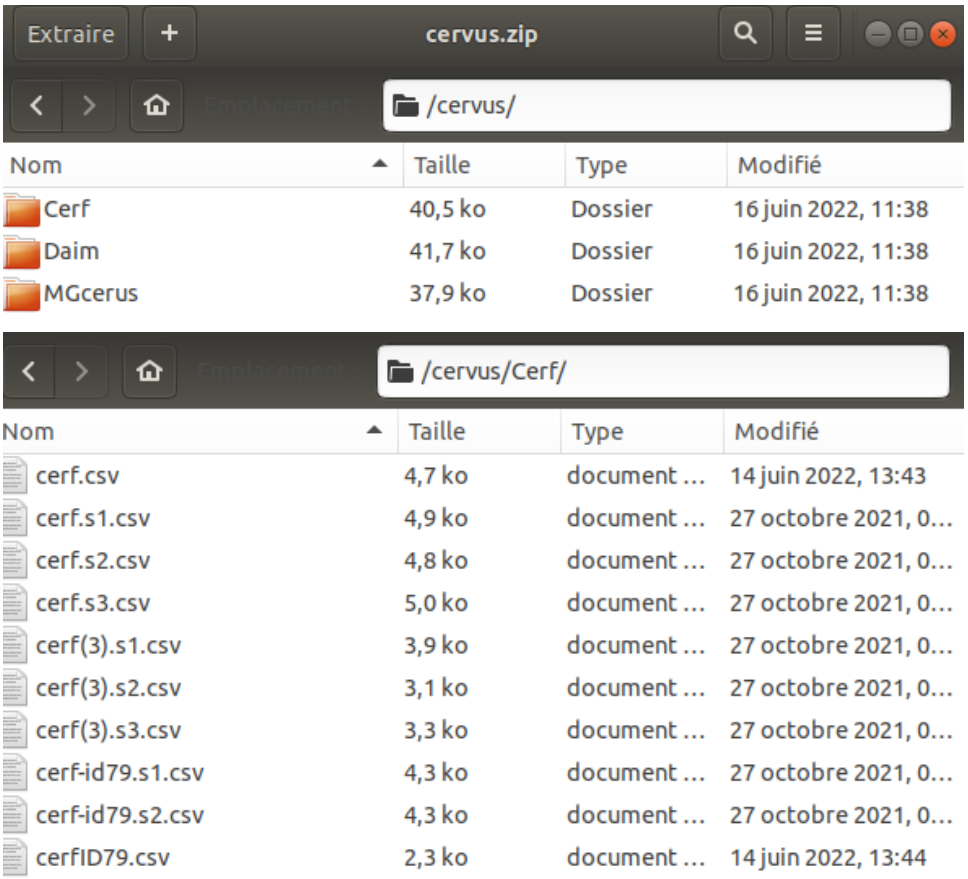


Figure 18 : Exemple de fichier zippé à uploader

Si cette étape est correctement réalisée, Metaboanalyst passe à l'étape suivante et nous propose 2 paramètres de traitements des pics ainsi que plusieurs étapes de filtrages et de normalisations afin d'arriver aux outils proposés précédemment.

Filtrage

Le filtrage a pour but d'éliminer les variables qui sont susceptibles de n'être d'aucune utilité.
Cette étape est recommandée pour les datasets de métabolomique non ciblée (comme les listes de pics).

Les variables non-informatives peuvent être cataloguées en 3 groupes :

- Les variables de très petite valeur (proches de la limite de détection). Elles peuvent être détectées en utilisant la moyenne ou la médiane.
- Les variables qui sont quasi-constantes. Elles peuvent être détectées en utilisant la déviation standard ou l'IQG (intequantile range)
- Les variables de faible répétabilité. Elles peuvent être mesurées en utilisant la déviation standard relative (RSD = SD / moyenne). Les variables avec RSD élevées devraient être supprimées de l'analyse (seuil suggéré : 20 % pour les LC-MS).

Pour les deux premières catégories, MetaboAnalyst utilise ces règles empiriques :

- < 250 variables : 5 % seront filtrées
- Entre 250 et 500 variables : 10 % seront filtrées
- Entre 500 et 1000 variables : 25 % seront filtrées
- > 1000 variables : 40 % seront filtrées

Normalisation

Les procédures de normalisation sont groupées en 3 catégories : sample normalisation, data transformation et data scaling
Sample Normalization

- None
- Sample-specific normalization (i.e., weight, volume)
- Normalization by sum
- Normalization by median
- Normalization by reference sample (PQN)
- Normalization by a pooled sample from group
- Normalization by reference feature Specify
- Quantile normalization

Data transformation

- None
- Log transformation (generalized logarithm transformation or glog)
- Cube root transformation (takes the cube root of data values)

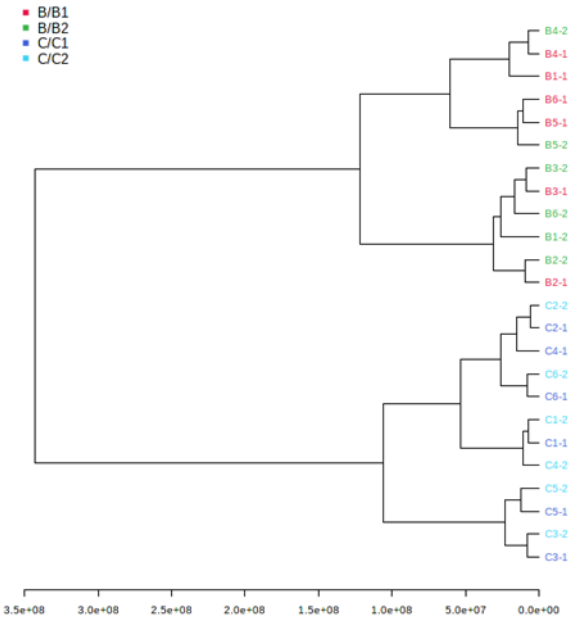
Data scaling

- None
- Mean centering (mean-centered only)
- Auto scaling (mean-centered and divided by the standard deviation of each variable)
- Pareto scaling (mean-centered and divided by the square root of the standard deviation of each variable)
- Range scaling mean-centered and divided by the range of each variable

Exemple de résultat

L'outil dendrogram offre plusieurs options de clusterings :

- mesure de distance: Euclidean / Spearman / Pearson
- algorithme de clustering: ward / average / complete / single



Annexe 2 : Faire fonctionner docker

Pour commencer il faut s'assurer que docker est bien installé, pour ça suivre le tutoriel suivant : https://www.digitalocean.com/community/tutorials/comment-installer-et-utiliser-docker-sur-ubuntu-18-04-fr

Une fois installé on va télécharger le docker suivant :
https://hub.docker.com/r/iguigon/metaboanalystr_feature_table

Pour cela on va rentrer dans un terminal (linux) la ligne suivant :

sudo docker pull iguigon/metaboanalystr_feature_table:v1

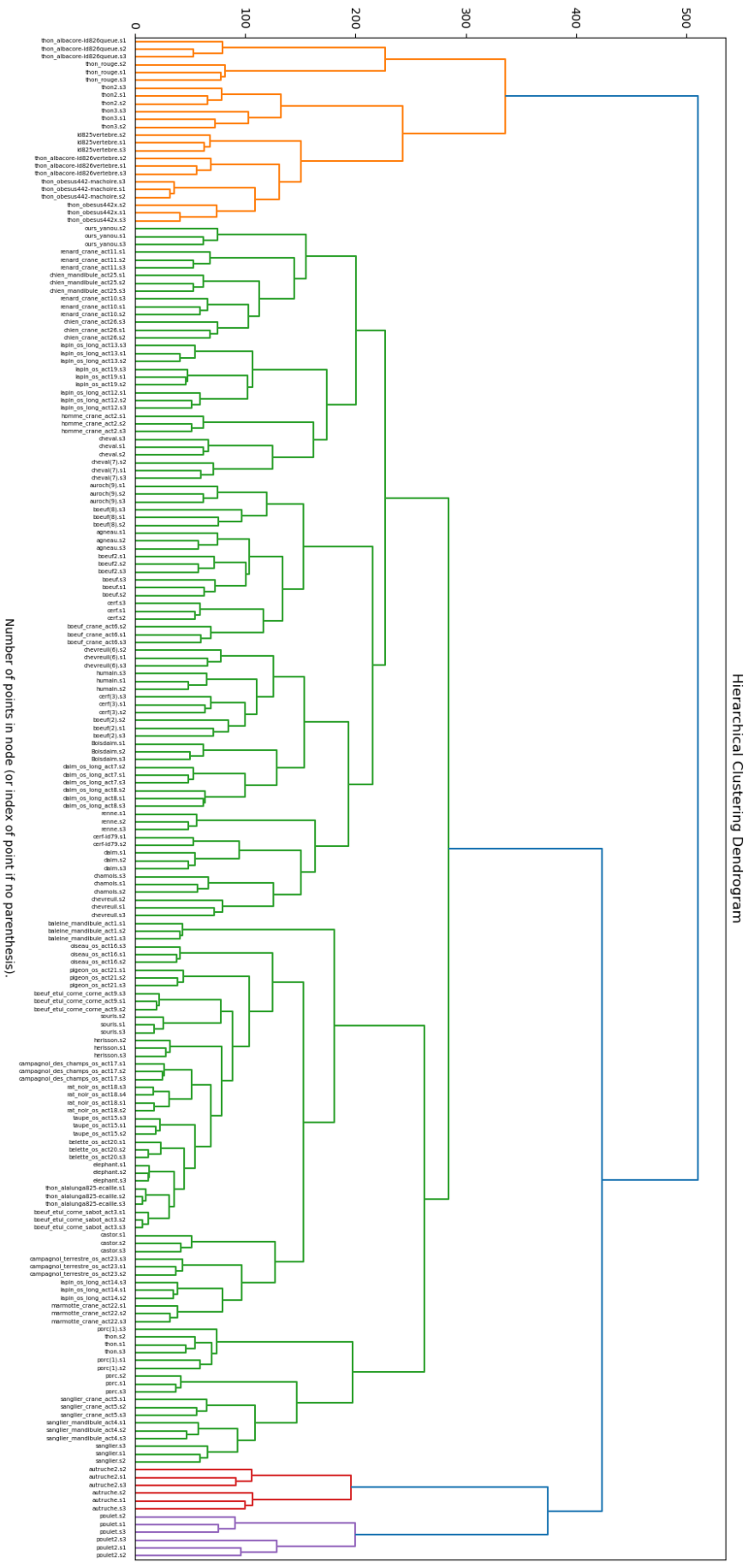
Ensuite afin d'assurer un bon fonctionnement du docker on va venir créer un chemin ou l'on viendra placer les fichiers zip (comme expliqué précédemment) :

/home/orkad/path/to/my/data

On viendra alors placer les données dans le dossier data.
Après normalement tout est bien configuré il suffit d'entrer la ligne suivante :
sudo docker run -v /path/to/my/data:/project myImage --input input.zip --mzwid 0.25 --bw 30 --minfrac 0.5

Il faut savoir que myImage correspond à la version de l'image du docker téléchargé et input.zip le nom du fichier placé dans le chemin créé précédemment, le reste corresponds aux paramètres de la fonction **GroupPeakList** que l'on a pu voir précédemment. Un petit exemple ci-dessous :
sudo docker run -v /home/orkad/path/to/my/data:/project iguigon/metaboanalystr_feature_table:v1 --input bos.zip --mzwid 0.25 --bw 30 --minfrac 0.1Une fois terminé le fichier généré se situera au même emplacement que le fichier zip.

Annexe 3 : Apprentissage non supervisé



Annexe 4 : TPOT

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPClassifier

# NOTE: Make sure that the outcome column is labeled 'target' in the data file
tpot_data = pd.read_csv('PATH/TO/DATA/FILE', sep='COLUMN_SEPARATOR', dtype=np.float64)
features = tpot_data.drop('target', axis=1)
training_features, testing_features, training_target, testing_target = \
            train_test_split(features, tpot_data['target'], random_state=None)

# Average CV score on the training set was: 0.936491935483871
exported_pipeline = MLPClassifier(alpha=0.001, learning_rate_init=0.001)

exported_pipeline.fit(training_features, training_target)
results = exported_pipeline.predict(testing_features)
```

VIII : Bibliographie

Introduction :

- https://www.cristal.univ-lille.fr/spip.php?page=article&id_article=1
- https://orkad.univ-lille.fr/
- https://www.cristal.univ-lille.fr/bonsai/start.html
- https://msap-lab.fr/
- Applications de la spectrométrie de masse type MALDI-TOF à la bactériologie et à la distinction de variants génétiques par Louardi Moussaoui : https://tel.archives-ouvertes.fr/tel-00872251/document

Phase exploratoire :

- https://tice.ac-montpellier.fr/ABCDORGA/Famille/SPECTREDEMASSE.html
- https://www.metaboanalyst.ca/
- *JDN,* 17/01/22, https://www.journaldunet.fr/web-tech/guide-de-l-intelligence-artificielle/1501309-apprentissage-non
- https://scikit-learn.org/stable/
- DataScientest, Machine Learning & Clustering: Focus sur l'algorithme CAH https://datascientest.com/machine-learning-clustering-focus-sur-algorithme-cah
- *JDN,* Antoine Crochet-Damais, 02/06/22, Réduction de dimensionnalité en machine learning : définition, https://www.journaldunet.fr/web-tech/guide-de-l-intelligence-artificielle/1501907-reduction-de-dimensionnalite
- *Wikipédia,* 13/03/22, https://fr.wikipedia.org/wiki/Algorithme_t-SNE
- *https://umap-learn.readthedocs.io/en/latest/*
- *LeMagIT*, George Lawton, 14/10/20, George Lawton https://www.lemagit.fr/conseil/Apprentissage-supervise-et-non-supervise-les-differencier-et-les-combiner
- *Machine Learnia*, Guillaume Saint-Cirgue, https://machinelearnia.com/apprendre-le-machine-learning-en-une-semaine/
- *Inside Machine Learning,* Tom Keldenich, 02/09/21, Recall, Precision, F1 Score – Explication Simple Métrique en ML, https://inside-machinelearning.com/recall-precision-f1-score/
- http://epistasislab.github.io/tpot/

Détermination automatique des espèces à partir d'un prélèvement osseux :

- https://www.docker.com/
- *NIH,* Bradley Worley et Robert Powers, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4990351/
- *Intelligence-artificielle.com*, Vonintsoa, 15/11/21, eXplainable AI : guide complet de la boîte noire de l'IA, https://intelligence-artificielle.com/explainable-ai-guide-complet/