Utilisation des images RMN 2D pour la quantification en métabolomique

Christian Adanmaho

UFR Sciences et Techniques

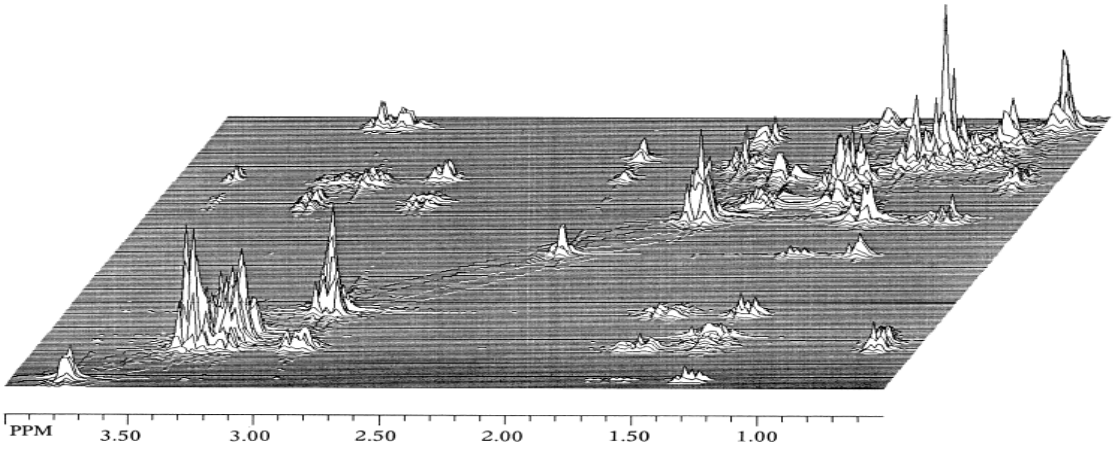
Département de Mathématiques

Master MIGS

*(**M**athématiques pour l'**I**ngénierie, al**G**orithmique, **S**tatistique)*

Mémoire sur le stage de 2e année (M2)

Utilisation des images RMN 2D pour la quantification en métabolomique



Effectué par :
ADANMAHO
Christian

Sous la direction de :
M. Rémi SERVIEN
Mme Gaëlle LEFORT
Mme Marie TREMBLAY-FRANCO

Encadrante universitaire : Mme Catherine LABRUERE

Du 11 Avril 2023 au 10 Octobre 2023
À l'Unité LBE, INRAE de Narbonne
Soutenu le 1er Septembre 2023

Remerciements

Je tiens avant tout à remercier Monsieur Rémi SERVIEN pour son soutien, ses conseils, et surtout pour sa disponibilité. Un encadrant de stage qui s'est impliqué dans la recherche de logement de son stagiaire et qui s'est même rendu disponible pour la visite des appartements, je ne l'oublierai jamais.

Je remercie également Mesdames Marie TREMBLAY-FRANCO et Gaëlle LEFORT, mes co-encadrantes pour leur disponibilité et surtout pour les outils méthodologiques indispensables au déroulement de mes travaux, qu'elles m'ont apportés.

Je tiens à remercier spécialement Madame Catherine LABRUERE, ma tutrice académique pour m'avoir suivi, encouragé et conseillé tout au long de l'année.

L'enseignement de qualité dispensé par le Master MIGS a vraiment nourri mes réflexions. Je remercie donc toute l'équipe pédagogique du Master MIGS.

Un remerciement spécial pour tout le personnel de l'Unité LBE INRAE Narbonne pour m'avoir accueilli, sans oublier les autres stagiaires, doctorants ou en CDD : Valentina, Jérôme, Bastien, Rabeb, Ali, David, Yin-Yan, Emmanuel, Lorenzo, Logan, Korantin, Safiata, ... pour ces discussions au resto routier.

Enfin, un grand merci à ma famille pour avoir toujours été là malgré ces milliers de kilomètres qui nous séparent.

Résumé

La métabolomique est une des disciplines de la grande famille des sciences dites « omiques ». Elle permet d'identifier et de mesurer une fraction importante des métabolites présents dans un échantillon donné. La spectroscopie à résonance magnétique nucléaire (RMN) et la spectrométrie de masse (SM) sont deux techniques couramment utilisées en métabolomique mais nous nous intéressons ici à la spectroscopie RMN qui fournit un spectre de mélange complexe qui correspond à la superposition des spectres des métabolites purs. Après l'identification et la quantification des métabolites présents dans un mélange complexe avec la spectroscopie RMN 1D, on obtient beaucoup de faux positifs qu'on pense pouvoir diminuer en utilisant la spectroscopie RMN 2D. Une des co-encadrantes de ce stage a développé un algorithme appelé BARSA qui permet d'annoter les métabolites présents dans un système biologique avec des spectres RMN 2D. Mais l'identification des pics se fait encore manuellement, ce qui est très fastidieux. L'objectif de ce stage est de combiner un algorithme automatique de détection de pics dans les spectres RMN 2D et l'algorithme BARSA. Nous testons cette procédure de détection des pics et d'identification des métabolites avec différentes séquences de spectres 2D (COSY, TOCSY, HSQC) et aussi leur combinaison puis avec plusieurs seuils sur la probabilité de présence des métabolites afin de trouver la séquence et le seuil qui nous donneront un bon compromis entre la sensibilité et la spécificité.

Mots clés : Métabolomique, résonance magnétique nucléaire (RMN), spectrométrie de masse, spectre.

Abstract

Metabolomics is one of the disciplines in the large family of so-called « omics » sciences. It enables the identification and measurement of a significant fraction of the metabolites present in a given sample. Nuclear magnetic resonance (NMR) spectroscopy and mass spectrometry (MS) are two techniques commonly used in metabolomics, but here we focus on NMR spectroscopy, which provides a complex mixture spectrum corresponding to the superposition of the spectra of pure metabolites. After identifying and quantifying the metabolites present in a complex mixture using 1D NMR spectroscopy, we obtain many false positives, which we believe can be reduced using 2D NMR spectroscopy. One of this internship's co-supervisors has developed an algorithm called BARSA, which can annotate the metabolites present in a biological system with 2D NMR spectra. But peak identification is still done manually, which is very tedious. The aim of this internship is to combine an automatic peak detection algorithm in 2D NMR spectra with the BARSA algorithm. We are testing this peak detection and metabolite identification procedure with different 2D spectral sequences (COSY, TOCSY, HSQC) and also their combination, then with several thresholds on the probability of metabolite presence, in order to find the sequence and threshold that will give us a good compromise between sensitivity and specificity.

Keywords : Metabolomics, nuclear magnetic resonance (NMR), mass spectrometry, spectrum.

Table des matières

Introduction

Dans le quatrième quart du XX^{e} siècle, les sciences « omiques » : la génomique (étude de l'ensemble des gènes), la transcriptomique (étude de l'ensemble des ARN messagers) et la protéomique (étude de l'ensemble des protéines) ont fait leur apparition sur la scène scientifique et ont déjà fait leurs preuves comme outils essentiels pour la caractérisation et la compréhension des mécanismes biologiques (Dormoy et al. 2013). Actuellement, la génomique et la protéomique sont toutes deux coûteuses et nécessitent beaucoup de main-d'œuvre, mais elles constituent potentiellement des outils puissants pour étudier les différents niveaux de la réponse biologique. Cependant, même combinées, elles ne fournissent pas l'ensemble des informations nécessaires à la compréhension de la fonction cellulaire intégrée dans les systèmes vivants, car elles ignorent toutes deux l'état métabolique dynamique de l'organisme entier. Une nouvelle approche appelée **métabolomique** est donc proposée. Elle vise à compléter les informations fournies par la mesure des réponses génétiques et protéomiques (Nicholson et al. 1999, Fiehn 2002).

A part les gènes, les ARN messagers et les protéines, il existe des molécules (acides aminés, sucres, lipides, nucléotides, vitamines, hormones, acides gras, ...) de petite taille, qui sont impliquées dans les processus biochimiques qui se produisent dans les cellules, les tissus ou les fluides biologiques : il s'agit des métabolites. Ces derniers sont les produits finaux des processus de régulation cellulaire. La métabolomique est l'étude des métabolites issus de l'organisme ou provenant de l'environnement. Elle fit son apparition au tout début du troisième millénaire et occupe une place particulière au sein de la famille des « omiques ». Elle se concentre sur l'identification et la quantification des métabolites présents dans un système biologique dans des conditions données et elle se caractérise par la grande diversité des propriétés physicochimiques des molécules.

La métabolomique est devenue de plus en plus populaire de par les récentes avancées technologiques. En recherche biomédicale, elle permet de caractériser les changements métaboliques qui se produisent en réponse à des perturbations externes ou internes telles que les maladies, les changements environnementaux, la nutrition, les médicaments ou les interventions thérapeutiques puis elle offre une vue globale des perturbations biologiques dans le but de découvrir des biomarqueurs qui d'une part permettront une meilleure compréhension des pathologies en identifiant les voies métaboliques impliquées (Dunn et al. 2013) mais qui d'autre part, représentent aussi l'espoir d'un diagnostic précoce des maladies (Boukedimi 2014). Elle permet par exemple, de mieux comprendre certains phénomènes biologiques comme les différences de maturité entre certaines races de porc à la naissance (Lefort 2021), les effets métaboliques de la restriction calorique (connue pour augmenter la longévité) dans divers modèles animaux, l'étiologie de l'épidémie d'obésité et des maladies métaboliques associées (le diabète de type 2 par exemple) (Ellero-Simatos et al. 2019). Elle peut également être utilisée en cancérologie (Oakman et al. 2011), en génétique (Illig et al. 2010), *etc.*..

Pour obtenir des profils métaboliques, la métabolomique utilise couramment deux techniques analytiques : la spectroscopie à résonance magnétique nucléaire (RMN) et la spectrométrie de masse (SM), deux techniques à fort potentiel qui suscitent un vif intérêt dans la communauté scientifique. Ces méthodes mesurent simultanément une large gamme de métabolites à partir d'une seule analyse fournissant des informations structurelles et

quantitatives (ou semi-quantitatives) (Ellero-Simatos et al. 2019). La spectroscopie RMN est la technique analytique utilisée pendant ce stage.

En recherche biomédicale, la recherche des biomarqueurs repose sur l'identification des métabolites, ce qui constitue un des problèmes majeurs en métabolomique. En spectroscopie bidimensionnelle, l'identification des pics se fait encore manuellement, ce qui est très fastidieux. L'objectif de ce stage est donc de rendre automatique cette procédure de détection en développant un algorithme pour l'identification automatique des pics dans les spectres RMN 2D. Ensuite, cette détection sera combinée avec un algorithme (déja existant) d'annotation des métabolites. Les informations extraites des spectres RMN 2D seront enfin combinées avec celles extraites des spectres RMN 1D.

Dans les premiers chapitres de ce mémoire, nous présenterons tout d'abord les spectroscopies RMN unidimensionnelle et bidimensionnelle avec quelques techniques d'identification des pics sur des images métabolomiques. Nous présenterons ensuite une méthode qui combine la spectroscopie RMN 1D et 2D afin de réduire au maximum les faux positifs détectés en 1D. Cette méthode sera testée sur différentes types de données et les résultats obtenues avec ces dernières seront présentés dans le dernier chapitre.

1 Présentation de l'institut

1.1 L'INRAE

L'INRAE (Institut National de Recherche en Agriculture, Alimentation et Environnement), né le 1^{er} janvier 2020, est le premier organisme de recherche spécialisé sur ses trois domaines scientifiques. Il est issu de la fusion entre l'Institut National de la Recherche Agronomique (INRA) et l'Institut national de Recherche en Sciences et Technologies pour l'Environnement et l'Agriculture (IRSTEA). Il a pour objectif de devenir l'un des leaders mondiaux de la recherche pour répondre aux enjeux sociétaux tels que : la sécurité alimentaire et nutritionnelle, la transition des agricultures (agroécologie, réduction de la chimie), la gestion des ressources naturelles et des écosystèmes (eau, sol, forêt), l'érosion de la biodiversité, l'économie circulaire et risques naturels puis aux enjeux plus territorialisés qui incluent les conditions de vie et de rémunération des agriculteurs, la compétitivité économique des entreprises, l'aménagement des territoires, l'accès à une alimentation saine et diversifiée pour chacun.

L'INRAE compte plus de 10 000 agents, 18 centres de recherche localisés dans toute la France, 14 départements scientifiques spécialisés dans des domaines variés tels que la biologie, la génétique, le numérique, *etc.* et 166 projets de recherche européens.

1.2 Le Laboratoire de Biotechnologie de l'Environnement (LBE)

Le laboratoire de biotechnologie de l'environnement est une unité de recherche du centre Occitanie-Montpellier de l'INRAE, située à Narbonne. Il est rattaché aux départements AgroEcoSystem, MICA et TRANSFORM pour la partie scientifique puis au centre INRAE Occitanie-Montpellier pour la partie administrative. Les recherches menées au LBE visent à développer le concept de bioraffinerie environnementale qui consiste à valoriser les résidus, déchets, effluents organiques, *etc.*, issus des activités humaines

ainsi que certaines biomasses en produits d'intérêt industriel (bioénergies, biomolécules, amendement et fertilisant organique) tout en minimisant leur impact environnemental et sanitaire. Cinq objets thématiques (OT) cohabitent au sein du LBE et sont détaillés dans l'organigramme présenté dans la figure 1. Ce stage s'inscrit dans le cadre de l'OT SAMI (Système, Analyse, Modélisation et Informatique).

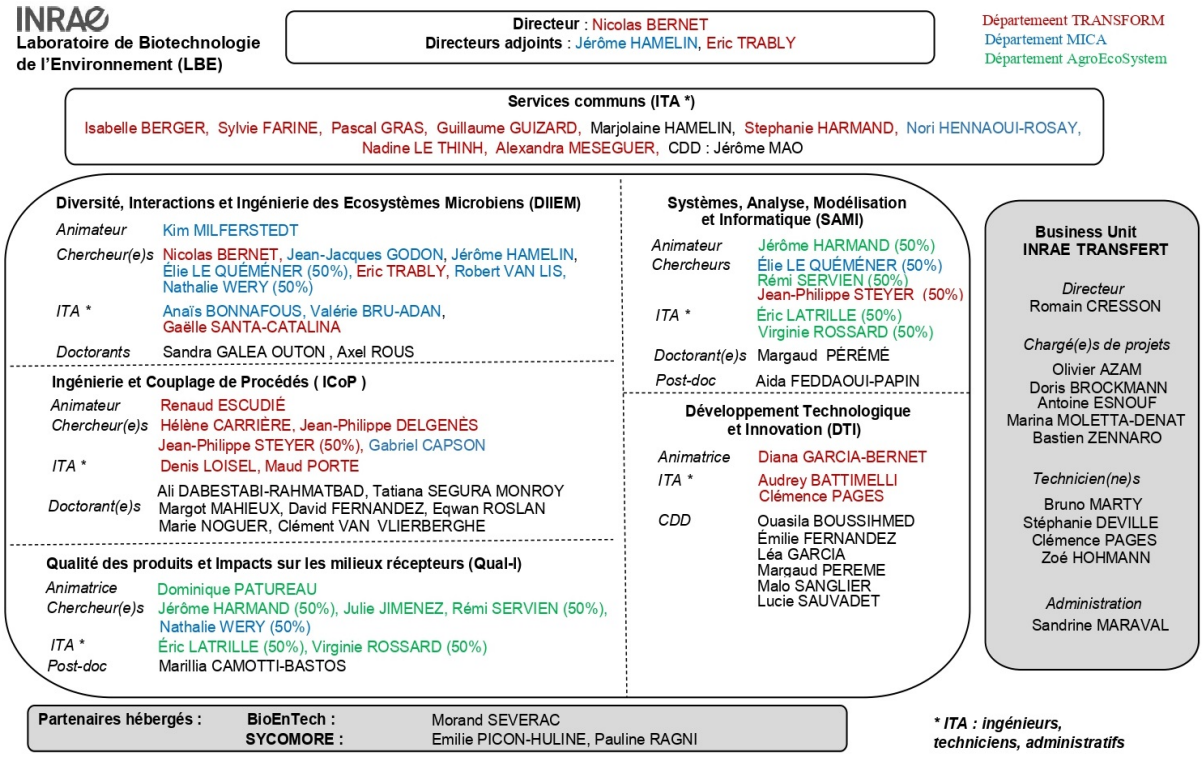


FIGURE 1 – Organigramme du LBE.

1.3 Le projet PANORAMICS

La réintroduction de l'énergie chimique contenue dans les eaux usées industrielles et municipales dans notre économie, au lieu de les traiter et de les jeter à grands frais, constitue un problème majeur pour obtenir une économie circulaire fonctionnelle, proche de zéro pollution et de zéro production de déchets. Les photogranules OPGs (agrégats microbiens compacts, à peu près sphériques, d'un diamètre de plusieurs millimètres) capturent une grande partie de la valeur énergétique et chimique des eaux usées dans leur biomasse, mais les connaissances essentielles sur leur développement et leur cycle de vie font toujours défaut. De plus, les omiques (en particulier la métabolomique) sont les meilleurs outils pour comprendre la photogranulation mais nécessitent encore un effort spécifique de modélisation statistique pour en tirer pleinement parti.

Le projet ANR **PANORAMICS** (Statistical develo**P**ments and **A**pplicatio**N** on **O**xygenic photog**RA**nules for longitudinal **M**ulti-om**ICS** data) coordonné par M. Rémi SERVIEN, vient combler ces lacunes en s'attaquant à ces deux défis : le développement de nouvelles méthodologies pour analyser les ensembles de données longitudinales métabolomiques et multi-omiques et l'approfondissement des connaissances sur le fonctionnement des OPGs afin de les rapprocher des essais industriels pour la dépollution des eaux usées.

2 La spectroscopie RMN unidimensionnelle

La spectroscopie de résonnance magnétique nucléaire (RMN) est une méthode particulièrement pertinente pour l'analyse de mélanges complexes, qu'on peut choisir d'analyser en l'état, avec très peu d'étapes de préparation (Dumez 2022). Via l'obtention de spectres, elle permet d'analyser les composés chimiques associés à un noyau d'intérêt et donne donc accès à une information globale sur l'ensemble des molécules dont la concentration est supérieure à un certain seuil de détection. Elle se base sur la possibilité de certains noyaux atomiques d'interagir avec un champ magnétique. L'idée est que les noyaux de certains atomes possèdent un moment magnétique nucléaire, c'est-à-dire qu'ils se comportent comme des aimants microscopiques. Le noyau le plus couramment utilisé est celui de l'hydrogène : on parle donc de la RMN ^{1}H ou RMN du proton (car le noyau de l'hydrogène ne contient qu'un seul proton) (Lefort 2021). Mais il faut savoir qu'il existe d'autres RMN, telles que celles du carbone-13 ^{13}C, du fluor-19 ^{19}F, de l'azote-15 ^{15}N, du phosphore-31 ^{31}P, *etc.*

En spectroscopie RMN, on mesure la différence d'énergie entre les différents états d'énergie d'un système de spins. Pour ce faire, il faut provoquer des transitions entre ces différents états, c'est-à-dire sortir le système de spins de son équilibre. Le signal temporel qui en résulte est désigné sous le nom d'interférogramme (décroissance libre d'induction), appelé plus communément FID (Free Induction Decay) et qui est constitué d'une superposition de sinusoïdes amorties. Pour plus de détails, voir (Bria et al. 1997). Ces signaux FID acquis ne sont pas directement utilisables, des pré-traitements sont donc nécessaires pour les transformer en spectres afin de les rendre interprétables. Trois principaux traitements sont utilisés : le zero-filling, l'apodisation et la transformée de Fourier. Un exemple de spectre RMN est donné dans la figure 2 ci- après :

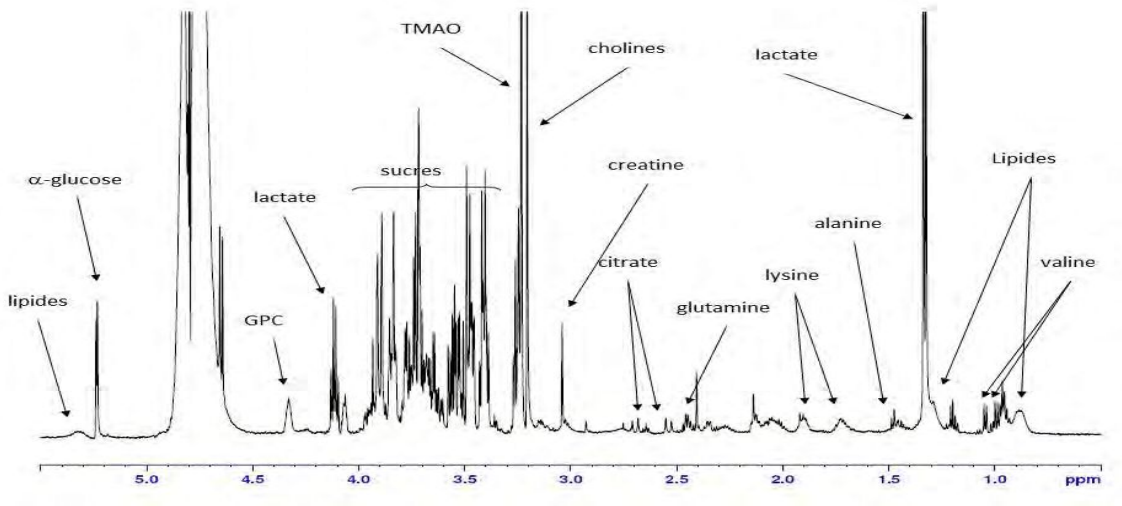


Figure 2 – Exemple de spectre RMN ^{1}H d'un mélange (P. Tardivel 2017).

Ce spectre de mélange (dit complexe) est la somme de tous les spectres des composés présents dans le mélange ou l'échantillon étudié, pondérés par leurs concentrations (Dumez 2022). Il se présente comme une série de signaux (sous la forme de pics), dont l'abscisse est appelée déplacement chimique et est notée δ.

2.1 Pré-traitements des spectres

La reconnaissance automatique des métabolites dans un mélange complexe est rendue délicate par des problèmes comme la déformation du spectre (translation, dilatation ...) ou la superposition des pics (P. J. Tardivel et al. 2017). La plupart de ces problèmes sont dus à la température entre l'acquisition des spectres ou au ph ou lorsque l'expérience est répétée ou porte sur des échantillons biologiques de même nature. Après l'importation des spectres bruts (déjà traités en partie ou non), toute une série d'étapes est nécessaire pour transformer ces spectres en une matrice de données analysables en statistique. Il s'agit de la correction de la ligne de base, l'alignement des pics, la suppression des régions non désirées (celle de l'eau : $[4,5\ ;\ 5,1]$ *ppm* et celle de l'urée : $[5,5\ ;\ 6,5]$ *ppm* dans le cas des échantillons d'urine) et la normalisation des spectres (très importaant pour comparer efficacement les intensités relatives des pics et d'obtenir des résultats plus fiables lors de l'analyse des données de RMN).

Un exemple de ces pré-traitements est présenté dans la figure 3 ci-après.

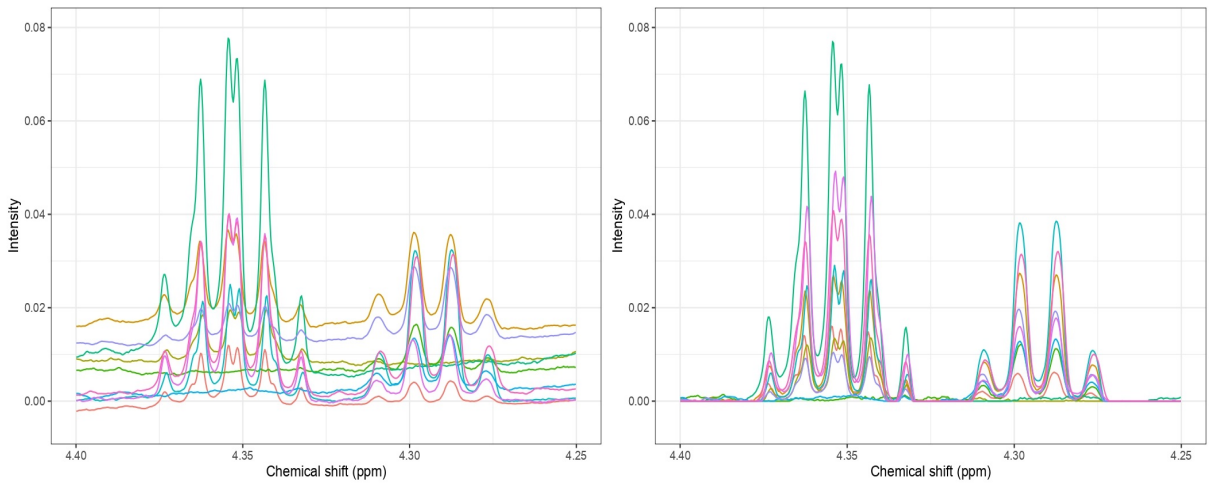


Figure 3 – Spectres RMN de 10 mélanges : A gauche, on a des spectres RMN obtenus à l'état brut après l'application des pré-traitements tels que le zero-filling, l'apodisation et la transformée de Fourier aux signaux FID acquis. A droite, on a les mêmes spectres obtenus après la correction de la ligne de base, l'alignement des pics et la normalisation.

2.2 Identification des composés dans les spectres RMN 1D

Le spectre RMN d'un échantillon étudié peut être considéré comme un objet multiidimensionnel, dont les dimensions pourraient être les concentrations des métabolites individuels mesurables ou plus simplement la distribution de l'intensité spectrale (Nicholson et al. 1999). Le spectre RMN d'une molécule contient une information riche sur sa structure et l'intensité des pics (signaux) renseigne sur sa concentration (Dumez 2022). La forme des pics est définie par le nombre, l'intensité relative et la distance (couplage scalaire J) entre les raies. Ces données traduisent directement le voisinage électronique des noyaux étudiés. En effet, chaque métabolite génère une résonnance spectrale qui lui est propre avec une intensité proportionnelle à sa concentration dans le mélange. La structure chimique de chaque métabolite et, plus précisement, de chaque groupe de protons qui le

compose, va induire une position spécifique des pics. Les couplages entre les groupes voisins (i.e. les liaisons entre les groupes) vont jouer sur la multiplicité d'un groupe de pics (le nombre de pics du groupe) (Lefort 2021).

Un proton ou un groupe de protons équivalents ayant n protons voisins eux-mêmes équivalents (portés par des atomes de carbone voisins) donne par couplage avec ceux-ci un signal possédant $(n+1)$ pics, comme l'indique la figure 4.

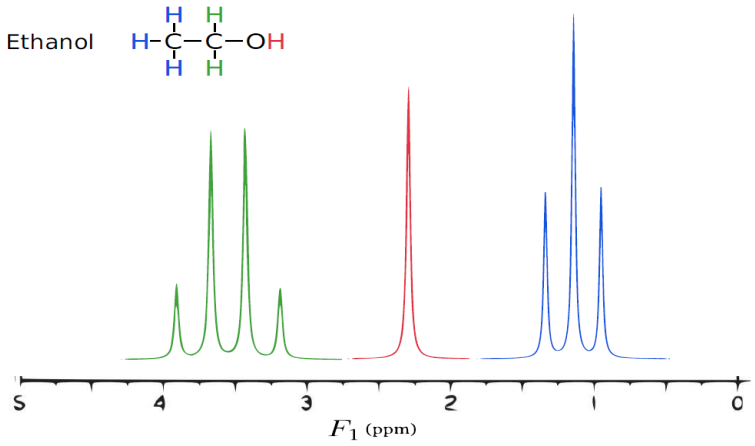


Figure 4 – Spectre RMN 1D de l'éthanol.

Sur ce spectre, on observe trois groupes de protons : le groupe OH est représenté par un singulet (un seul pic), le groupe CH_2 est représenté par un triplet (trois pics) car il a 2 voisins et le groupe CH_3 est représenté par un quadruplet (quatre pics) car il a 3 voisins.

La présence d'un métabolite dans un mélange complexe est déduite de la présence de tous ses pics aux positions théoriques avec les bonnes multiplicités dans le spectre obtenu à partir de ce mélange complexe. La concentration de chaque métabolite dans le mélange est proportionnelle à l'aire sous chacun des pics du métabolite (Lefort 2021).

Il existe déjà plusieurs méthodes pour identifier les pics et quantifier les métabolites présents dans un mélange. La plupart de ces méthodes sont basées sur la déconvolution des mélanges complexes grâce à des spectres de métabolites purs ; un exemple est donné dans la figure 5.

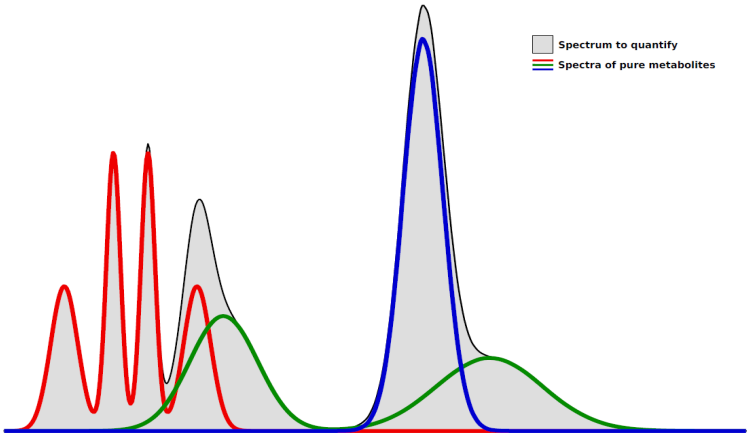


Figure 5 – Déconvolution d'un spectre à l'aide de spectres purs (Lefort 2021).

Désignons par g un spectre de mélange complexe contenant un certain nombre q de métabolites. Ce spectre de mélange complexe peut être défini par une combinaison linéaire des spectres purs de métabolites . On a :

$$g(t) \simeq \sum_{i=1}^{q} \beta_i f_i(t)$$

où $f_i(t)$ correspond au spectre pur du $i^{ème}$ métabolite et β_i à la contribution du métabolite i dans le mélange (Lefort 2021).

Les méthodes les plus couramment utilisées sont : **Autofit** (Mercier et al. 2011) inclus dans le logiciel commercial Chenomx, **BATMAN** (Hao et al. 2012) dans un package R, **BAYESIL** (Ravanbakhsh et al. 2015) disponible à travers une interface Web, **Dolphin** (Gómez et al. 2014) implémenté dans le package rDolphin et **ASICS** (P. J. Tardivel et al. 2017), un package R pour l'identification et la quantification automatique des métabolites sur des spectres RMN ^1H de mélange complexe (Lefort et al. 2019) qui se montre supérieur aux méthodes précédentes en termes de quantifications et d'identification. C'est d'ailleurs cette dernière qui a été utilisée tout au long de ce stage.

Une librairie de spectres de métabolites purs est requise pour l'identification des composés. On procède à la suppression de tous les spectres dont les pics ne sont pas présents dans le spectre du mélange complexe (dit autrement : un spectre de la librairie est sélectionné si tous ses pics sont présents dans le spectre du mélange) et on définit un seuil en dessous duquel certains pics seront ignorés lors de l'identification.

Pour la quantification des métabolites (avec ASICS), le spectre du mélange complexe est défini par une combinaison linéaire des spectres de la librairie de référence :

$$g(t) = \sum_{i=1}^{q} \beta_i f_i\left(\phi_i(t)\right) + \epsilon(t) \quad avec \quad \beta_i \geq 0$$

où ϕ est une fonction de déformation (translation et distorsion) ; les $f_i \circ \phi_i$ correspondent aux spectres prétraités et présélectionnés de la librairie de référence ; $\beta = (\beta_1, \beta_2, \cdots, \beta_q)$: les coefficients associés à ces spectres (ou aux métabolites correspondants) et ϵ le bruit. Une étape de sélection basée sur le contrôle du Family Wise Error Rate (FWER) au risque α est mise en place pour obtenir un $\hat{\beta}$ parcimonieux. Les quantifications $\hat{\beta}_i$ des métabolites sélectionnés sont ré-estimées en restreignant le modèle linéaire précédent au $\hat{\beta}$ obtenu et on divise chaque quantification $\hat{\beta}_i$ par le nombre de protons du métabolite sélectionné correspondant (P. J. Tardivel et al. 2017).

Après la quantification des métabolites, des analyses statistiques sont faites pour mettre en évidence les métabolites ayant les pouvoirs discriminants les plus élevés. Ces analyses permettent d'évaluer l'effet d'un facteur d'intérêt (une maladie par exemple) sur les métabolites identifiés. En effet, les biomarqueurs recherchés proviennent des variables qui sont discriminantes dans les analyses statistiques. Leur découverte se fait par des analyses statistiques multivariées telles que l'ACP (Analyse en Composantes Principales) ou l'OPLS-DA (Orthogonal Projections to Latent Structures Discriminant Analysis).

Toutes ces analyses statistiques sont disponibles dans le package R ASICS. D'après les résulats de Lefort 2021, ASICS est très performant pour non seulement l'identification et la quantification des métabolites (même ceux qui n'ont pas pu être identifiés par des experts), mais aussi pour l'extraction des métabolites les plus pertinents liés à une condi-

tion d'intérêt (une maladie par exemple). Il rencontre quand même quelques difficultés pour identifier les métabolites avec de faibles concentrations et également ceux dont les pics sont dans une région à forte densité de pics. En plus des métabolites non identifiés, il y en a qui sont encore identifiés à tort : les faux positifs. La RMN 1D peut donc avoir une sensibilité relativement faible. Ces problèmes dévoilent les limites de la spectroscopie RMN unidimensionnelle.

2.3 Quelques limites de la spectroscopie RMN 1D

Bien que la spectroscopie RMN 1D fournisse de précieuses informations sur les composés qu'elle peut détecter, son champ d'applications est limité par des seuils de détection peu favorables. Ainsi l'expérience de la RMN ^{1}H 1D permet de détecter en quelques minutes des molécules de concentrations supérieures à 10 micromoles par litre. Dans un mélange complexe, de nombreux composés peuvent avoir une concentration très inférieure à cette valeur (Dumez 2022) et ne pourront donc pas être détectés. De plus, en RMN 1D, la résolution est souvent limitée, en particulier lorsque les pics des différents noyaux de la molécule se chevauchent. Cela peut rendre difficile l'identification et la séparation des pics individuels des différents noyaux, en particulier dans des échantillons complexes. Il faut noter également que la RMN 1D est limitée à la détection des noyaux ayant des moments magnétiques non nuls. Par exemple, la RMN 1D ne peut pas détecter directement les noyaux d'hydrogène liés au carbone ^{12}C ou à l'oxygène ^{16}O (protons d'eau) car ces deux noyaux possèdent un spin nucléaire égal à 0.

Dans l'éthanol dont le spectre RMN 1D est représenté en figure 4, on observe trois signaux distincts qui peuvent être attribués respectivement aux protons des groupements CH_2, OH et CH_3. Il est possible d'obtenir une information encore plus précise sur la structure de cette molécule, telle que la proximité entre les différents protons : le groupement CH_2 est ici lié au groupe méthyle CH_3 mais sépare ce dernier du groupemment OH. Pour atteindre ce niveau de détail de l'architecture moléculaire, il faut néanmoins recourir à une analyse des couplages scalaires de chaque signal du spectre. Malheureusement, la spectroscopie RMN 1D, pourtant fondatrice de la spectroscopie RMN moderne, peut s'avérer insuffisante de par la complexité des spectres de molécules dont la taille est telle que de trop nombreux signaux viennent se superposer (Courtieu et al. 2012).

Pour pallier ces problèmes, des techniques avancées telles que la RMN multidimensionnelle en particulier la RMN 2D, sont souvent utilisées pour obtenir une meilleure résolution, une sensibilité accrue et des informations plus détaillées sur les molécules.

3 La spectroscopie RMN bidimensionnelle

La spectroscopie RMN 2D ou bidimensionnelle a été proposée par le chimiste et physicien belge Jean JEENER en 1971. C'est un ensemble de dispositifs de reconnaissance de relations de proximité, dans l'espace ou à travers les liaisons, entre plusieurs noyaux actifs en RMN. Il s'agit de la RMN de corrélation (wikipedia). Elle est une extension puissante de la RMN 1D qui offre des informations supplémentaires et spécifiques sur la structure tridimensionnelle, les interactions atomiques et les contacts spatiaux dans une molécule, ce qui permet une analyse approfondie des composés chimiques.

En effet, la RMN 1D donne la position et l'intensité des pics de résonance ainsi que les couplages à courtes distances mais la RMN 2D permet de montrer les corrélations plus complexes existant entre les spins, ce qui est très utile lorsque les molécules sont complexes (comportant beaucoup d'atomes couplés entre eux, couplage à longue portée entre spins ou à travers l'espace , ...). Elle apporte donc plus d'informations que la RMN 1D, notamment pour identifier les métabolites présents dans des échantillons complexes. Elle permet également d'obtenir une résolution améliorée par rapport à la RMN 1D.

Généralement, dans une expérience de spectroscopie RMN bidimensionnelle, on obtient un spectre en trois dimensions : le déplacement chimique pour le noyau 1 (δ_1), le déplacement chimique pour le noyau 2 (δ_2) et l'intensité du signal. La représentation tridimensionnelle n'étant pas pratique pour l'exploitation des spectres, on fait une projection sur un plan, où l'information d'intensité du pic est donnée sous forme de courbes de niveaux (contour) (Cours Univ. Paris Diderot) comme l'indique la figure 6 ci-dessous. Cette représentation facilite la séparation et l'identification des signaux de différents noyaux qui pourraient se chevaucher dans un spectre 1D.

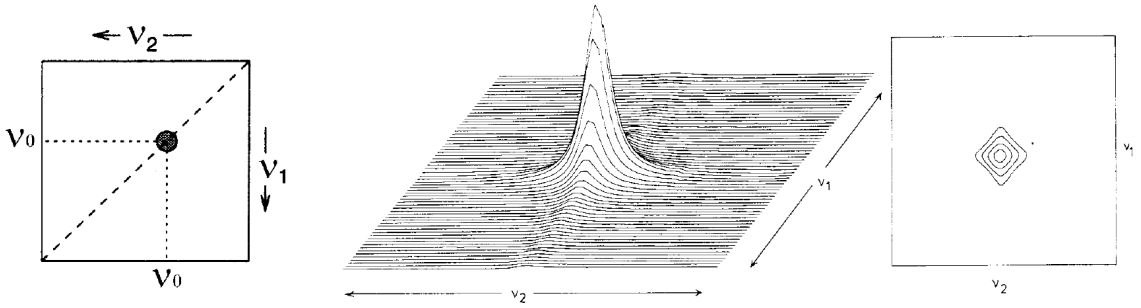


FIGURE 6 – Représentations tridimensionnelle et bidimensionnelle d'un spectre 2D.

Par contre, la RMN 2D est beaucoup plus longue en termes de temps d'acquisition de spectres de haute qualité ; ce qui limite ainsi le nombre d'échantillons pouvant être analysés pendant un créneau donné. C'est donc compliqué de la faire à chaque fois pour chaque échantillon mais si on a une centaine d'échantillons, on peut avoir quelques spectres 2D de pools d'échantillons qui peuvent aider à l'identification.

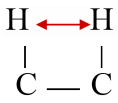
En spectroscopie RMN 2D, il existe plusieurs types (séquences) de spectres qui permettent d'obtenir des informations détaillées sur les interactions entre les noyaux atomiques dans une molécule. Ces séquences permettent de trouver les corrélations entre les déplacements chimiques des différents spins qui sont couplés entre eux. On peut ainsi obtenir des spectres de corrélation homonucléaire $^1H - ^1H$ ou hétéronucléaire $^1H - ^{13}C$.

3.1 Séquences RMN 2D les plus couramment utilisées

3.1.1 COSY (COrrelation SpectroscopY)

La plus simple des expériences de spectroscopie RMN bidimensionnelle est la corrélation homonucléaire appelée COSY. C'est la séquence de RMN 2D la plus utilisée, incontournable pour l'attribution des signaux d'un spectre RMN 1D complexe. Elle permet de mettre en évidence les protons qui sont couplés de façon scalaire dans une molécule.

Ceci permet de déterminer les connexions et les liaisons $^1H - ^1H$ entre les atomes d'hydrogène. L'exemple du proprionate d'éthyle est donné dans la figure 7 ci-dessous.



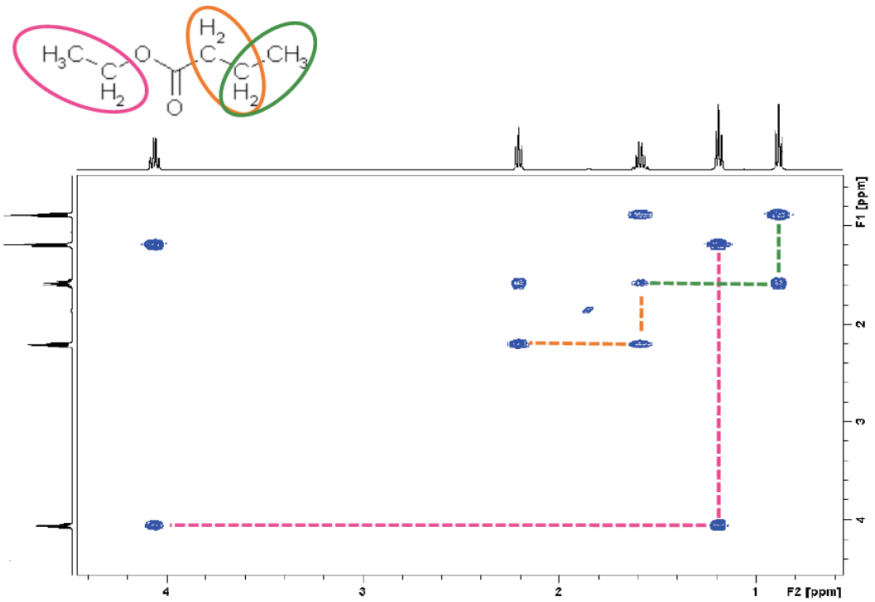
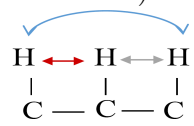


FIGURE 7 – Spectre COSY du proprionate d'éthyle.

Sur un spectre COSY, on observe une diagonale qui correspond au spectre 1D du mélange et des taches hors diagonale qui traduisent les corrélations $^1H - ^1H$. Ces taches sont symétriques par rapport à la diagonale.

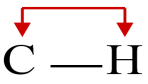
3.1.2 TOCSY (TOtal Correlation SpectroscopY)

La spectroscopie bidimensionnelle TOCSY est une bonne alternative à la séquence COSY pour les macromolécules (ou molécules polymères). Elle permet de détecter les corrélations entre tous les protons (en couplant les spins adjacents et ceux distants) dans une molécule. Elle fournit une image complète des interactions entre les protons, ce qui est peut être utile pour la détermination de la structure et l'attribution des signaux.



3.1.3 HSQC (Heteronuclear Single Quantum Coherence)

La spectroscopie bidimensionnelle HSQC permet de détecter les changements dans les signaux de RMN pendant une réaction chimique en révélant des couplages entre spins de natures différentes à travers une liaison. Elle assure la corrélation des signaux des noyaux de carbone ^{13}C (ou d'autres noyaux hétéronucléaires tels que le fluor-19 ^{19}F, l'azote-15 ^{15}N, ou le phosphore-31 ^{31}P) avec les protons ^1H correspondants (L'exemple du proprionate d'éthyle est donné dans la figure 8). Elle est souvent utilisée pour attribuer les signaux de carbone dans une molécule organique. Elle fournit des informations sur la connectivité entre les atomes et des indices sur la structure des molécules.



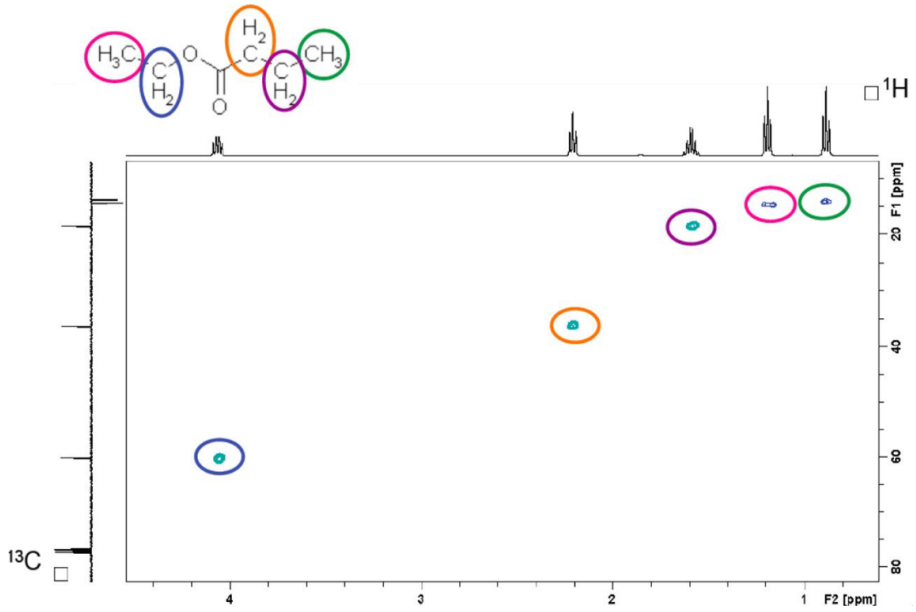


FIGURE 8 – Spectre HSQC du proprionate d'éthyle.

Dans un spectre HSQC, les signaux des atomes d'hydrogène sont représentés sur l'axe horizontal (l'axe des abscisses) et ceux des noyaux hétéronucléaires sont représentés sur l'axe vertical (l'axe des ordonnées). Chaque pic dans le spectre correspond à une paire d'atomes liés entre eux via une seule liaison. L'HSQC sert également de base à des séquences beaucoup plus complexes utilisées pour des expériences de RMN 3D.

Il existe d'autres séquences 2D telles que la J-RESolved (JRES), la NOESY (Nuclear Overhauser Effect SpectroscopY), l'HMBC (Heteronuclear Multiple Bond Coherence), *etc.*, mais qui n'ont pas été utilisées pendant ce stage.

Ces différents types de spectres RMN 2D sont souvent utilisés en combinaison avec d'autres techniques d'analyse spectrale pour caractériser et identifier les composés chimiques présents dans un système biologique. La principale limitation de la RMN 2D à l'identification des métabolites se situe dans l'usage des techniques de détection automatique des pics.

3.2 Détection des taches sur des spectres RMN 2D

En spectroscopie bidimensionnelle, les spectres RMN 2D sont généralement représentés sous forme de matrices de données qui contiennent les intensités relatives des pics. Juste après l'importation d'un spectre RMN 2D, il est d'abord crucial de réduire le bruit tout en préservant les caractéristiques et les informations importantes du spectre. Pour cela, il faut définir un seuil en dessous duquel toutes les valeurs (intensités) seront remplacées par zéro ou réduites à une valeur de bruit de référence. Il faut noter que le choix de ce seuil dépend de plusieurs facteurs tels que : la qualité du spectre, le niveau de bruit et les caractéristiques spécifiques du spectre. Il peut donc avoir un impact sur la qualité des résultats ; dit autrement, un seuil trop élevé peut entraîner une suppression excessive du signal d'intérêt, tandis qu'un seuil trop bas peut ne pas éliminer suffisamment de bruit.

Il existe plusieurs approches (visuelle, statistique ou adaptative) pour calculer ce seuil. Mais, seulement trois ont été testées pendant mes travaux :

- la première consiste à prendre le maximum des intensités de pics situés dans une région dans laquelle on n'est pas censé trouver des pics.
- la deuxième consiste à calculer l'écart type des intensités du spectre et à multiplier celui-ci par un facteur multiplicatif $(2, 3, \ldots)$ pour définir le seuil de bruit. Ce facteur dépend des spécificités du spectre.
- La troisième est une méthode très robuste basée sur l'écart interquartile (**IQR** Interquartile Range), une mesure statistique couramment utilisée pour évaluer la dispersion des données. Elle est utilisée lorsque les données du spectre suivent approximativement une distribution normale. L'IQR est divisé par 1.348 (inverse de la fonction de répartition cumulative d'une distribution normale standardisée à 90%) avant d'être mulitiplié par un facteur pour obtenir le seuil.

Une fois la suppression de bruit effectuée, on peut passer à la détection des pics sur les spectres. Ceci peut être fait avec l'usage de plusieurs techniques telles que la recherche des maxima locaux, la persistance topologique ou le clustering.

3.2.1 La recherche des maxima locaux

La détection des pics par la méthode des maxima locaux (Définition 1) est une technique couramment utilisée dans de nombreux domaines tels que le traitement du signal, la chimie analytique, la spectroscopie et d'autres domaines où l'identification et la localisation des pics sont essentielles pour l'analyse et l'interprétation des données.

Définition 1. *Soit $f : U \longrightarrow \mathbb{R}$ une fonction de deux variables, où U est un ouvert de $\mathbb{R}^2$. On dit que f admet un **maximum local** en $(x_0, y_0) \in U$ s'il existe un disque ouvert $D \subset U$, centré en (x_0, y_0) tel que : $\forall (x, y) \in D, \quad f(x, y) \leq f(x_0, y_0)$.*

Cette méthode repose sur la recherche des points où la valeur (ou l'intensité) du signal est plus élevée que celle de ses voisins immédiats. Ces points correspondent aux maxima locaux et peuvent être considérés comme des pics. Pour chaque maximum local détecté, on enregistre sa position (abscisse et ordonnée) ainsi que sa valeur de l'intensité du pic détecté.

Le problème avec cette méthode est que les signaux sont souvent mesurés avec du bruit, ce qui provoque de nombreux maxima locaux qui ne sont pas pertinents pour notre analyse. Le maximum global ainsi que certains maxima locaux peuvent ne pas correspondre à ce que nous recherchons.

3.2.2 La persistance topologique

L'analyse de données topologiques (en anglais : Topological Data Analysis **TDA**) est un domaine récent et en pleine croissance fournissant des méthodes mathématiques, statistiques et algorithmiques bien fondées pour déduire, analyser et exploiter les structures géométriques et topologiques non évidents dans un ensemble des données éventuellement complexe (Chazal et al. 2021). Cette méthode s'est avérée très utile dans plusieurs contextes, surtout quand on travaille en dimensions élevées, et avec une forte présence de

bruit. L'idée est que les données possèdent souvent une certaine forme intrinsèque telle que la forme d'un nuage de points, la forme d'un signal ou la forme d'un objet géométrique (Huber 2021). L'outil le plus important en TDA est **l'homologie persistante** qui, en spectroscopie RMN permet non seulement de détecter les pics sur des spectres, mais aussi de quantifier la **significativité** (la dominance ou l'importance) de ces pics d'une manière naturelle, ce qui nous permet donc de sélectionner les pics qui sont pertinents pour notre analyse.

Avec l'homologie persistante, on s'intéresse à la durée de vie des classes d'homologie de chaque dimension au cours d'une filtration donnée (pour plus de détails, voir Callet 2019 et Mottet 2011). En spectroscopie bidimensionnelle, comme on est en dimension 2, les classes d'homologie sont uniquement de dimension 0 c'est-à-dire des composantes connexes ou de dimension 1 (des cycles). Mais pour la détection des pics, seule la dimension 0 est à regarder.

Considérons par exemple la fonction $f : \mathbb{R}^2 \longrightarrow \mathbb{R}$ dont le graphe est un spectre 2D. L'idée de la persistance consiste à déterminer les composantes connexes des sur-niveaux $(f \geq t)$ lorsque l'on fait décroitre le paramètre $t \in \mathbb{R}$ entre le maximum global de f et 0 et à reporter ensuite sur un diagramme les valeurs de t pour lesquelles on observe l'apparition ou la disparition d'une composante. Une composante connexe qui apparaît pour une valeur i de t et qui disparaît pour une valeur j de t aura pour persistance le réel $j - i$ et sera représentée sur un diagramme appelé **diagramme de persistance** comme l'indique la deuxième sous-figure de la figure 9 ci-dessous.

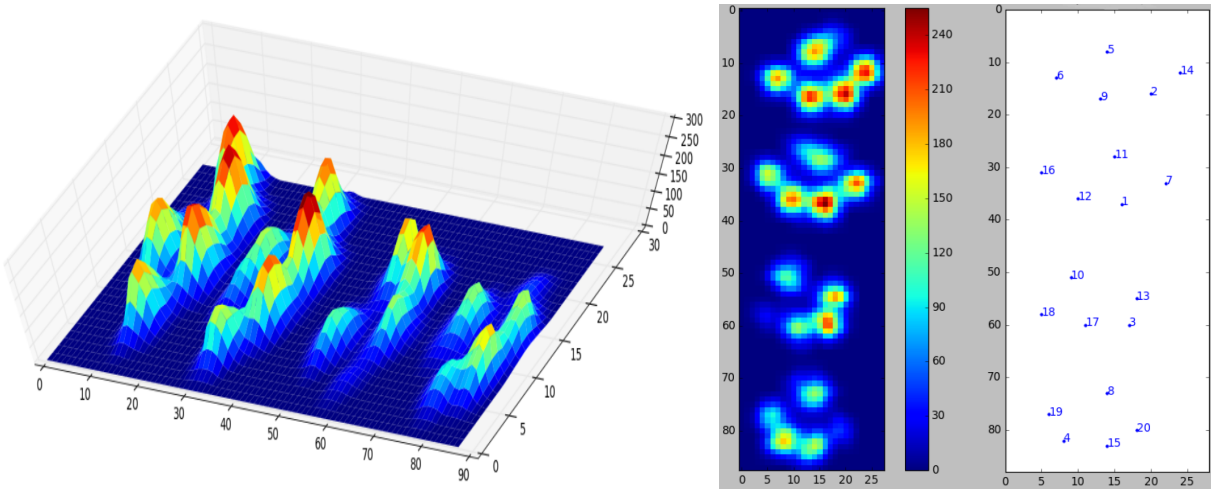


Figure 9 – Diagramme de persistance.

En spectroscopie bidimensionnelle, la persistance en dimension 0 peut être vue comme la durée de vie d'une île. Considérons par exemple un niveau d'eau qui descend continuellement vers des niveaux inférieurs. Aux maxima locaux, des îles apparaissent (naissance) et à chaque minimum local, deux îles fusionnent et on considère que l'île inférieure est fusionnée avec l'île supérieure (la mort). Les codes barres commencent aux naissances et se développent vers le bas jusqu'au niveau où l'île correspondante fusionne avec une autre île. Seule l'île soulevée par le maximum global ne meurt jamais. La persistance d'une île est alors la différence entre le niveau des naissances et le niveau des décès. De plus les îles sont étiquetées par persistance décroissante.

La méthode de persistance topologique a été implémentée sous python. Etant donné que les travaux de ce stage sont réalisés sous le logiciel R, nous avons donc utilisé le package R **reticulate** qui est une bibliothèque du langage de programmation R qui permet l'intégration des scripts Python dans des projets R existants ou l'appel des fonctions Python spécifiques à partir de R. Cela nous a donc permis d'exécuter le script Python de la persistance topologique.

3.2.3 Le clustering

Le clustering est une technique de classification (supervisée ou non) qui partitionne un ensemble de données en un nombre restreint de groupes (clusters) les plus homogènes possibles, c'est à dire les groupes dont les individus sont semblables entre eux à l'intérieur d'un groupe et différents d'un groupe à l'autre. Avoir un bon regroupement revient donc à minimiser la similarité inter-classe (distance entre les individus d'un même groupe) et à maximiser la similarité intra-classe (distance entre les individus de groupes différents).

La détection des pics peut être réalisée à l'aide du clustering en utilisant différentes approches qui fournissent des clusters avec des formes arbitraires.

i Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

C'est l'un des algorithmes de clustering les plus utilisés dans la communauté scientifique d'aujourd'hui. Il se base sur une fonction de densité ou de connectivité. Il estime la densité (Définition 2) autour de chaque point en utilisant le concept de ϵ-voisinage (Hahsler et al. 2019).

Définition 2. *Soit $\mathcal{D} = \{x_i \in \mathbb{R}^n\}_{i=1}^N$, un ensemble de N points décrits par n attributs et $\epsilon \in \mathbb{R}_+$. Le ϵ-voisinage $N_\epsilon(p)$ d'un point $p \in \mathcal{D}$ est l'ensemble des points situés dans un rayon de ϵ autour de p.*

$$N_\epsilon(p) = \{q \in \mathcal{D} \mid d(p,q) < \epsilon\}$$

où d est une distance.

*La **densité** autour d'un point $p \in \mathcal{D}$ peut être estimée par $|N_\epsilon(p)|$, le cardinal de l'ϵ-voisinage de ce point.*

DBSCAN utilise non seulement la notion de ϵ-voisinage mais aussi un seuil appelé **minPts** qui représente le nombre minimal de points dans le ϵ-voisinage d'un point. L'idée qui le sous-tend est que : étant donné un ensemble de points, des points sont assignés à un même cluster s'ils sont atteignables (voir (Hahsler et al. 2019) pour plus de détails). L'algorithme DBSCAN identifie tous les groupes en trouvant systématiquement tous les points centraux et en étendant chacun d'eux à tous les points atteignables par la densité. L'algorithme commence par un point arbitraire p et récupère son ϵ-voisinage. S'il s'agit d'un point central, un nouveau groupe est créé. Ce groupe est étendu en lui assignant tous les points de le ϵ-voisinage de p. Si un autre point central supplémentaire est trouvé dans ϵ-voisinage de p, la recherche est étendue à tous les points de son ϵ-voisinage. Si aucun autre point central n'est trouvé dans le voisinage élargi, alors le groupe est complet et l'algorithme visite les points restants pour voir si un autre point central peut être trouvé pour commencer un nouveau groupe. Après avoir traité tous les points, les points qui n'ont pas été affectés à un groupe sont considérés comme du bruit.

ii K-means (clustering par partitionnement)

C'est un algorithme de clustering basé sur le partitionnement qui vise à regrouper les données en K clusters, où K est un nombre prédéfini. Etant donné un ensemble $\mathcal{D} = \{x_i \in \mathbb{R}^n\}_{i=1}^N$ de N points décrits par n attributs, il produit une partition de $\mathcal{D}$ en $K < N$ clusters et cette partition doit optimiser la similarité intra-classe. L'algorithme se déroule selon les étapes suivantes :

i. sélectionner K points initiaux représentant les centres de gravité des K clusters,

ii. affecter chaque point au cluster dont le centre est plus proche,

iii. recalculer le centre de gravité de chaque cluster,

iv. répéter les étapes *ii.* et *iii.* jusqu'à ce que les centres se stabilisent, autrement dit lorsque l'inertie intra classe cesse de décroitre.

L'algorithme des K-means est rapide et permet de traiter rapidement de grands ensembles d'individus, mais suppose que le nombre K de classes est fixé à priori. De plus les partitions obtenues dépendent des K centres choisis à l'étape initiale de l'algorithme. En utilisant les K-means, on peut identifier des clusters compacts dans les données, qui peuvent correspondre à des pics sur des spectres RMN 2D.

iii Classification Ascendante Hiérarchique (CAH)

Contrairement au K-means, les méthodes hiérarchiques ascendantes utilisent une matrice de distances, ne nécessitent pas de spécifier le nombre de clusters et produisent des suites de partitions en classes de plus en plus vastes. Leur fonctionnement est le suivant : à l'étape initiale, chaque point forme un cluster, on calcule la matrice de distance entre chaque couple de cluster ; à chaque étape, on recherche les deux classes les plus proches, on les fusionne et on ré-itère le processus jusqu'à ce qu'il n'y ait plus qu'une seule classe. Les fusions hiérarchiques et successives des clusters les plus proches sont représentées sous la forme d'un arbre appelé dendogramme. De cette façon, il est possible de voir plus facilement le bon nonbre de clusters à retenir, ce qui permet d'obtenir un clustering en coupant le dendogramme à un niveau choisi. Ces méthodes créent donc une hiérarchie des clusters : les points les plus similaires sont regroupés dans les clusters aux plus bas niveaux et ceux moins similaires sont regroupés au plus haut niveaux.

Le problème avec ces méthodes est que les notions de centre de gravité et d'inertie n'ont plus de sens si la distance entre individus n'est pas euclidienne. Il faut donc considérer d'autres stratégies d'agrégation des classes telles que :

i. **Le critère du saut minimum (single linkage)** : la distance entre deux groupes A et B est la plus petite distance entre éléments des deux groupes. Il est défini par $d(A, B) = \min_{x_i \in A, x_j \in B} d(x_i, x_j)$. Ce critère favorise le regroupement des classes qui possèdent des individus proches.

ii. **Le critère du diamètre (complete linkage)** : la distance entre deux groupes A et B est la plus grande distance entre éléments des deux groupes. Il est défini par $d(A, B) = \max_{x_i \in A, x_j \in B} d(x_i, x_j)$. Il exige que les points les plus éloignés soient proches.

iii. **Le critère de la moyenne (average linkage)** : il offre un compromis entre les deux critères précédents. Il est défini par : $d(A, B) = \frac{1}{n_A n_B} \sum_{x_i \in A} \sum_{x_j \in B} d(x_i, x_j)$.

iv. **Le critère de Ward** : Pour passer d'une partition en $k+1$ classes à une partition en k classes, il fusionne les deux classes A et B pour lesquelles la perte d'inertie inter-classe : $\delta(A,B) = \frac{n_A n_B}{n_A + n_B}\|\mu_A - \mu_B\|^2$ est la plus faible (où μ_A et μ_B sont les les centres de gravité des deux classes).

En général, si les données montrent une structure claire, avec des clusters bien séparés les uns des autres et suffisamment compacts, les quatres méthodes produisent des résultats similaires.

3.3 Annotation des métabolites avec l'algorithme BARSA

L'algorithme BARSA (**B**idimension**A**l nm**R** **S**pectra **A**nnotation) est un algorithme semi-automatique (unpublished) développé sur *R* pour l'annotation des métabolites présents dans une matrice biologique complexe. Cet algorithme prend en argument la liste des couples *ppm* (peak-list) correspondant aux positions des pics détectés, puis la base de données des peak-lists des composés de référence. En paramètres, il lui faut le nom de la séquence 2D à utiliser (JRES, COSY, TOCSY, HSQC et/ou HMBC), la tolérance sur les déplacements chimiques, un seuil (sur la probabilité de présence) et une condition d'unicité.

L'algorithme parcourt la base de données des composés de référence et pour chaque composé, il compare toutes les paires de pics du composé à la liste de paires de pics de la matrice biologique. Un score de présence est donc calculé pour chaque composé de référence :

$$Score\ de\ présence = \frac{Nombre\ de\ pics\ trouvés}{Nombre\ théorique\ de\ pics\ du\ composé\ de\ référence}.$$

Ensuite l'algorithme procède à la suppression de tous les métabolites dont le score de présence est inférieur au seuil. Pour aller plus loin, lorsque la condition d'unicité est requise, l'algorithme supprime toutes les paires de pics assignées à plusieurs métabolites. L'algorithme retourne à la fin la liste des métabolites présents dans le système biologique étudié avec leurs scores de présence.

3.4 Objectifs du stage

L'annotation des métabolites avec l'algorithme BARSA fonctionne bien, mais la détection des pics sur des spectres RMN 2D se fait manuellement, ce qui est fastidueux dans la réalité. Le premier objectif de ce stage est donc de rendre automatique cette procédure de détection en développant un algorithme pour l'identification automatique des pics dans les spectres RMN 2D. Le second est de combiner cette détection avec l'algorithme BARSA afin de le rendre automatique. Le dernier revient à déterminer comment combiner dans le modèle statistique, l'information extraite des spectres RMN 2D avec celles extraites des spectres RMN 1D.

4 Outils et données utilisés

Tous les travaux liés à ce stage ont été réalisés avec le logiciel **R** et plusieurs packages tels que : **ASICS** (Automatic Statistical Identification in Complex Spectra) pour l'identification et la quantification automatique des métabolites sur des spectres RMN ^1H de mélange complexe (P. J. Tardivel et al. 2017), **reticulate**, **dbscan**, **cluster**, **factoextra**, **NbClust**, **magrittr**, **tibble**, **tidyr**, **dplyr**, **stringr**, **openxlsx** et **gridExtra**. Certaines figures ont été créées à l'aide du package **ggplot2**.

Pour l'annotation des métabolites, les listes des positions de pics des composés de référence ont été récupérés sur **HMDB** (Human Metabolome Database), l'une des premières bases de données dédiées à la métabolomique. C'est une base de données en ligne en accès libre, de haute qualité, et contenant des informations détaillées sur les métabolites présents dans le corps humain.

Dans le cadre de ce stage, il a été mis à ma disposition des données de spectres obtenus sur des échantillons synthétiques dont la composition est connue, ce qui nous permet de savoir ce qu'on cherche et ce qu'on ne cherche pas. On pourra donc évaluer la sensibilité et la spécificité de notre algorithme de détection. De plus, certains échantillons contiennent des composés avec des concentrations différentes : certains métabolites peuvent être présents à des concentrations élevées, tandis que d'autres peuvent être présents à des concentrations très faibles. Cette gamme de concentrations des métabolites nous permet également d'évaluer la sensibilité de notre algorithme à détecter des molécules en grandess ou en petites quantités dans les mélanges.

Les données ont été récupérées sur différentes plateformes analytiques pour tester la généricité des méthodes étudiées.

Dix mélanges synthétiques

Il s'agit de 10 mélanges d'une quinzaine de métabolites en concentrations différentes mais connues (les concentrations sont présentées dans le tableau 2 en annexe). Les données ont été obtenues sur le Plateau technique de Résonnance Magnétique Nucléaire du **MNHN** (Musée National d'Histoire Naturelle) de Paris. Elles contiennent des spectres RMN 1D avec des séquences 2D TOCSY et HSQC. De plus, les mélanges contiennent des métabolites ayant des pics au mêmes endroits, des métabolites n'ayant qu'un seul pic et d'autres avec beaucoup de pics, ainsi que des métabolites faciles à détecter et d'autres moins.

Urine de synthèse

C'est une matrice « blanche » à laquelle ont été ajoutés 33 composés de concentration connue mais différente (100μM $-$ 20mM). Les données obtenues à partir de cette matrice contiennent des spectres RMN 1D avec des séquences 2D TOCSY et HSQC. Sa composition est détaillée dans le tableau 5 en annexe.

Mélange des 23 composés

C'est un mélange de 23 composés standards connus de même concentration (10 mM). Les données obtenues à partir de ce mélange contiennent des spectres RMN 1D avec des séquences 2D COSY, TOCSY et HSQC. Ces composés sont listés en annexe (tableau 3).

Plasma NIST

Plasma de référence humain (NIST SRM 1950), c'est un échantillon biologique réel qui a été largement étudié et déjà caractérisé par plusieurs équipes, en particulier les experts en métabolomique qui ont pu identifier 40 métabolites (la liste de ces métabolites est présentée dans le tableau 4 en annexe). Cet échantillon nous a permis d'évaluer notre algorithme sur une vraie matrice biologique très complexe. Les données obtenues contiennent des spectres RMN 1D avec des séquences 2D COSY et HSQC. Cet échantillon a été utilisé dans l'UMR Toxalim (INRAE Toulouse) pour tester l'algorithme d'annotation 2D BARSA.

Evaluation des méthodes

Pour évaluer les algorithmes de détection de pics et les différentes méthodes de réduction du nombre de faux positifs, j'ai utilisé la sensibilité et la spécificité.

La sensibilité est la capacité à détecter correctement les métabolites qui sont réellement présents dans le mélange étudié. Elle mesure la proportion des vrais positifs (les métabolites correctement identifiés).

La spécificité est la capacité à exclure correctement les métabolites qui ne sont pas dans le mélange étudié. Elle mesure la proportion des vrais négatifs (les métabolites absents correctement identifiés).

	Métabolites		Total
	Présents	Absents	
Détectés	Vrais Positifs (VP)	Faux Positifs (FP)	VP + FP
Non détectés	Faux Négatifs (FN)	Vrais Négatifs (VN)	FN + VN
Total	VP + FN	FP + VN	

La sensibilité et la spécificité sont donc définies par :

$$Sensibilité = \frac{VP}{VP + FN} \quad et \quad Spécificité = \frac{VN}{VN + FP}$$

5 Résultats obtenus et discussions

5.1 Traitement d'images

Le premier challenge a été de définir un seuil sur les intensités de pics afin de réduire le bruit tout en préservant les caractéristiques et les informations importantes du spectre. Trois méthodes (présentées dans une section précédente) ont été testées.

Premièrement, la méthode du maximum n'a pas été performante sur tous les types de spectres RMN 2D que nous avons étudiés car, sur certains spectres, il y a des pics avec de fortes intensités dans les régions sélectionnées. On obtient donc un seuil très élevé, ce qui entraîne la suppression d'une grande partie des pics.

La méthode de l'écart type ne fonctionne pas non plus sur tous les types de spectres RMN 2D que nous étudions car pour certains spectres, l'écart type des intensités est tellement élevé que plusieurs pics d'intérêt sont supprimés même si celui-ci n'est pas multiplié par un facteur multiplicatif.

La troisième méthode, quant à elle fonctionne convenablement mais le challenge avec cette méthode est de trouver un facteur multiplicatif qui pourra fonctionner avec n'importe quel type de spectre, puisque dans l'algorithme final d'identification et de quantification des métabolites, le calcul du seuil de bruit doit se faire automatiquement après l'importation des spectres. Pour cela, plusieurs tests ont été effectués avec plusieurs facteurs multiplicatifs $(1, 2, \ldots, 6)$ sur différentes séquences de spectres RMN 2D. Des exemples avec les séquences COSY et HSQC de l'échantillon Plasma NIST sont présentés dans les figures 10 et 11.

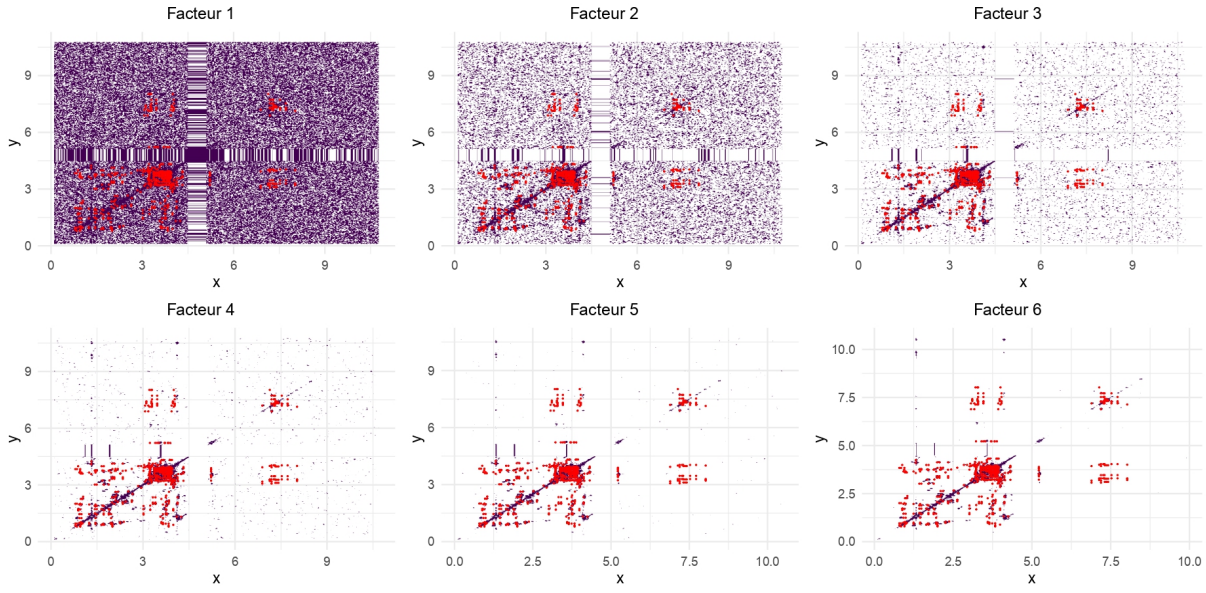


FIGURE 10 – Réduction de bruit sur le spectre COSY du Plasma NIST.

On peut remarquer que le spectre devient de plus en plus net à partir du facteur 4. Mais afin de réduire le risque de suppression de certains pics d'intérêt (en rouge) et au vu des résultats obtenus à partir des séquences COSY et TOCSY des spectres des autres

echantillons utilisés, mon choix s'est porté sur le facteur multiplicatif 3. Ce facteur est également appliqué aux séquences TOCSY.

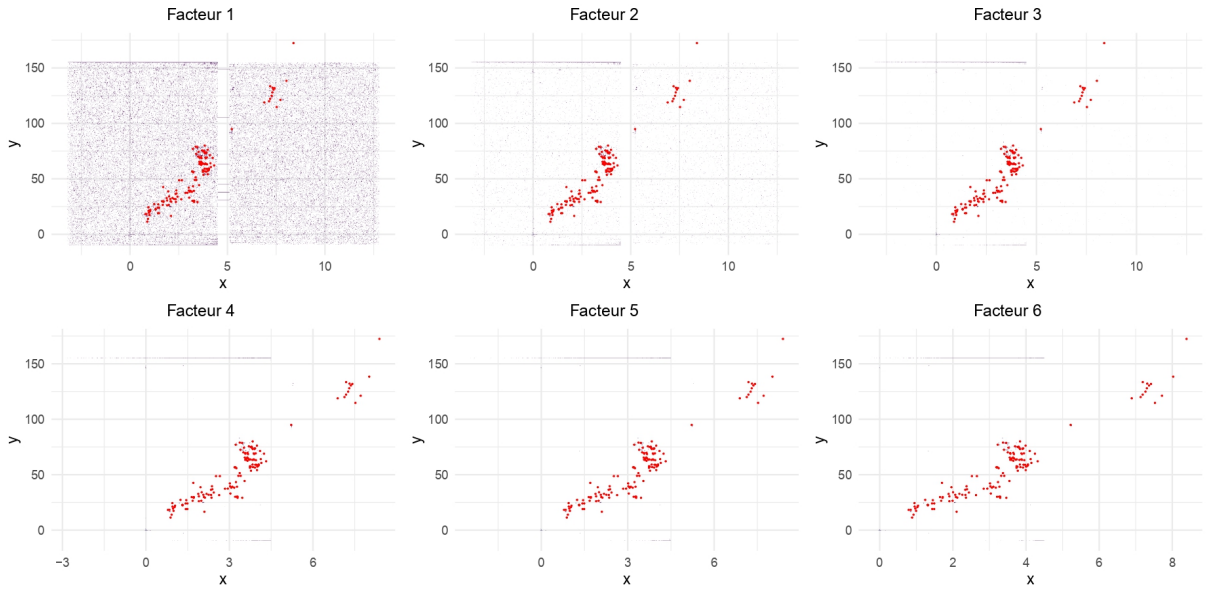


FIGURE 11 – Réduction de bruit sur le spectre HSQC du Plasma NIST.

Le spectre devient de plus en plus net à partir du facteur 3, mais pour la même raison que précédemment, mon choix s'est donc porté sur le facteur multiplicatif 2.

Ces deux facteurs multiplicatifs sont définis par défaut dans les algorithmes de détection de pics et ils pourront être changés par l'utilisateur.

5.2 Choix d'un algorithme de détection de pics

Les cinq méthodes présentées précédemment ont été testées durant ce stage.

La prémière méthode, basée sur la recherche des maxima locaux, produit de nombreux maxima locaux qui ne sont pas pertinents pour notre analyse ; ceux-ci correspondent en majorité à des faux positifs. La méthode basée sur l'homologie persistante vient corriger très légèrement ce problème en identifiant les pics les plus pertinents pour notre analyse.

Les trois autres méthodes sont basées sur du clustering. La prémière (DBSCAN), basée sur la densité, fournit de très bons résultats (surtout avec les séquences HSQC) puisqu'il détecte facilement les régions compactes et denses des spectres. De plus, il gère des quantités variables de bruit et ne nécessite aucune connaissance préalable sur le nombre de clusters. Mais, le problème est qu'il est très sensible à ses deux paramètres principaux (ϵ *et* $minPts$) qui s'influencent mutuellement et dépendent des caractéristiques de chaque spectre. Il existe une méthode pour déterminer les valeurs appropriées de ces deux paramètres, mais cela nécessite la connaissance des vrais clusters, ce qui s'avère impossible dans notre étude, puisque nous sommes en apprentissage non supervisé. Après plusieurs tests sur différentes séquences 2D, j'ai défini par défaut $\epsilon = 0.013$ et $MinPts = 1$ dans l'algorithme de détection de pics avec DBSCAN.

N'ayant pas trouvé un moyen de rendre automatique le calcul de ces deux paramètres pour chaque spectre, je suis donc passé aux méthodes non hiérarchiques (partionnement) et hiérarchiques (ascendantes).

La méthode des K-means a été très vite abandonnée puisqu'elle nécessite la connaissance au préalabre du nombre de clusters. Ce qui n'est pas pratique pour ce stage, car le calcul du nombre de clusters doit se faire automatiquement en fonction de chaque spectre.

Evidemment il existe des méthodes pour déterminer le nombre optimal de clusters à utiliser dans une analyse de clustering. Seulement deux ont été utilisées pendant ce stage. Il s'agit premièrement de la méthode de la silhouette qui est basée sur la distance moyenne entre un point et tous les autres points appartenant au même cluster et également la distance moyenne entre ce point et tous les points du cluster le plus proche différent de celui auquel le point appartient puis la méthode du coude (Elbow method) qui est basée sur la variation de l'inertie intra-cluster (ou la somme des carrés des distances des points au centre de leur cluster) en fonction du nombre de clusters. Le nombre optimal de clusters est généralement associé au point où la courbe présente un « coude », c'est-à-dire où la diminution de l'inertie ralentit considérablement. Malheureusement, ces méthodes sont très gourmandes en temps de calcul, vu les tailles importantes (des dizaines de milliers de points) des données obtenues à partir des spectres. Il faut compter jusqu'à 2 heures de temps environ pour effectuer ce calcul pour un seul spectre.

Le même problème s'est posé avec le clustering ascendant hiérarchique, puisqu'il faut à un moment donné, fixer un nombre de clusters pour élaguer le dendrogramme. Mais une alternative a été trouvée pour contourner ce problème, celle de fixer une profondeur (hauteur) qui sera utilisée par défaut pour faire l'élagage du dendrogramme associé à n'importe quelle séquence. Plusieurs valeurs ont été testées avec différents spectres et la valeur $h = 0.08$, qu'on peut lire sur la figure 12, nous a donné de très bons résultats. Elle a donc été définie par défaut dans l'algorithme de détection de pics et pourra être changée par l'utilisateur. Il faut ajouter que toutes les quatre stratégies d'agrégation ont été testées et c'est le critère du diamètre (complete linkage) qui a été finalement utilisé puisqu'il correspond au mieux à ce que nous recherchons.

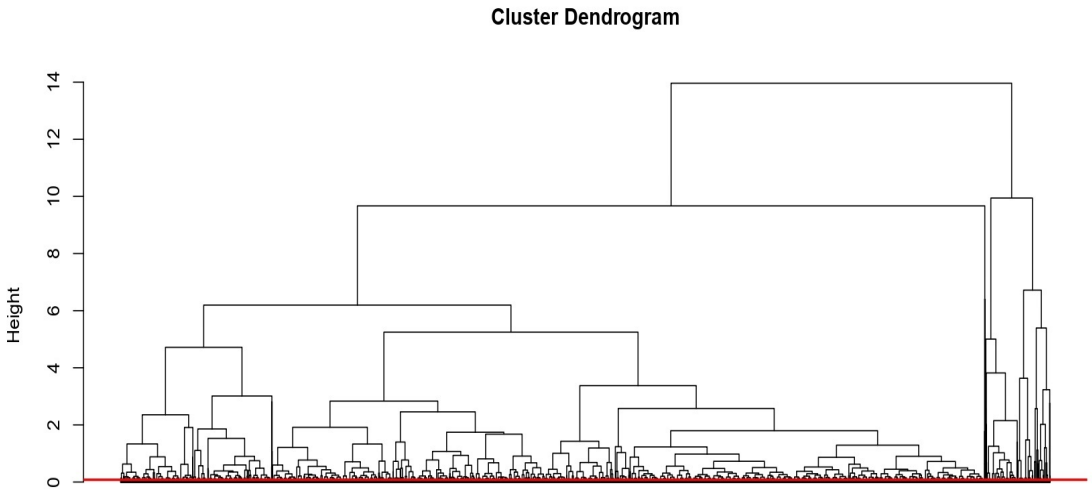


FIGURE 12 – Dendrogramme obtenu avec le spectre COSY du Plasma NIST.

La figure 13 ci-dessous présente les résultats de la détection de pics faite sur la séquence COSY des spectres du Plasma NIST à partir des 4 méthodes retenues.

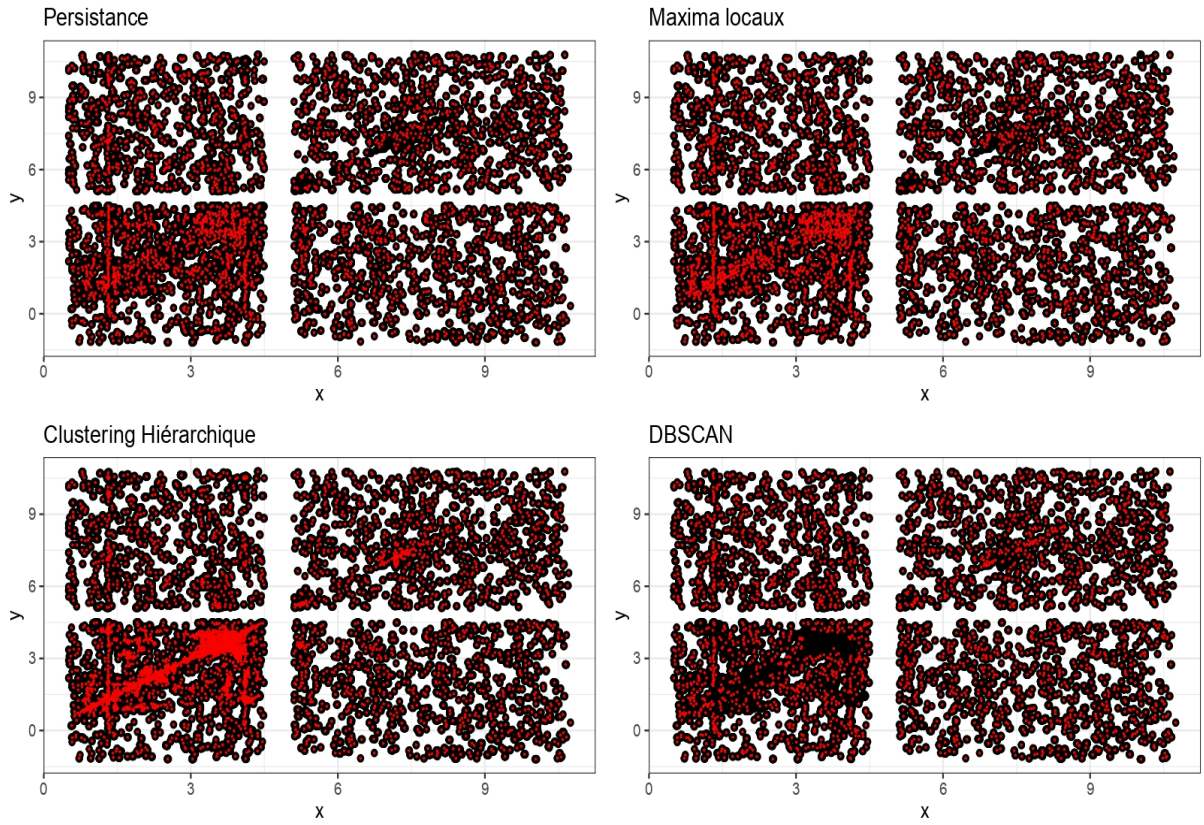


FIGURE 13 – Détection par les 4 méthodes avec le spectre COSY du Plasma NIST.

On remarque avec la pésence de grosses tâches noires que pour les séquences COSY et TOCSY, DBSCAN considère certains ensembles de clusters comme un seul cluster ; ce qui va évidemment jouer sur l'identification des métabolites puisque les pics des métabolites recherchés sont beaucoup plus concentrés dans ces zones non détectées par DBSCAN, comme l'indique la figure 14.

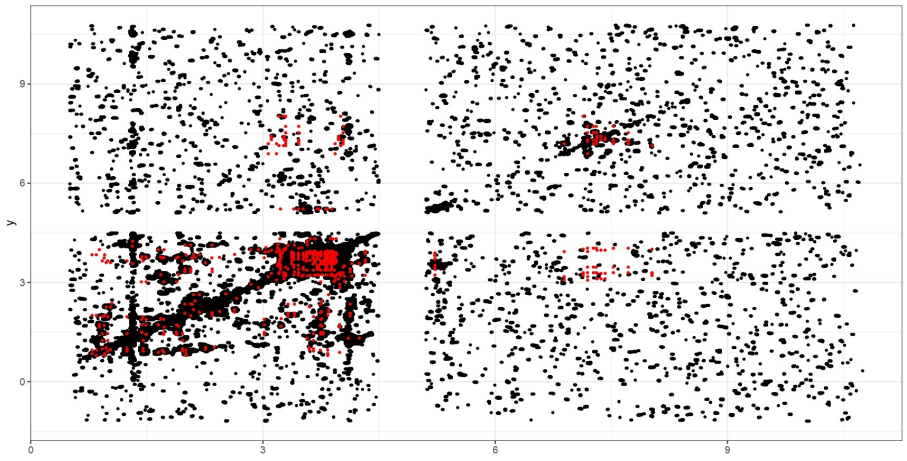


FIGURE 14 – Pics des métabolites recherchés.

Ceci n'est pas le cas avec l'HSQC. Les trois autres méthodes permettent de balayer tous les pics et même ceux qui sont représentés par un seul point. De plus, le clustering détecte beaucoup plus de pics que les deux autres. Cela justifie le fait qu'on a un peu plus de faux positifs (par rapport aux composés théoriques qu'on recherche) avec le clustering hiérarchique dans le cas de la TOCSY et avec DBSCAN dans le cas de l'HSQC, comme on peut le voir sur la figure 15. Il faut noter que dans cette partie, pendant l'annotation des spectres RMN 2D, le seuil sur le score de présence a été laissé à 0.

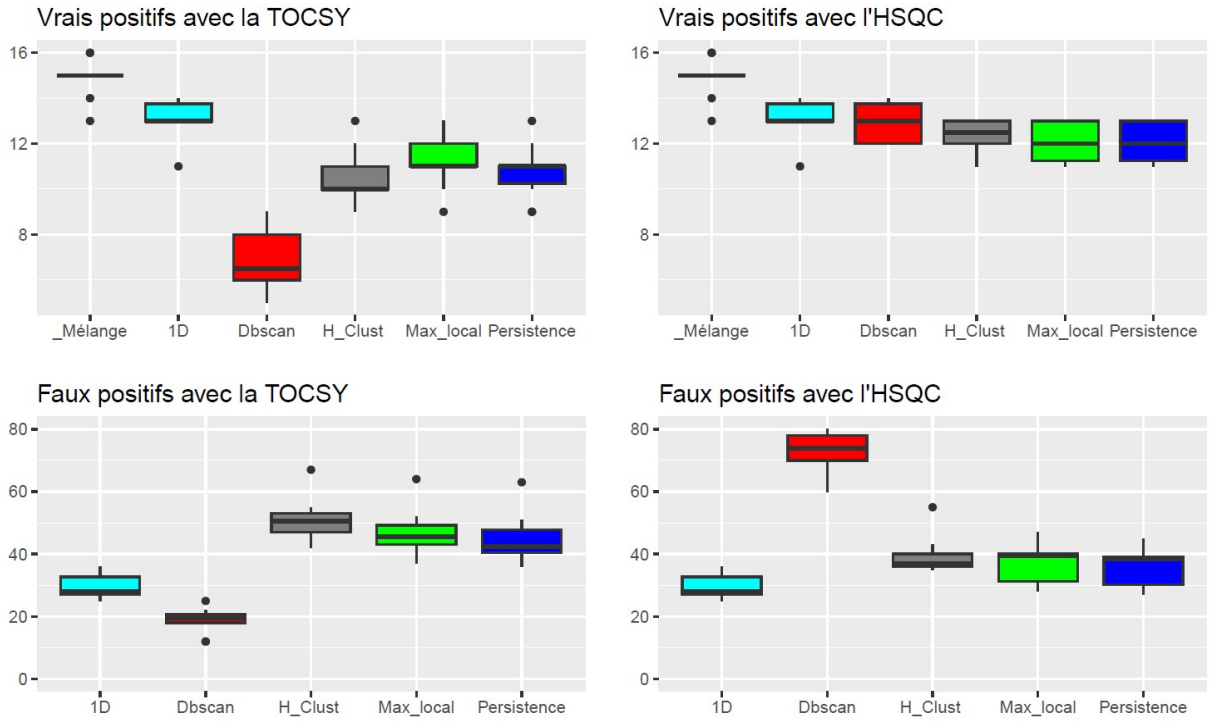


FIGURE 15 – Performances globales des méthodes de détection avec les 10 mélanges.

Chacun des 10 mélanges contient environ une quinzaine de métabolites de concentrations différentes. Le premier boxplot ("Mélange") sur les 2 sous-figures du haut correspond au nombre de métabolites présents par échantillon. On remarque premièrement qu'avec l'HSQC, on identifie beaucoup plus de métabolites qu'avec la TOCSY. De plus, l'HSQC nous permet de nous rapprocher au mieux du contenu réel des mélanges. En termes de vrais positifs, les méthodes de persistance et des maxima locaux nous donnent les meilleurs résultats avec la TOCSY tandis que ce sont les deux méthodes de clustering qui sont meilleures pour l'HSQC. En termes de faux positifs, les méthodes de clustering en produisent plus que les deux autres surtout DBSCAN avec l'HSQC.

Ces constats ont été confirmés avec l'échantillon Plasma NIST. Les résultats obtenus en termes de vrais et faux positifs sont présentés dans la figure 16 ci-après.

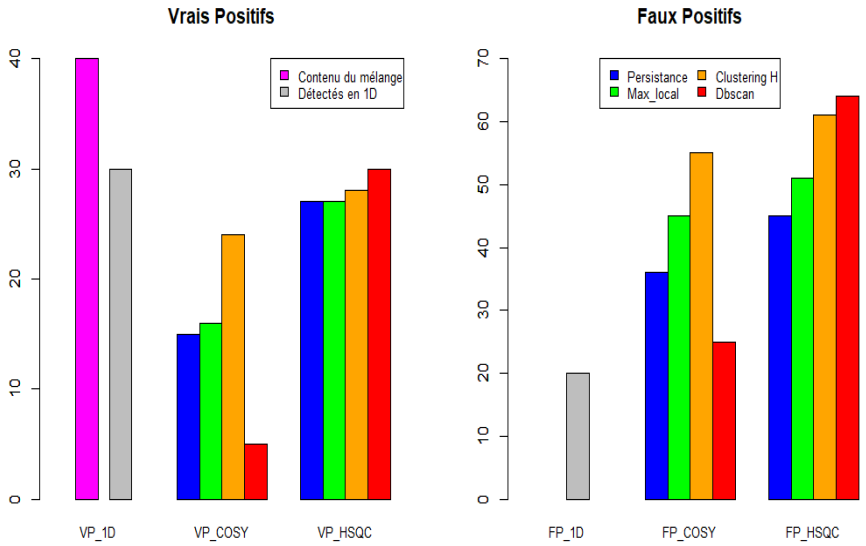


FIGURE 16 – Performances globales des méthodes de détection avec Plasma NIST.

L'objectif principal de ce stage étant de valider les vrais positifs et d'éliminer au maximum les faux positifs détectés en 1D, alors avant de procéder à l'annotation des métabolites dans les spectres 2D, on restreint les bases de données de référence à la liste des métabolites détectés en 1D. Les résultats obtenus sur les 10 mélanges sont présentés dans la figure 17. En effet, après l'annotation, on obtient une nouvelle liste de métabolites (évidemment incluse dans la première) qui contient beaucoup moins de faux positifs. C'est ce que nous voulons, mais il faut également qu'on arrive à garder une grande partie des vrais positifs détectés en 1D.

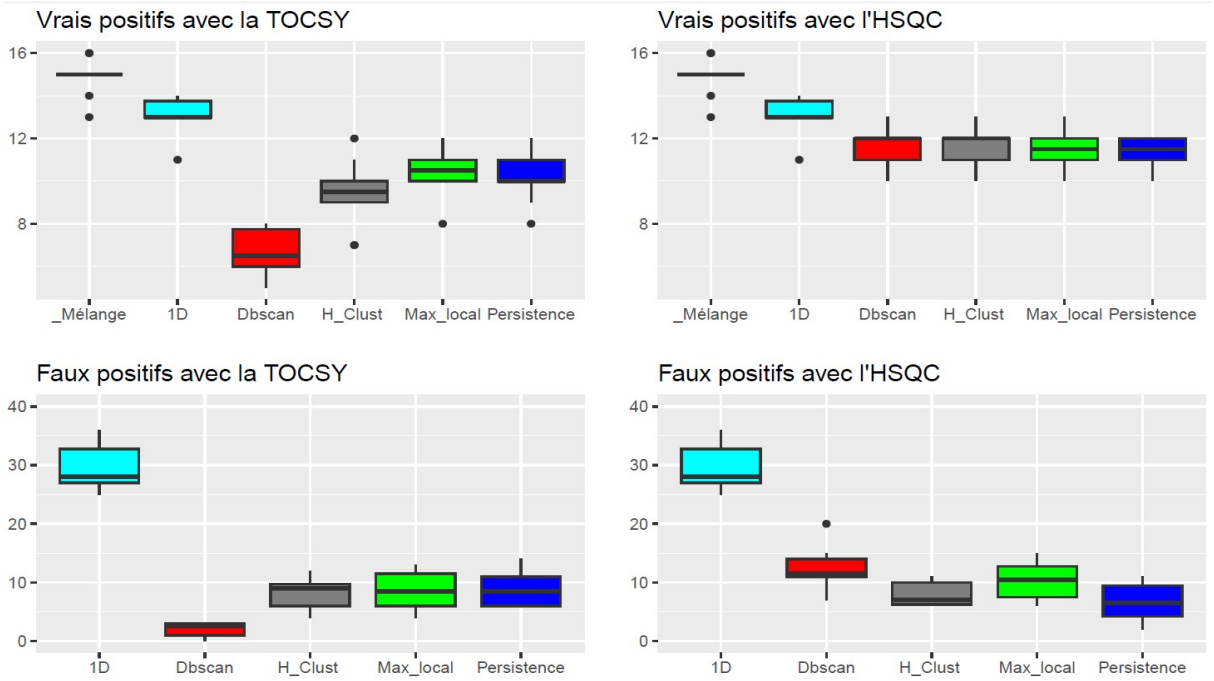


FIGURE 17 – Performances des 4 méthodes de détection avec les 10 mélanges lorsque l'on restreint la liste des métabolites à ceux détectés en 1D.

Avec la TOCSY, les méthodes de persistance et des maxima locaux sont celles qui produisent un bon compromis entre les deux objectifs fixés. Avec l'HSQC, les 4 méthodes ont presque les mêmes performances pour la validation des vrais positifs détectés en 1D. Mais c'est la méthode de persistance qui se démarque pour la réduction des faux positifs. Les résultats obtenus sur l'échantillon Plasma NIST présenté dans la figure 18 viennent confirmer les résultats obtenus avec les 10 mélanges.

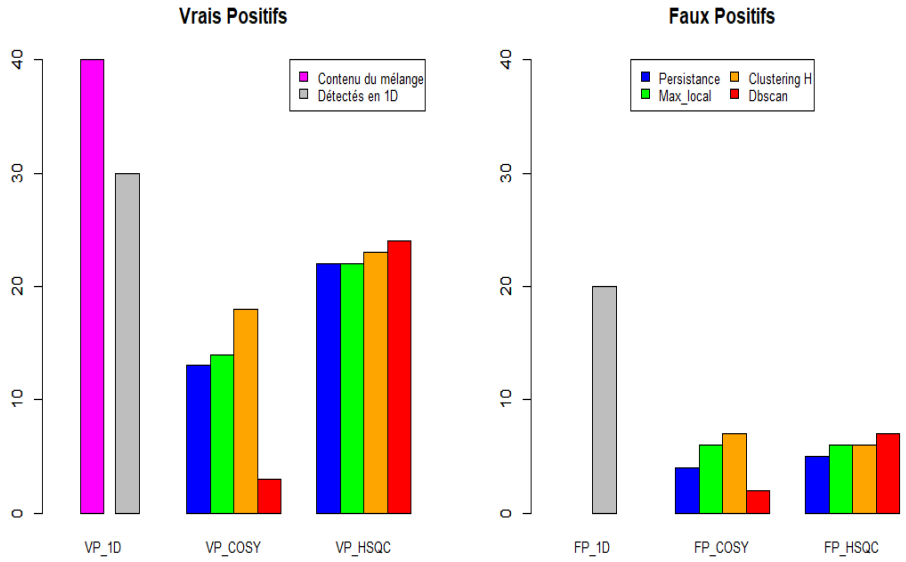


FIGURE 18 – Performances des 4 méthodes de détection avec le Plasma NIST.

Les résultats du clustering hiérarchique sont également meilleurs avec le Plasma NIST, mais en termes de temps d'exécution, cette méthode se montre plus gourmande que les autres. La complexité en temps de chacun des 4 méthodes est donnée dans la figure 19 suivante :

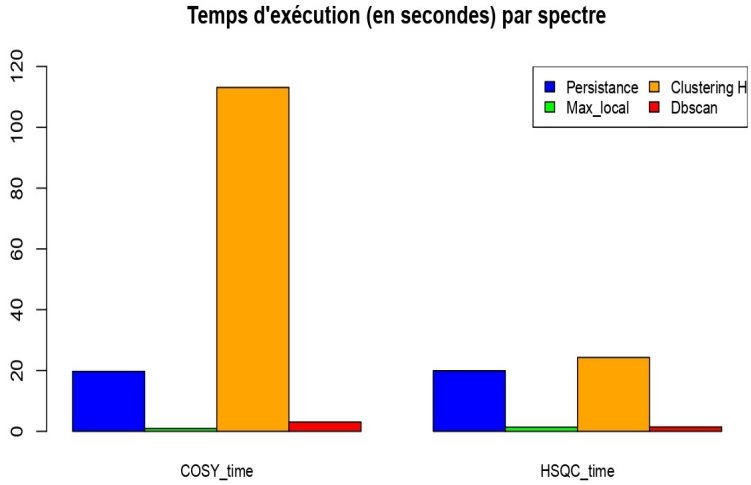


FIGURE 19 – Temps d'exécution des 4 méthodes de détection avec Plasma NIST.

Au vu de tous ces résultats, on peut conclure que la méthode la plus performante pour la détection des pics est celle basée sur la persistance topologique.

La figure 20 ci-dessous présente une brève comparaison entre la détection manuelle et celle faite avec la persistance topologique.

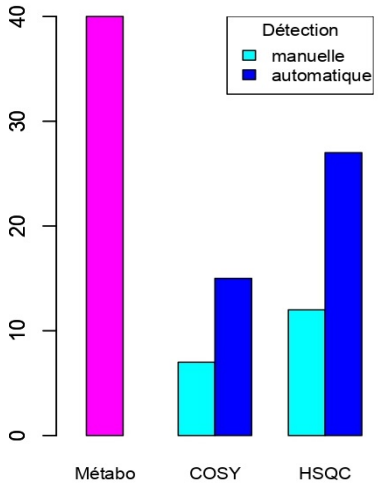


FIGURE 20 – Détection manuelle vs détection automatique avec Plasma NIST.

Sur les 40 métabolites, seulement 7 sont identifiés avec la COSY et 12 avec l'HSQC lorsque la détection est faite manuellement. Mais, lorsqu'on procède à une détection automatique avec la persistance topoligique, on identifie 15 avec la COSY et 27 avec l'HSQC.

5.3 Réduction optimale du nombre de faux positifs

Après l'identification et la quantification des métabolites sur les spectres RMN 1D d'un mélange, on utilise l'algorithme BARSA avec le seuil fixé à zéro pour annoter ces métabolites en utilisant les spectres RMN 2D du mélange. Après cette annotation, il va falloir non seulement choisir soit l'une des séquences ou soit une combinaison de ces séquences mais aussi trouver le seuil optimal sur les scores de présences pour réduire au maximum les faux positifs sans perdre trop de vrais positifs détectés en 1D. Plusieurs approches ont éte testées pour ce but : l'utilisation de la COSY, l'utilisation de la TOCSY, l'utilisation de l'HSQC, la recherche des métabolites détectés à la fois avec toutes les séquences utilisées, la combinaison des séquences en moyennant leurs scores, la recherche des métabolites détectés avec l'une ou l'autre des séquences utilisées ; cela revient à prendre le maximum des scores. Un aperçu de quelques valeurs des scores est présenté dans le tableau 1. Un score nul signifie que le métabolite qui lui est associé n'est pas identifié.

Metabolites	Score_TOCSY	Score_HSQC	Score_moyen	Score_max
Lactate	1	1	1	1
L-Alanine	0,1667	0,3333	0,25	0,3333
L-Isoleucine	0,7	0,8333	0,76665	0,8333
Formate	0	0	0	0
L-Proline	0,5333	1	0,76665	1
L-Leucine	0,1667	0	0,08335	0,1667
L-Lysine	0,4667	1	0,73335	1

TABLE 1 – Quelques valeurs de scores obtenus avec l'un des 10 mélanges

5.3.1 Avec les 10 mélanges

La sensibilité et la spécificité de chaque approche, ont été calculées et il est question de trouver la méthode qui nous fournira un bon compromis entre elles. La première méthode testée consiste à sélectionner que les métabolites détectés à la fois avec toutes les séquences 2D utilisées (la TOCSY et l'HSQC pour les 10 mélanges). On recherche donc les métabolites dont $Score\_TOCSY > 0$ et $Score\_HSQC > 0$. Les résultats obtenus sont présentés dans la figure 21 ci-dessous.

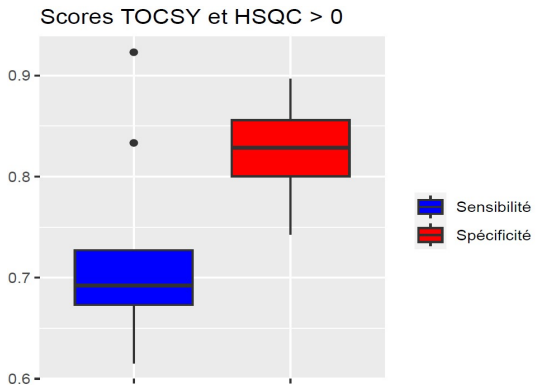


FIGURE 21 – Utilisation des deux séquences à la fois.

Pour les autres approches, nous avons fait varier le seuil sur le score de présence de 0 à 0.5. Les résultats obtenus sont présentés dans les figures 22, 23, 24 et 25 ci-après.

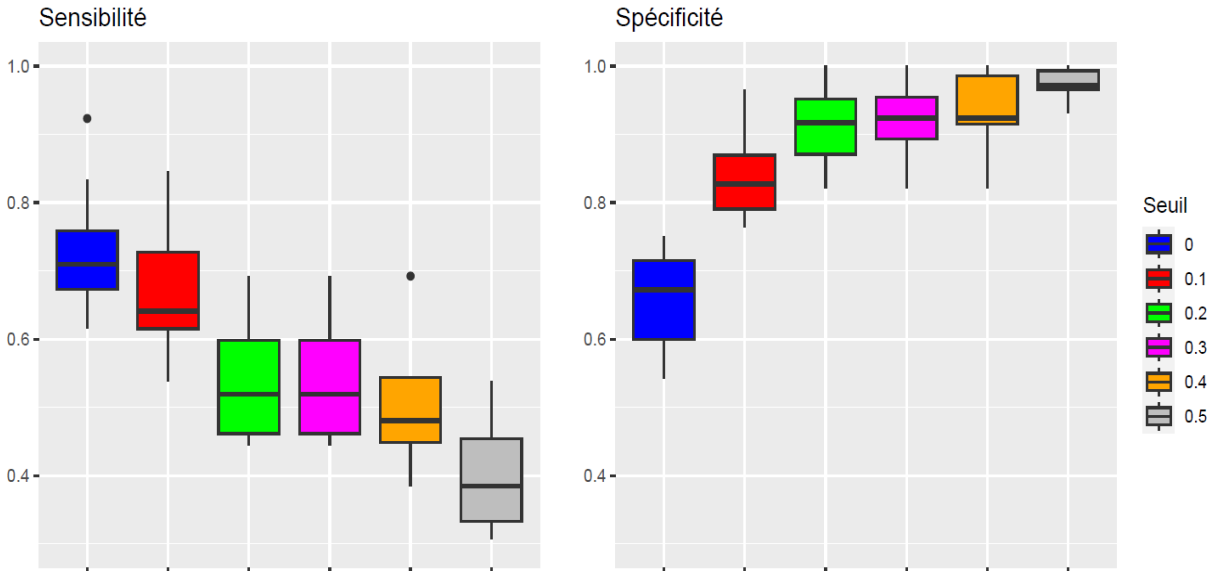


FIGURE 22 – Utilisation de la TOCSY seule.

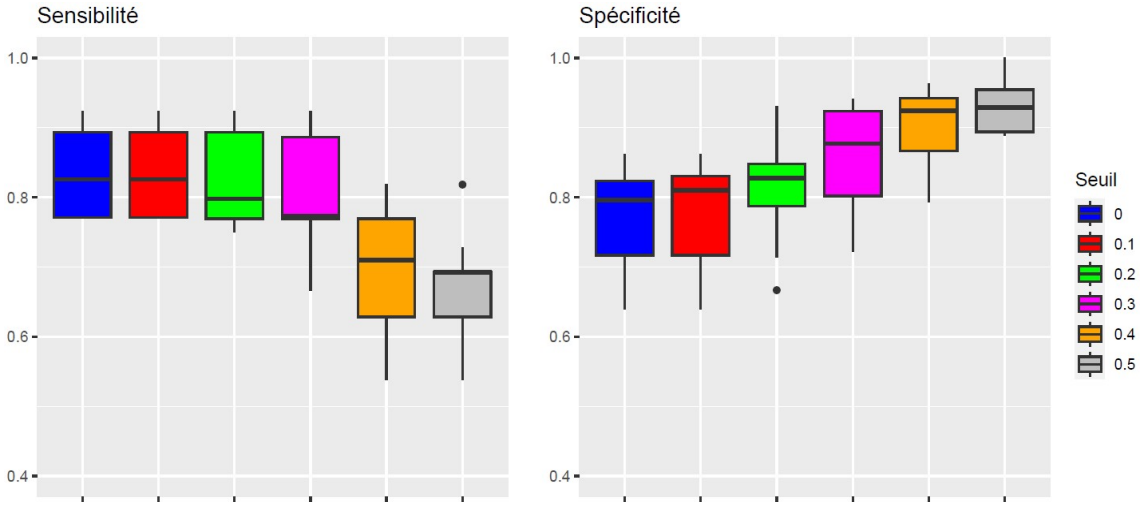


FIGURE 23 – Utilisation de la HSQC seule.

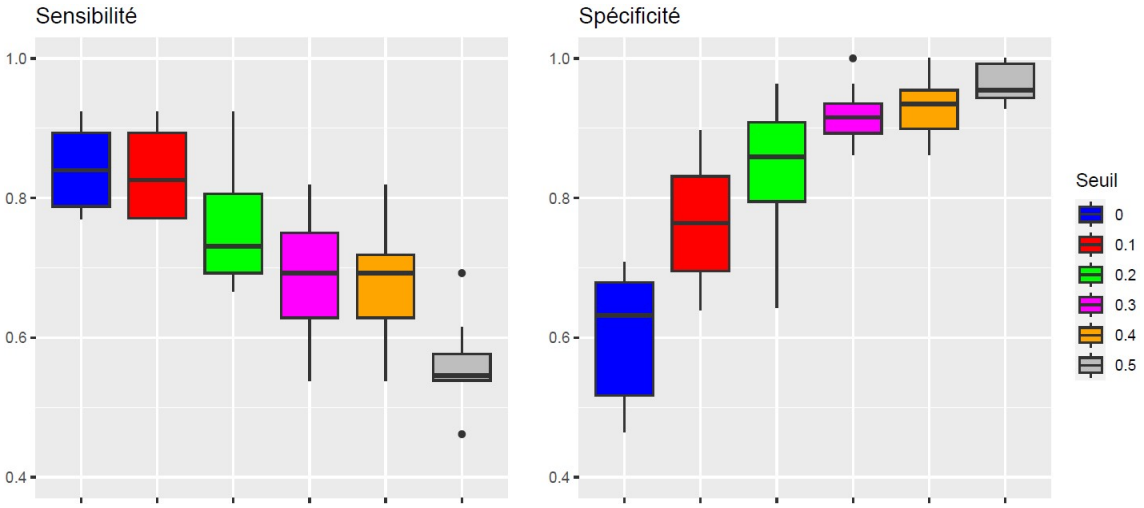


FIGURE 24 – Combinaison des deux séquences avec le score moyen.

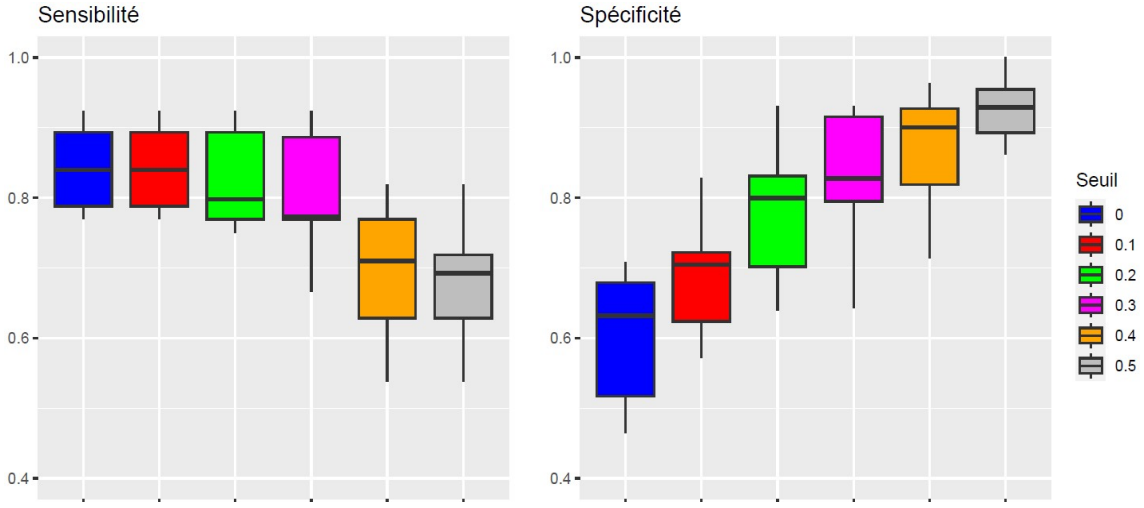


FIGURE 25 – Combinaison des deux séquences avec le score maximal.

On observe que la TOCSY toute seule ne permet pas d'avoir des valeurs élevées de sensibilité et de spécificité même en laissant le seuil à zéro. Dans les autres cas de figures, on a non seulement une bonne sensibilité, et une bonne spécificité ($\sim 80\%$), mais aussi un bon compromis entre les deux surtout avec les scores 0.1 et 0.2 comme l'indique la figure 26 ci-après.

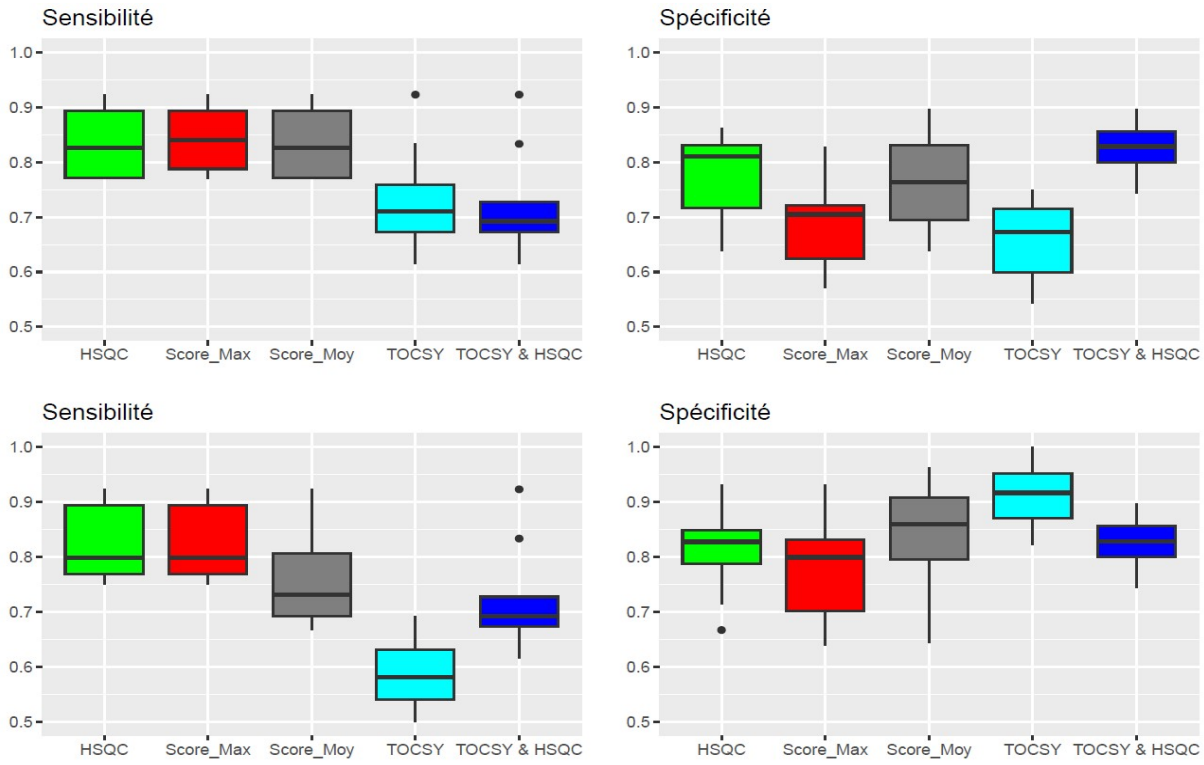


FIGURE 26 – Sensibilité et spécificité de chacune des 5 approches testées sur les 10 mélanges. Pour la TOCSY, l'HSQC, le score moyen et le score maximal, le seuil a été fixé à 0.1 sur les deux premières sous-figures et 0.2 sur les deux dernières.

Au vu de ces résultats, on est tenté de garder seulement l'HSQC pour l'annotation des métabolites. Mais, certains métabolites peuvent être identifiés avec la COSY ou la TOCSY sans être identifiés avec l'HSQC. Alors il est préférable de combiner toutes les séquences utilisées pour prendre tous les métabolites en compte. on peut réévaluer ces méthodes en utilisant le Plasma NIST.

5.3.2 Avec le Plasma NIST

Le Plasma NIST étant un échantillon, on a donc un seul spectre 1D avec les séquences COSY et HSQC. Les résultats obtenus sont présentés dans la figure 27.

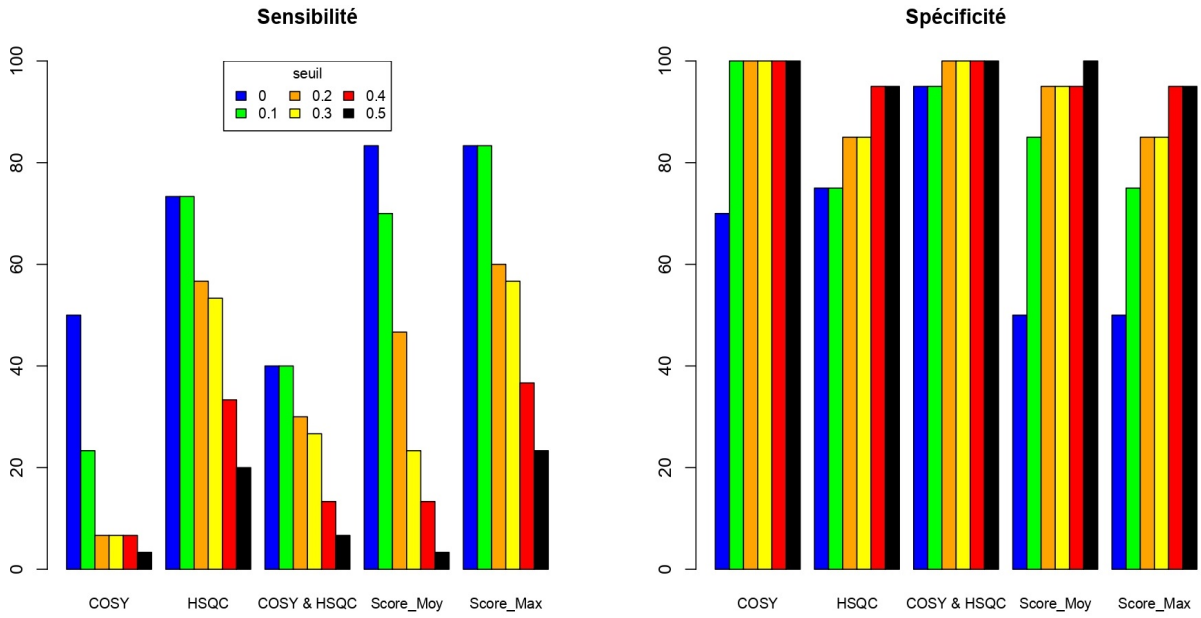


FIGURE 27 – Performances de la 2D avec Plasma NIST.

On peut voir que la combinaison des deux séquences avec le score maximal est l'approche la plus performante, et ceci avec le seuil fixé à 0.1 (la couleur verte). Le même constat est fait lorsqu'on évalue la performance globale de l'ensemble de la méthode (1D + 2D), toujours sur le Plasma NIST, comme l'indique le figure 28.

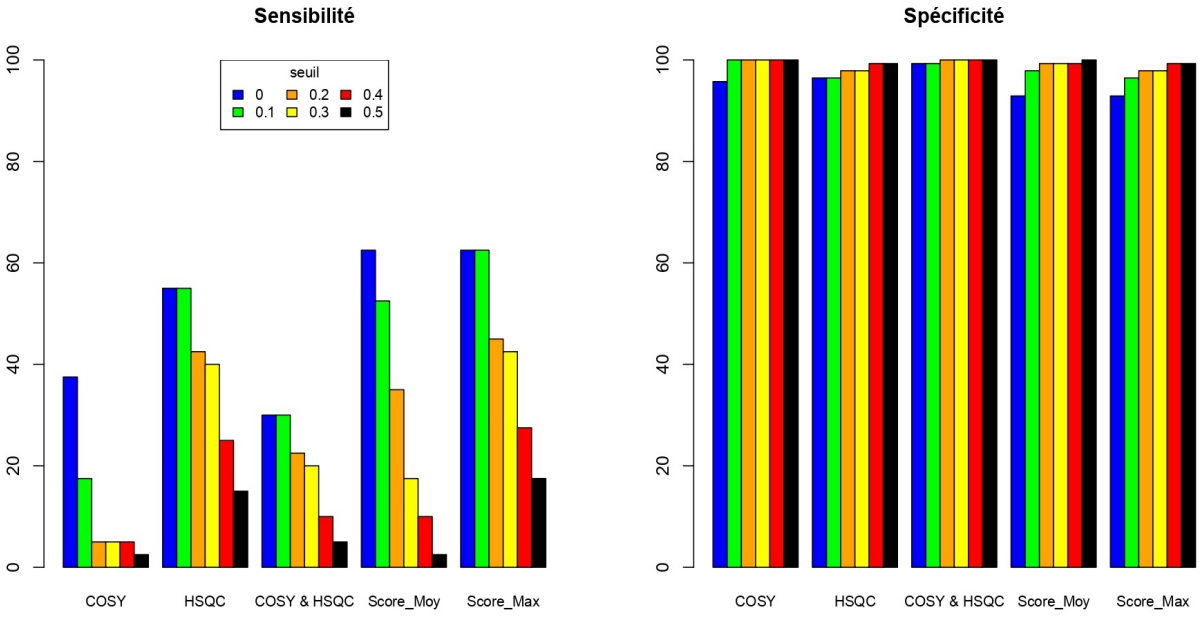


FIGURE 28 – Performances de l'enchainement 1D puis 2D avec Plasma NIST.

Récapitulatif

Les experts en métabolomique ont identifié 40 métabolites dans l'échantillon du Plasma NIST. Comme présenté dans la figure 29, sur ces 40 métabolites, 30 ont été identifiés en spectroscopie RMN 1D avec 20 faux positifs. En appliquant la combinaison des deux séquences avec le maximum des scores et en fixant le seuil à 0.1, la spectroscopie RMN 2D nous a permis de valider 25 vrais positifs sur les 30 et de réduire le nombre de faux positifs de 20 à 5 ; un bon résultat.

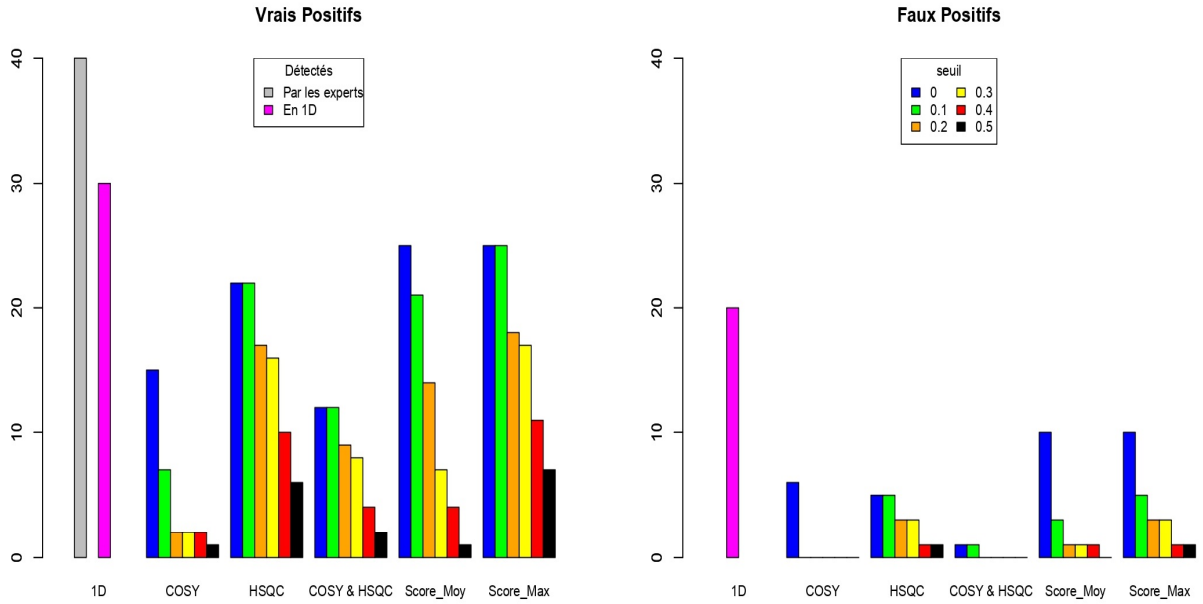


FIGURE 29 – Récapitulatif sur les détections avec le Plasma NIST.

Conclusion et perspectives

Ce stage fut une expérience très enrichissante tant sur le plan professionnel que personnel, puisqu'il m'a permis d'augmenter mon champ de compétences. J'ai eu à travailler sur un problème rencontré en métabolomique, un domaine très récent et en plein essor, qui vise à caractériser la composition d'un système biologique. Comme dans toute phase d'apprentissage, dans les débuts, j'ai eu du mal à intégrer les notions liées à ce domaine qui est complètement nouveau pour moi. Mais, malgré des études purement mathématiques, la métabolomique a été d'un grand d'intérêt pour moi puisque ce stage m'a fait découvrir ce que la combinaison de ces deux champs de recherche associés à l'informatique pouvait apporter.

Dans le but de rendre automatique la détection des taches sur des spectres RMN 2D, plusieurs méthodes ont été comparées et parmi celles-ci la méthode basée sur la persistance topologique s'est montrée supérieure aux autres en termes de sensibilité, de spécificité et de complexité de temps de calcul. De plus, la combinaison de toutes les séquences utilisées avec le score maximal est l'approche la plus performante, et celle-ci avec le seuil fixé à 0.1. Pour valider mes choix, il faudra tester les méthodes choisies sur d'autres jeux de données reéls.

Plusieurs problèmes ont été rencontrés tout au long de ce stage. Les premiers concernent l'identification des métabolites puisque tous les métabolites présents dans un échantillon ne sont pas nécessairement connus ou répertoriés dans les bases de données de référence. Certains métabolites peuvent être des composés nouveaux ou rares, dont la structure chimique n'a pas encore été caractérisée. Il va falloir mettre à jour régulièrement les bases de données des composés de référence pour optimiser les résultats d'annotation. De plus, certains métabolites sont bien présents, mais leur liste de pics est vide : TMAO, L-Glycine, Formate, *etc.,* par exemple ont leurs peaklists vides dans la base de données de la TOCSY. De tels métabolites ne pourront jamais être identifiés avec la COSY seule ou la TOCSY seule quelque soit la performance de l'algorithme d'annotation utilisé. C'est pour cette raison qu'il est plus adéquat de combiner les différentes séquences 2D COSY, TOCSY et HSQC. Il faut ajouter que contrairement à la spectroscopie RMN 1D, les pré-traitements des spectres 2D ne sont pas encore totalement automatisés ; cela pourrait faire l'objet d'un stage à part entière. Enfin, ce travail que j'ai réalisé durant mon stage sera implémenté dans un package *R*.

Références

Articles

Alexander, James W (1926). "Combinatorial analysis situs". In : *Transactions of the American Mathematical Society* 28.2, p. 301-329.

Boukedimi, Yasmin (2014). "Développement d'une méthode de profilage métabolomique semi-ciblée des métabolites endogènes chez C. Elegans par chromatographie liquide couplée à la spectrométrie de masse". In.

Bria, Marc et Pierre Watkin (1997). "La spectroscopie de resonance magnetique nucleaire a deux dimensions ou l'aide a la determination structurale des molecules organiques". In : *Actualite Chimique* 5.2, p. 24-35.

Callet, Victoria (2019). "Suites spectrales et Homologie persistante". In.

Chazal, Frédéric et Bertrand Michel (2021). "An introduction to topological data analysis : fundamental and practical aspects for data scientists". In : *Frontiers in artificial intelligence* 4, p. 667963.

Courtieu, Jacques, Nicolas Giraud, Olivier Lafon, Philippe Lesot, Cédric Lorthioir et Jean-Marc Nuzillard (2012). "La RMN en chimie organique". In : *l'actualité chimique* 364-365.

Dormoy, Valérian et Thierry Massfelder (2013). "La métabolomique au service de la médecine-L'exemple du carcinome rénal". In : *médecine/sciences* 29.5, p. 463-468.

Dumez, Jean-Nicolas (2022). "La RMN diffusionnelle ultrarapide analyse un mélange en moins d'une seconde". In : *L'Actualité Chimique*.

Dunn, Warwick B, Alexander Erban, Ralf JM Weber, Darren J Creek, Marie Brown, Rainer Breitling, Thomas Hankemeier, Royston Goodacre, Steffen Neumann, Joachim Kopka et al. (2013). "Mass appeal : metabolite identification in mass spectrometry-focused untargeted metabolomics". In : *Metabolomics* 9, p. 44-66.

Ellero-Simatos, S, S Claus et H Guillou (2019). "La métabolomique : applications médicales". In : *Médecine des Maladies Métaboliques* 13.3, p. 263-267.

Fiehn, Oliver (2002). "Metabolomics–the link between genotypes and phenotypes". In : *Plant molecular biology* 48, p. 155-171.

Gómez, Josep, Jesús Brezmes, Roger Mallol, Miguel A Rodriguez, Maria Vinaixa, Reza M Salek, Xavier Correig et Nicolau Cañellas (2014). "Dolphin : A tool for automatic targeted metabolite profiling using 1D and 2D 1 H-NMR data". In : *Analytical and Bioanalytical Chemistry* 406, p. 7967-7976.

Hahsler, Michael, Matthew Piekenbrock et Derek Doran (2019). "dbscan : Fast density-based clustering with R". In : *Journal of Statistical Software* 91, p. 1-30.

Hao, Jie, William Astle, Maria De Iorio et Timothy MD Ebbels (2012). "BATMAN—an R package for the automated quantification of metabolites from nuclear magnetic resonance spectra using a Bayesian model". In : *Bioinformatics* 28.15, p. 2088-2090.

Huber, Stefan (2021). "Persistent Homology in Data Science". In.

Illig, Thomas, Christian Gieger, Guangju Zhai, Werner Römisch-Margl, Rui Wang-Sattler, Cornelia Prehn, Elisabeth Altmaier, Gabi Kastenmüller, Bernet S Kato, Hans-Werner Mewes et al. (2010). "A genome-wide perspective of genetic variation in human metabolism". In : *Nature genetics* 42.2, p. 137-141.

Lefort, Gaëlle, Laurence Liaubet, Cécile Canlet, Patrick Tardivel, Marie-Christine Père, Hélène Quesnel, Alain Paris, Nathalie Iannuccelli, Nathalie Vialaneix et Rémi Servien (2019). "ASICS : an R package for a whole analysis workflow of 1D 1H NMR spectra". In : *Bioinformatics* 35.21, p. 4356-4363.

Mercier, Pascal, Michael J Lewis, David Chang, David Baker et David S Wishart (2011). "Towards automatic metabolomic profiling of high-resolution one-dimensional proton NMR spectra". In : *Journal of biomolecular NMR* 49, p. 307-323.

Mottet, Antoine (2011). "Homologie Persistante et application en sécurité informatique". In.

Nicholson, Jeremy K, John C Lindon et Elaine Holmes (1999). "'Metabonomics' : understanding the metabolic responses of living systems to pathophysiological stimuli via multivariate statistical analysis of biological NMR spectroscopic data". In : *xenobiotica* 29.11, p. 1181-1189.

Oakman, Catherine, Leonardo Tenori, Laura Biganzoli, Libero Santarpia, Silvia Cappadona, Claudio Luchinat et Angelo Di Leo (2011). "Uncovering the metabolomic fingerprint of breast cancer". In : *The international journal of biochemistry & cell biology* 43.7, p. 1010-1020.

Ravanbakhsh, Siamak, Philip Liu, Trent C Bjordahl, Rupasri Mandal, Jason R Grant, Michael Wilson, Roman Eisner, Igor Sinelnikov, Xiaoyu Hu, Claudio Luchinat et al. (2015). "Accurate, fully-automated NMR spectral profiling for metabolomics". In : *PloS one* 10.5, e0124219.

Savorani, Francesco, Morten Arendt Rasmussen, Åsmund Rinnan et Søren Balling Engelsen (2013). "Interval-based chemometric methods in NMR foodomics". In : 28, p. 449-486.

Tardivel, Patrick JC, Cécile Canlet, Gaëlle Lefort, Marie Tremblay-Franco, Laurent Debrauwer, Didier Concordet et Rémi Servien (2017). "ASICS : an automatic method for identification and quantification of metabolites in complex 1D 1 H NMR spectra". In : *Metabolomics* 13, p. 1-9.

Tremblay Franco, Marie et Cécile Canlet (2023). "BARSA, Un algorithme pour l'annotation automatique de spectres RMN 2D de matrices complexes". In.

Worley, Bradley et Robert Powers (2016). "PCA as a practical indicator of OPLS-DA model reliability". In : *Current Metabolomics* 4.2, p. 97-103.

Thèses

Lefort, Gaëlle (2021). "Quantification automatique de métabolites dans un spectre RMN et application à la description de la maturité périnatale chez le porc". Thèse de doct.

Martel, Dimitri (2015). "Spectroscopie 2D de corrélation quantitative : Méthode de quantification, Études expérimentales Et applications in vivo". Thèse de doct. INSA de Lyon.

Tardivel, Patrick (2017). "Représentation parcimonieuse et procédures de tests multiples : application à la métabolomique". Thèse de doct. Université Paul Sabatier-Toulouse III.

Annexes

A1 - Composition des mélanges

Métabolites	***Mix1***	***Mix2***	***Mix3***	***Mix4***	***Mix5***	***Mix6***	***Mix7***	***Mix8***	***Mix9***	***Mix10***
D-Fucose	*100*	*0*	*100*	*0*	*500*	*150*	*250*	*150*	*500*	*250*
D-Glucose	*250*	*500*	*100*	*500*	*100*	*150*	*250*	*500*	*250*	*150*
L-Alanine	*500*	*100*	*100*	*500*	*100*	*20*	*20*	*250*	*250*	*250*
L-Histidine	*50*	*250*	*250*	*20*	*150*	*50*	*100*	*150*	*100*	*20*
L-Isoleucine	*100*	*50*	*50*	*250*	*250*	*100*	*0*	*150*	*150*	*100*
L-Lysine	*150*	*250*	*0*	*100*	*500*	*250*	*0*	*500*	*100*	*150*
L-Ornithine	*20*	*100*	*100*	*20*	*50*	*150*	*20*	*50*	*0*	*150*
L-Proline	*100*	*250*	*50*	*150*	*100*	*150*	*250*	*0*	*50*	*0*
L-Tryptophane	*20*	*150*	*50*	*100*	*20*	*250*	*250*	*150*	*100*	*50*
Myo-Inositol	*150*	*0*	*250*	*100*	*0*	*250*	*150*	*50*	*100*	*50*
TMAO	*0*	*150*	*150*	*100*	*100*	*50*	*50*	*0*	*250*	*250*
Betaine	*250*	*0*	*250*	*50*	*100*	*100*	*20*	*20*	*100*	*50*
L-Glycine	*0*	*250*	*100*	*100*	*50*	*100*	*250*	*0*	*50*	*50*
L-Threonine	*100*	*50*	*0*	*250*	*150*	*0*	*50*	*150*	*250*	*100*
Lactate	*500*	*250*	*250*	*20*	*0*	*150*	*150*	*20*	*100*	*100*
Glycerol	*50*	*100*	*250*	*100*	*150*	*150*	*50*	*0*	*0*	*250*
Formate	*100*	*50*	*20*	*0*	*20*	*100*	*250*	*50*	*50*	*100*

Table 2 – Composition des 10 mélanges.

L-Alanine	*L-Arginine*	*L-Asparagine*	*L-Aspartate*
Betaine	*Butyrate*	*Cadaverine*	*L-Carnitine*
CholineChloride	*Dimethylamine*	*FumaricAcid*	*D-Galactose*
D-Glucose	*L-Glutamine*	*L-Glycine*	*HippuricAcid*
L-Histidine	*L-Isoleucine*	*Lactate*	*L-Proline*
L-Threonine	*L-Tryptophane*	*L-Valine*	

Table 3 – Liste des 23 métabolites.

2-HydroxybutyricAcid	*2-oxoisocaproic acid*	*3-Hydroxybutyrate*	*2-Oxoisovalerate*
AceticAcid	*L-Alanine*	*L-Arginine*	*L-Asparagine*
Betaine	*L-Carnitine*	*CholineChloride*	*Citrate*
Creatine	*Creatinine*	*Formate*	*D-Glucose*
L-GlutamicAcid	*L-Glutamine*	*Glycerol*	*Glycerophosphocholine*
L-Glycine	*L-Histidine*	*Isobutyrate*	*L-Isoleucine*
Lactate	*L-Leucine*	*L-Lysine*	*D-Mannose*
Methanol	*L-Methionine*	*L-Phenylalanine*	*Phosphocholine*
L-Proline	*Pyruvic-Acid*	*L-Serine*	*Succinate*
L-Threonine	*L-Tryptophane*	*L-Tyrosine*	*L-Valine*

Table 4 – Liste des 40 métabolites identifiés par les experts avec le Plasma NIST.

Métabolites	Literature values (µM/mM)	Concentration (µM)	Masse molaire (g/mol)	Quantité (mg) 1L
1-Methyl-L-Histidine	15,9	197	169,18	33,30
2-HydroxyglutaricAcid	33	409	148,11	60,51
3-Methyl-L-Histidine	15,1	187	169,18	31,63
AceticAcid	45,3	561	60,05	33,68
Acetoacetate	33	409	102,09	41,71
Allantoin	19,7	244	158,12	38,56
Creatine	46	569	131,13	74,68
L-Cysteine	33,4	413	121,16	50,10
D-Glucose	31,12	385	180,16	69,41
Dimethylamine	39,3	487	45,08	21,93
Ethanolamine	21,4	265	61,08	16,18
Formate	20,39	252	46,03	11,62
L-Glutamine	33,32	413	146,14	60,28
Glycerol	20	248	92,09	22,80
L-Glycine	151	1869	75,07	140,33
GuanidoaceticAcid	89	1102	117,11	129,03
HippuricAcid	257	3182	179,18	570,07
Indoxylsulfate	19,74	244	213,21	52,10
IsocitricAcid	58	718	192,12	137,95
Lactate	12,3	152	90,08	13,72
L-alanine	22	272	89,09	24,27
L-Histidine	60,75	752	155,16	116,69
L-Lysine	18	223	146,19	32,58
Myo-Inositol	18,8	233	180,16	41,93
Phenylacetylglutamine	47,03	582	264,28	153,87
PyroglutamicAcid	28,8	357	129,12	46,04
L-Serine	26	322	105,09	33,83
Trigonelline	16,08	199	173,60	34,56
TMAO	118,7	1470	75,11	110,38
D-Fucose	x	99	180,16	17,84
L-Threonine	x	184	119,1	21,91

Table 5 – Composition de l'urine de synthèse.

A2 - Les scripts R

Importation des spectres 2D

inDir <- "../data/MTH-Plasma-NIST-090416/8/pdata/1"

Internal function for parsing Bruker processing files

```
parseProcs <- function(inDir, params){

  ## Designate parameters if not provided
  if (missing(params))
    params <- c('SW_p','SF','SI','OFFSET','NC_proc','BYTORDP','XDIM')
  paramVar <- paste('##$', params, sep='')

  ## Search inDir for necessary processing parameter files
  procs <- list.files(inDir, full.names=TRUE, pattern='^procs$')[1]
  if (is.na(procs))
    procs <- list.files(inDir, full.names=TRUE, pattern='^proc$')[1]
  if (is.na(procs)) {
    paste('Could not find processing parameter files ("proc" or "procs")',
          ' in:\n"', inDir, '".', sep='')
    return()
  }
  proc2s <- list.files(inDir, full.names=TRUE, pattern='^proc2s$')[1]
  if (is.na(proc2s))
    proc2s <- list.files(inDir, full.names=TRUE, pattern='^proc2$')[1]
  if (is.na(proc2s))
    files <- procs
  else
    files <- c(procs, proc2s)

  ## Search processing files for designated parameters
  pars <- NULL
  for (i in seq_along(files)){

    ## Determine paramater/value separator
    for (paramSep in c('= ', '=', ' =', ' = ')){
      splitText <- strsplit(readLines(files[i]), paramSep)
      parNames <- sapply(splitText, function(x) x[1])
      parVals <- sapply(splitText, function(x) x[2])
      matches <- match(paramVar, parNames)
      if (any(is.na(matches)))
        next
      else
        break
    }

    ## Return an error if any parameters can not be found
    if (any(is.na(matches)))
      stop(paste('One or more of the following parameters could not be found: ',
                 paste("'", params[which(is.na(matches))], "'", sep='',
                       collapse=', '), ' in:\n"', files[i], sep=''))
    pars <- rbind(pars, parVals[matches])
```

```
  }

  ## Format the data
  colnames(pars) <- params
  pars <- data.frame(pars, stringsAsFactors=FALSE)
  if (!is.null(pars$BYTORDP))
    pars$BYTORDP <- ifelse(as.numeric(pars$BYTORDP), 'big', 'little')
  if (!is.na(proc2s))
    rownames(pars) <- c('w2', 'w1')
  pars$SW=as.numeric(pars$SW_p)/as.numeric(pars$SF)
  pars=pars[3:8]
  return(pars)
}
```

Importation des spectres

```
import2d <- function(inDir) {

  # aquisition parameters
  storedpars <- parseProcs(inDir)
  storedpars$OFFSET <- as.numeric(storedpars$OFFSET)
  storedpars$XDIM <- as.numeric(storedpars$XDIM)
  storedpars$NC_proc <- as.numeric(storedpars$NC_proc)
  storedpars$SW <- as.numeric(storedpars$SW)

  # ppm grid
  w1 <- seq(storedpars$OFFSET[1], storedpars$OFFSET[1] -
              storedpars$SW[1], length.out = storedpars$XDIM[1])
  w2 <-  seq(storedpars$OFFSET[2], storedpars$OFFSET[2] -
               storedpars$SW[2], length.out = storedpars$XDIM[2])

  # read spectrum
  readCon <- file(file.path(inDir, "2rr"), "rb")
  spec2d <- matrix(as.numeric(readBin(readCon, size = 4,
                                      what = "integer", n = storedpars$XDIM[1] *
                                        storedpars$XDIM[2],
                                      endian = storedpars$BYTORDP[1])),
                   nrow = storedpars$XDIM[1],
                   ncol = storedpars$XDIM[2])
  close(readCon)

  spec2d <- spec2d / (2^-storedpars$NC_proc)

  rownames(spec2d) <- w1
  colnames(spec2d) <- w2
  return(spec2d)
}
```

PLOT DES SPECTRES 2D AVEC OU SANS IDENTIFICATION DES PICS

exclusion.areas : région à exclure
type : nom sequence 2D a utiliser (TOCSY",“HSQC")
level : facteur multiplicatif pour le calcul su seuil pour la réduction de bruit

Les packages nécessaires

```
library(magrittr)   ## Pour utiliser l'opérateur %>% qui permet d'enchaîner des fonctions
library(tibble)  ## pour la fonction rownames_to_column
library(tidyr)  # pour la fonction pivot_longer
library(dplyr)  # pour la fonction mutate

library(ggplot2)

plot2d <- function(spec2d, level,
                   exclusion.areas = list(matrix(c(4.5, 5.1), ncol = 2),
                                          matrix(c(4.5, 5.1), ncol = 2)),
                   peaks = NULL, type = c("TOCSY", "HSQC")[1]) {
  min_val <- NA
  if(missing(level)){
    if(type == "TOCSY"){
      min_val <- ((IQR(spec2d)/2)/0.674) * 3
    }
    else{
      min_val <- ((IQR(spec2d)/2)/0.674) * 2
    }
  }
  else{
    min_val <- ((IQR(spec2d)/2)/0.674) * level
  }
  ## remove noise and exclusion areas
  spec2d[spec2d <= min_val] <- 0

  # exclude on x-axis
  for (i in seq_len(nrow(exclusion.areas[[1]]))) {
    spec2d[as.numeric(rownames(spec2d)) > exclusion.areas[[1]][i, 1] &
             as.numeric(rownames(spec2d)) < exclusion.areas[[1]][i, 2], ] <- 0
  }
  # exclude on y-axis
  for (i in seq_len(nrow(exclusion.areas[[2]]))) {
    spec2d[, as.numeric(colnames(spec2d)) > exclusion.areas[[2]][i, 1] &
             as.numeric(colnames(spec2d)) < exclusion.areas[[2]][i, 2]] <- 0
  }
  ## data in longer format
  spec2D_longer <- as.data.frame(spec2d) %>% rownames_to_column(var = "x") %>%
    pivot_longer(!contains("x"), names_to = "y",
                 values_to = "intensity") %>%
    mutate(x = as.numeric(x),
           y = as.numeric(y)) %>%
    filter(intensity > min_val)

  p <- ggplot() +
    geom_contour_filled(data = spec2D_longer, aes(x, y, z = intensity)) +
```

```
    theme_minimal() +
    theme(legend.position='none')

  if (!is.null(peaks)) {
    p <- p + geom_point(data = peaks, aes(x = x, y = y), col = "red",
                        size = 0.5)
  }
  return(p)
}
```

DETECTION DE PICS SUR DES SPECTRES RMN 2D

AVEC LA PERSISTANCE

Importer le script de la persistance

Path <- "…/persistence"

imagepers <- import_from_path("imagepers", path = Path)

```
library(reticulate)

peak_detection <- function(spec2d, level = NULL,
                           exclusion.areas = list(matrix(c(4.5, 5.1), ncol = 2),
                                                  matrix(c(4.5, 5.1), ncol = 2)),
                           max.shift = 0.02, type = c("COSY", "TOCSY", "HSQC")[1]) {
  min_val <- NA

  if(is.null(level)){
    if(type == "TOCSY" | type == "COSY"){min_val <- ((IQR(spec2d)/2)/0.674) * 3}
    else{min_val <- ((IQR(spec2d)/2)/0.674) * 2}
  }
  else{min_val <- ((IQR(spec2d)/2)/0.674) * level}

  ## remove noise and exclusion areas
  spec2d[spec2d <= min_val] <- 0

  # exclude on x-axis
  for (i in seq_len(nrow(exclusion.areas[[1]]))) {
    spec2d[as.numeric(rownames(spec2d)) > exclusion.areas[[1]][i, 1] &
             as.numeric(rownames(spec2d)) < exclusion.areas[[1]][i, 2], ] <- 0
  }
  # exclude on y-axis
  for (i in seq_len(nrow(exclusion.areas[[2]]))) {
    spec2d[, as.numeric(colnames(spec2d)) > exclusion.areas[[2]][i, 1] &
             as.numeric(colnames(spec2d)) < exclusion.areas[[2]][i, 2]] <- 0
  }
  ## peak detection
  g0 <- imagepers$persistence(spec2d)
  persistence <-
    data.frame(t(sapply(g0, function(x) c(x[[1]][[2]] + 1,
                                          x[[1]][[1]] + 1, x[[3]]))))
  colnames(persistence) <- c("x", "y", "pers")
```

```
  persistence <- persistence %>%  filter(pers > 0)
  persistence$x <- as.numeric(colnames(spec2d)[persistence$x])
  persistence$y <- as.numeric(rownames(spec2d)[persistence$y])

  temp <- persistence$x
  persistence$x <- persistence$y
  persistence$y <- temp

  # Suppression des points alignés sur O
  persistence <- persistence[persistence$x > 0.5,]

  if(type == "TOCSY") {
    ## difference between x and y
    persistence$diff <- persistence$x - persistence$y
    # remove peak when x = y
    persistence <- persistence[abs(persistence$diff) > max.shift, ]
  }
  peak.index <- as.numeric(rownames(persistence))
  persistence <- cbind(peak.index, persistence)

  return(persistence)
}
```

AVEC LES MAXIMA LOCAUX

```
peak_detection_localMax <- function(spec2d, level = NULL,
                                    exclusion.areas = list(matrix(c(4.5, 5.1), ncol = 2),
                                                           matrix(c(4.5, 5.1), ncol = 2)),
                                    max.shift = 0.02, type = c("TOCSY", "HSQC")[1]) {
  min_val <- NA

  if(is.null(level)){
    if(type == "TOCSY"){min_val <- ((IQR(spec2d)/2)/0.674) * 3}
    else{min_val <- ((IQR(spec2d)/2)/0.674) * 2}
  }
  else{min_val <- ((IQR(spec2d)/2)/0.674) * level}

  ## remove noise and exclusion areas
  spec2d[spec2d <= min_val] <- 0

  # exclude on x-axis
  for (i in seq_len(nrow(exclusion.areas[[1]]))) {
    spec2d[as.numeric(rownames(spec2d)) > exclusion.areas[[1]][i, 1] &
             as.numeric(rownames(spec2d)) < exclusion.areas[[1]][i, 2], ] <- 0
  }
  # exclude on y-axis
  for (i in seq_len(nrow(exclusion.areas[[2]]))) {
    spec2d[, as.numeric(colnames(spec2d)) > exclusion.areas[[2]][i, 1] &
             as.numeric(colnames(spec2d)) < exclusion.areas[[2]][i, 2]] <- 0
  }

  ## peak detection
  nC <- ncol(spec2d)
  nR <- nrow(spec2d)
```

```
  vMax <- intersect(which(c(NA, spec2d) < c(spec2d, NA)),
                    which(c(NA, spec2d) > c(spec2d, NA)) - 1)
  hMax <- intersect(which(c(NA, t(spec2d)) < c(t(spec2d), NA)),
                    which(c(NA, t(spec2d)) > c(t(spec2d), NA)) - 1) - 1
  hMax <- (hMax %% nC * nR) + hMax %/% nC + 1

  ### Find diagonal maxima
  hvMax <- intersect(vMax, hMax)
  hvMax <- hvMax[spec2d[hvMax] > min_val]

  ident_mat <- matrix(1:(dim(spec2d)[1] * dim(spec2d)[2]),
                      nrow = dim(spec2d)[1], ncol = dim(spec2d)[2])
  colnames(ident_mat) <- colnames(spec2d)
  peaklist <- as.data.frame(ident_mat) %>%
    mutate(x = rownames(spec2d)) %>%
    pivot_longer(!contains("x"), names_to = "y", values_to = "ident") %>%
    mutate(y = as.numeric(y), x = as.numeric(x)) %>% filter(ident %in% hvMax) %>%
    mutate(intensity = as.numeric(spec2d)[ident])

  # Suppression des points alignés sur 0
  peaklist <- peaklist[peaklist$x > 0.5,]

  if(type == "TOCSY") {
    ## difference between x and y
    peaklist$diff <- peaklist$x - peaklist$y
    # remove peak when x = y
    peaklist <- peaklist[abs(peaklist$diff) > max.shift, ]
  }
  peak.index <- as.numeric(rownames(peaklist))
  peaklist <- cbind(peak.index, peaklist)
  return(peaklist)
}
```

TRANSFORMATION D'UNE MATRICE DE SPECTRE SOUS LA FORME (x,y,intensity)

```
longer_2D <- function(spec2d, min_val,
                      exclusion.areas = list(matrix(c(4.5, 5.1), ncol = 2),
                                             matrix(c(4.5, 5.1), ncol = 2)),
                      max.shift = 0.02, type = c("TOCSY", "HSQC")[1]) {
  if(missing(min_val)){
    min_val <- NA
    if(type == "TOCSY"){min_val <- ((IQR(spec2d)/2)/0.674) * 3}
    else{min_val <- ((IQR(spec2d)/2)/0.674) * 2}
  }
  else{min_val <- ((IQR(spec2d)/2)/0.674) * level}

  ## remove noise and exclusion areas
  spec2d[spec2d <= min_val] <- 0

  # exclude on x-axis
  for (i in seq_len(nrow(exclusion.areas[[1]]))) {
    spec2d[as.numeric(rownames(spec2d)) > exclusion.areas[[1]][i, 1] &
             as.numeric(rownames(spec2d)) < exclusion.areas[[1]][i, 2], ] <- 0
  }
```

```
  # exclude on y-axis
  for (i in seq_len(nrow(exclusion.areas[[2]]))) {
    spec2d[, as.numeric(colnames(spec2d)) > exclusion.areas[[2]][i, 1] &
             as.numeric(colnames(spec2d)) < exclusion.areas[[2]][i, 2]] <- 0
  }
  ## data in longer format
  spec2D_longer <- as.data.frame(spec2d) %>%
    rownames_to_column(var = "x") %>%
    pivot_longer(!contains("x"), names_to = "y",
                 values_to = "intensity") %>%
    mutate(x = as.numeric(x),
           y = as.numeric(y)) %>%
    filter(intensity > min_val)

  # Suppression des points alignés sur 0
  spec2D_longer <- spec2D_longer[spec2D_longer$x > 0.5,]

  # Suppression de la diagonale lorsque le spectre est de type TOCSY
  if(type == "TOCSY") {
    spec2D_longer <- spec2D_longer[abs(spec2D_longer$x - spec2D_longer$y) > max.shift,]
  }
  return(spec2D_longer)
}
```

AVEC LE CLUSTERING HIERARCHIQUE

```
hclust_detection <- function(spec2d, haut, level = NULL,
                             exclusion.areas = list(matrix(c(4.5, 5.1), ncol = 2),
                                                    matrix(c(4.5, 5.1), ncol = 2)),
                             max.shift = 0.02, type = c("TOCSY", "HSQC")[1]) {
  if(missing(haut)){haut = 0.08}
  min_val <- NA
  if(is.null(level)){
    if(type == "TOCSY"){min_val <- ((IQR(spec2d)/2)/0.674) * 3}
    else{min_val <- ((IQR(spec2d)/2)/0.674) * 2}
  }
  else{min_val <- ((IQR(spec2d)/2)/0.674) * level}

  spec2D_longer <- longer_2D(spec2d, min_val, exclusion.areas, max.shift, type)

  # Suppression de la diagonale lorsque le spectre est de type TOCSY

  if(type == "TOCSY") {
    spec2D_longer <- spec2D_longer[abs(spec2D_longer$x - spec2D_longer$y) > max.shift,]
  }
  spec2D_long <- spec2D_longer[,1:2]
  hc <- hclust(dist(spec2D_long), method = "complete")
  clusters <- cutree(hc, h = haut)
  cluster_centers <- aggregate(spec2D_longer, list(clusters), mean)

  # Suppression des points alignés sur 0
  cluster_centers <- cluster_centers[cluster_centers$x > 0.5,]
  return(cluster_centers)
}
```

AVEC DBSCAN

```
library(dbscan)

dbscan_detection <- function(spec2d, epsilon, MinPts, level = NULL,
                             exclusion.areas = list(matrix(c(4.5, 5.1), ncol = 2),
                                                    matrix(c(4.5, 5.1), ncol = 2)),
                             max.shift = 0.02, type = c("TOCSY", "HSQC")[1]) {

  if(missing(epsilon)){epsilon = 0.013}
  if(missing(MinPts)){MinPts = 1}

  min_val <- NA

  if(is.null(level)){
    if(type == "TOCSY"){min_val <- ((IQR(spec2d)/2)/0.674) * 3}
    else{min_val <- ((IQR(spec2d)/2)/0.674) * 2}
  }
  else{min_val <- ((IQR(spec2d)/2)/0.674) * level}

  spec2D_longer <- longer_2D(spec2d, min_val, exclusion.areas, max.shift, type)

  # Suppression de la diagonale lorsque le spectre est de type TOCSY
  if(type == "TOCSY") {
    spec2D_longer <- spec2D_longer[abs(spec2D_longer$x - spec2D_longer$y) > max.shift,]
  }
  spec2D_long <- spec2D_longer[,1:2]
  db <- dbscan(spec2D_long, eps = epsilon, minPts = MinPts)
  cluster_centers <- aggregate(spec2D_longer, list(db$cluster), mean)

  return(cluster_centers)
}
```

ANNOTATION SPECTRE 2D MATRICE COMPLEXE BASEE SUR UNE SEQUENCE RMN

matriceComplexe : data.frame liste couples ppm de la matrice a annoter
BdDStandards : objet contenant la base de donnees des composes standards
nom_sequence : nom sequence 2D a utiliser pour annotation (“JRES”,“COSY”,“TOCSY”,“HMBC”,“HSQC”)
ppm1Tol : tolerance ppm axe abscisses
ppm2Tol : tolerance ppm axe ordonnees
nb_ligne_template : preciser le nombre total de ligne de la feuille de calcul ? annoter
seuil : valeur du score de presence en deça de laquelle les metabolites annotés ne sont pas retenus
unicite : boolean pour ne retenir que les métabolites qui n’ont pas de pics en communs

```
annotationRmn2Dr <- function(matriceComplexe, BdDStandard, Liste = NULL, L = FALSE,
                             nom_sequence, ppm1Tol=0.01, ppm2Tol=0.01,
                             seuil=0, unicite="NO")
{
  ## Longueur de la peak-list de la matrice a annoter
  PeakListLength <- length(matriceComplexe[, 1])

  BdDStandards = NA
  if(L == FALSE){BdDStandards = BdDStandard}
```

```
  else{BdDStandards <- BdDStandard[which(names(BdDStandard) %in% Liste)]}

  ## Nombre de metabolites inclus dans BdD de composes standards
  nbMetabolitesBdD <- length(BdDStandards)
  matrixAnnotation <- data.frame()
  allMetabolitesList <- data.frame()
  seuil_score <- seuil

  ## Boucle sur les metabolites inclus dans BdD
  for (i in 1:nbMetabolitesBdD)
  {
    ## Infos metabolite en cours
    iMetabolite <- BdDStandards[[i]]
    ppm1M <- iMetabolite[,1]
    ppm2M <- iMetabolite[,2]
    nbPeakMetabolite <- length(ppm1M)
    MetaboliteName <- names(BdDStandards[i])
  ##    print(MetaboliteName)
    ## Initialisation
    k <- 0
    presenceScore <- 0
    annotatedPpmRef <- data.frame()
    annotatedPpmList <- data.frame()
    annotatedPeakLength <- 0
    metabolites <- data.frame()
    metabolitesList <- data.frame()

    ## Boucle sur les couples de pics de la matrice a annoter
    for (p in 1:PeakListLength)
    {
      ppmAnnotationF1 <- as.numeric(matriceComplexe[p, 3])
      ppmAnnotationF2 <- as.numeric(matriceComplexe[p, 2])
      e <- simpleMessage("end of file")
      tryCatch({
        if (!is.na(ppmAnnotationF1))
        {
          matrixAnnotation <- unique.data.frame(rbind.data.frame(matrixAnnotation,
      matriceComplexe[p, ]))
        }
        # Recherche du couple de pics de la matrice la liste des couples du metabolite st
andard
        metaboliteIn <- (ppm1M >= (ppmAnnotationF2-ppm1Tol) &
      ppm1M <= (ppmAnnotationF2+ppm1Tol) &
      ppm2M >= (ppmAnnotationF1-ppm2Tol) &
      ppm2M <= (ppmAnnotationF1+ppm2Tol))
        WhichMetaboliteIn <- which(metaboliteIn)
        # Si au moins un couple de la matrice a annoter dans liste couples metabolite sta
ndard
        if (length(WhichMetaboliteIn) > 0)
        {
          for (a in 1:length(WhichMetaboliteIn))
          {
            annotatedPpmList <- data.frame(ppm1=ppm1M[WhichMetaboliteIn[a]],
                                           ppm2=ppm2M[WhichMetaboliteIn[a]],
```

```
                                                theoricalLength=nbPeakMetabolite)
            annotatedPpmRef <- rbind(annotatedPpmRef,annotatedPpmList)
          }
        }
      }, error=function(e){cat ("End of file \n");})
    }

    # Au - 1 couple de ppm de la matrice complexe annote
    if (nrow(annotatedPpmRef) >= 1)
    {
      ## Nombre couples annotes
      annotatedPeakLength <- nrow(annotatedPpmRef)

      ## Recherche doublons
      annotatedDoublons <- duplicated(annotatedPpmRef)
      if (sum(duplicated(annotatedPpmRef)) > 0)
      {
        annotatedPeakLength <- nrow(annotatedPpmRef) - sum(duplicated(annotatedPpmRef))
        annotatedPpmRef <- annotatedPpmRef[-duplicated(annotatedPpmRef), ]
      }
      presenceScore <- round(annotatedPeakLength/nbPeakMetabolite, 4)
    }
    ## Conservation metabolites dont score > seuil
    if (presenceScore > seuil_score)
    {
      metabolites <- data.frame(Metabolite=MetaboliteName, score=presenceScore)
      metabolitesList <- cbind.data.frame(annotatedPpmRef, metabolites)
      allMetabolitesList <- rbind.data.frame(allMetabolitesList, metabolitesList)
    }
  }
  # Initialisation
  commonPpm <- data.frame()
  commonPpmList <- data.frame()
  metaboliteAdd <- data.frame()
  metaboliteAddList <- data.frame()
#  metabolite_ref <- data.frame()
  commonMetabolitesList <- data.frame()
  commonMetabolitesPpmList <- data.frame()
  commonMetabolitesPpmAllList1 <- data.frame()
  commonMetabolitesPpmAllList <- data.frame()
  listeTotale_2D_unicite <- allMetabolitesList[, 1:4]
  allMetabolitesList <- allMetabolitesList[, -3]
  metabolitesAllUnicite <- data.frame()

  ## Boucle sur tous couples annotes
  for (j in 1:length(allMetabolitesList$ppm1))
  {
    ## Boucle sur metabolites dans BdD composes standards
    for (i in 1:nbMetabolitesBdD)
    {
      ppmMetaboliteBdD <- BdDStandards[[i]]
      ppm1M <- ppmMetaboliteBdD[,1]
      ppm2M <- ppmMetaboliteBdD[,2]
      # Nombre de couples metabolite
```

```
      nbPeakMetabolite <- length(ppm1M)
      MetaboliteName <- names(BdDStandards[i])

      metabolitesInAll <- (ppm1M >= (allMetabolitesList[j,1]-ppm1Tol) &
                             ppm1M <= (allMetabolitesList[j,1]+ppm1Tol) &
                             ppm2M >= (allMetabolitesList[j,2]-ppm2Tol) &
                             ppm2M <= (allMetabolitesList[j,2]+ppm2Tol))
      WhichMetabolitesInAll <- which(metabolitesInAll)

      if (MetaboliteName != allMetabolitesList[j, 3] & length(WhichMetabolitesInAll) > 0)
      {
        metabolitesAllUnicite <- rbind.data.frame(metabolitesAllUnicite, listeTotale_2D_u
nicite[j,])
        commonPpm <- data.frame(ppm1=allMetabolitesList[j,1], ppm2=allMetabolitesList[j,2
])
        commonPpmList <- rbind.data.frame(commonPpmList, commonPpm)
        commonPpmList <- unique(commonPpmList)
        metaboliteAdd <- data.frame(nom_metabolite=MetaboliteName)
        metaboliteAddList <- rbind.data.frame(metaboliteAddList, metaboliteAdd)
#        metabolite_ref <- data.frame(nom_metabolite=allMetabolitesList[j,3])
        commonMetabolitesList <- rbind.data.frame(data.frame(nom_metabolite=allMetabolite
sList[j, 3]),
                                                  metaboliteAddList)
        commonMetabolitesPpmList <- cbind.data.frame(commonPpm, commonMetabolitesList)
        commonMetabolitesPpmAllList1 <- rbind.data.frame(commonMetabolitesPpmAllList1,
                                                         commonMetabolitesPpmList)
        commonMetabolitesPpmAllList1 <- unique.data.frame(commonMetabolitesPpmAllList1)
      }
    }
    commonMetabolitesPpmAllList <- rbind.data.frame(commonMetabolitesPpmAllList,
                                                    commonMetabolitesPpmAllList1)
    commonMetabolitesPpmAllList <- unique.data.frame(commonMetabolitesPpmAllList)

    #initialisation des data.frame
    commonPpm <- data.frame()
    metaboliteAdd <- data.frame()
    metaboliteAddList <- data.frame()
    metabolite_ref <- data.frame()
    commonMetabolitesList <- data.frame()
    commonMetabolitesPpmList <- data.frame()
    commonMetabolitesPpmAllList1 <- data.frame()
  }

  unicityAllList <- listeTotale_2D_unicite
  if (nrow(listeTotale_2D_unicite)!=0 & nrow(metabolitesAllUnicite)!=0)
    unicityAllList <- setdiff(listeTotale_2D_unicite, metabolitesAllUnicite)

  unicitynbCouplesRectif <- data.frame()
  for (g in 1:nrow(unicityAllList))
  {
    metaboliteUnicity <- (unicityAllList$Metabolite == unicityAllList$Metabolite[g])
    WhichMetaboliteUnicity <- which(metaboliteUnicity)
    nb_occurence <- length(WhichMetaboliteUnicity)
    unicitynbCouplesRectif <- rbind.data.frame(unicitynbCouplesRectif, nb_occurence)
```

```
  }
  names(unicitynbCouplesRectif) <- "NbCouplesAnnotes"
  unicityAllList <- cbind.data.frame(unicityAllList, unicitynbCouplesRectif)

  unicityAllList <- cbind.data.frame(unicityAllList,
                                     score_unicite=unicityAllList$NbCouplesAnnotes/unicit
yAllList$theoricalLength)
  unicityAllList <- unicityAllList[, -3]
  unicityAllList <- unicityAllList[, -4]

##  unicityAllList <- filter(unicityAllList, unicityAllList$score_unicite > seuil_score)
  unicityAllList <- unicityAllList[unicityAllList$score_unicite > seuil_score,]

  listeTotale_metabo <- data.frame()
  if (nrow(commonPpmList) !=0)
  {
    for (o in 1:length(commonPpmList[, 1]))
    {
      tf6 <- (commonMetabolitesPpmAllList$ppm1 == commonPpmList[o,1] &
                commonMetabolitesPpmAllList$ppm2 == commonPpmList[o,2])
      w6 <- which(tf6)

      for (s in 1:length(w6))
      {
        metaboliteAdd <- data.frame(nom_metabolite=commonMetabolitesPpmAllList[w6[s],3])
        commonMetabolitesList <- paste(commonMetabolitesList, metaboliteAdd[1,], sep=" ")
      }
      liste_metabo_ppm<-cbind.data.frame(ppm1=commonPpmList[o,1],ppm2=commonPpmList[o,2],
                                         commonMetabolitesList)
      listeTotale_metabo <- rbind.data.frame(listeTotale_metabo, liste_metabo_ppm)
      commonMetabolitesList <- data.frame()
    }
  }

  # Representation graphique
  if (nom_sequence == "HSQC" | nom_sequence == "HMBC")
  {
    atome <- "13C"
    indice_positif <- 1
    indice_negatif <- -10
  }else{
    atome <- "1H"
    indice_positif <- 0.5
    indice_negatif <- -0.5
  }

  matriceComplexe <- matrixAnnotation
  ppm1 <- as.numeric(matriceComplexe[,2])
  ppm2 <- as.numeric(matriceComplexe[,3])

  if (unicite == "NO")
  {
    listeTotale_2D_a_utiliser <- allMetabolitesList
    d1.ppm <- allMetabolitesList$ppm1
```

```
    d2.ppm <- allMetabolitesList$ppm2
  }else{
    listeTotale_2D_a_utiliser <- unicityAllList
    d1.ppm <- listeTotale_2D_a_utiliser$ppm1
    d2.ppm <- listeTotale_2D_a_utiliser$ppm2
  }

  if (nrow(listeTotale_2D_a_utiliser) > 0)
  {
    ## Taches de correlations
    # Matrice biologique + Annotations
    maxX <- max(round(max(as.numeric(matriceComplexe[,2])))+0.5,
                round(max(as.numeric(matriceComplexe[,2]))))
    maxY <- max(round(max(as.numeric(matriceComplexe[,3])))+indice_positif,
                round(max(as.numeric(matriceComplexe[,3]))))
    probability.score <- as.factor(round(listeTotale_2D_a_utiliser[,4],2))
    lgr <- length(unique(probability.score))
    ## X11()
    sp <- ggplot(matriceComplexe, aes(x=ppm1, y=ppm2))
    sp <- sp + geom_point(size=2) + scale_x_reverse(breaks=seq(maxX, 0, -0.5)) +
      scale_y_reverse(breaks=seq(maxY, 0, indice_negatif)) +
      xlab("1H chemical shift (ppm)") +
      ylab(paste(atome, " chemical shift (ppm)")) +
      ggtitle(nom_sequence) +
      geom_text(data=listeTotale_2D_a_utiliser,
                aes(d1.ppm, d2.ppm,
                label=str_to_lower(substr(listeTotale_2D_a_utiliser[,3],1,3)),
                                                col=probability.score),
                size=6, hjust=0, nudge_x=0.02, vjust=0, nudge_y=0.2) +
      scale_colour_manual(values=viridis(lgr)) +
      theme(axis.title.x=element_text(size=16), axis.title.y=element_text(size=16)) +
      theme(legend.text=element_text(size=16), legend.title=element_text(size=16))
##      scale_color_colormap('Annotation', discrete=T, reverse=T)
    print(sp)
  }

  # Liste des résultats (couples pmm / metabolite / score) + liste ppms metabolites commu
ns
  if (unicite == "NO")
  {
    return(list(liste_resultat=allMetabolitesList, listing_ppm_commun=listeTotale_metabo)
)
  }else{
    return(list(liste_resultat_unicite=unicityAllList, listing_ppm_commun_affichage=liste
Totale_metabo))
  }
}
```

Suppression des zones sans intérêt après la détection des pics

```
nettoyage <- function(peaklist, type){
  if(type == "COSY"){
    peaklist <- subset(peaklist, !(peaklist$x < 0.7))
    peaklist <- subset(peaklist, !(peaklist$y < 0.7))
    peaklist <- subset(peaklist, !(peaklist$x > 8.05))
    peaklist <- subset(peaklist, !(peaklist$y > 8.05))
    peaklist <- subset(peaklist, !(peaklist$x > 5.1 & peaklist$y < 3))
    peaklist <- subset(peaklist, !(peaklist$y > 5.1 & peaklist$x < 3))
    peaklist <- subset(peaklist, !(peaklist$x > 4.2 & peaklist$x < 6.7 & peaklist$y>5.1))
    peaklist <- subset(peaklist, !(peaklist$y > 4.2 & peaklist$y < 6.7 & peaklist$x>5.1))
    peaklist <- subset(peaklist, !(peaklist$x > 5.4 & peaklist$x < 6.7 & peaklist$y<4.5))
    peaklist <- subset(peaklist, !(peaklist$y > 5.4 & peaklist$y < 6.7 & peaklist$x<4.5))
  }
  else if(type == "TOCSY"){
    peaklist <- subset(peaklist, !(peaklist$x < 0.9 | peaklist$y < 0.9 |
                         peaklist$y > 8.1 | peaklist$x > 8.1))
    peaklist <- subset(peaklist, !(peaklist$x < 3 & peaklist$y > 5.2))
    peaklist <- subset(peaklist, !(peaklist$x > 5.1 & peaklist$y < 3))
  }
  else if(type == "HSQC"){
    peaklist <- subset(peaklist, !(peaklist$x < 0.78))
    peaklist <- subset(peaklist, !(peaklist$y < 11.2))
    peaklist <- subset(peaklist, !(peaklist$x > 8.5))
    peaklist <- subset(peaklist, !(peaklist$y > 140))
    peaklist <- subset(peaklist, !(peaklist$x > 5.1 & peaklist$y < 90))
    peaklist <- subset(peaklist, !(peaklist$x > 3.6 & peaklist$y < 50))
    peaklist <- subset(peaklist, !(peaklist$x > 5.3 & peaklist$y < 110))
    peaklist <- subset(peaklist, !(peaklist$x < 4.5 & peaklist$y > 85))
    peaklist <- subset(peaklist, !(peaklist$x < 2.5 & peaklist$y < 85 & peaklist$y > 50))
    peaklist <- subset(peaklist, !(peaklist$x > 5.1 & peaklist$x < 6.7 & peaklist$y>110))
  }
  return(peaklist)
}
```

Combinaison des sequences

Annotation d'une seule séquence

```
annotation <- function(spect, liste_1D, type = c("COSY", "TOCSY", "HSQC"),
                       level = NULL, shift = FALSE, net = FALSE,
                       ppm1Tol=tolPpm1, ppm2Tol=tolPpm1){

  if(shift == TRUE){
    if(type == "COSY" | type == "TOCSY"){
      rownames(spect) <- as.numeric(rownames(spect)) + 0.1
      colnames(spect) <- as.numeric(colnames(spect)) + 0.1
```

```
    }
    else if(type == "HSQC"){
      rownames(spect) <- as.numeric(rownames(spect)) + 0.1
      colnames(spect) <- as.numeric(colnames(spect)) + 3
    }
  }
  peaklist <- peak_detection(spect, level, type = type)
  if(net == TRUE){
    peaklist <- nettoyage(peaklist, type)
  }

  BdDReference <- NA
  if(type == "COSY"){BdDReference = BdDReference_TOCSY}
  else if(type == "TOCSY"){BdDReference = BdDReference_TOCSY}
  else{BdDReference = BdDReference_HSQC}

  annotat <- annotationRmn2Dr(peaklist, BdDReference, Liste = liste_1D,
                              L = TRUE, type, ppm1Tol, ppm2Tol, seuil=0, unicite="NO")
  return(annotat)
}
```

Annotation de toutes les sequences à la fois

```
annotationGlob <- function(spect_1 = NULL, spect_2 = NULL, spect_3, liste_1D,
                           cosy = 1, tocsy = 1, hsqc = 1,
                           tolPpm1 = 0.01, tolPpm2C = 0.5,
                           shift = FALSE, net = TRUE){
  if(cosy == 1){
    res_annC <- annotation(spect_1, liste_1D = liste_1D, type = "COSY", net = net,
                           ppm1Tol=tolPpm1, ppm2Tol=tolPpm1)
  }
  if(tocsy == 1){
    res_annT <- annotation(spect_2, liste_1D = liste_1D, type = "TOCSY", net = net,
                           ppm1Tol=tolPpm1, ppm2Tol=tolPpm1)
  }
  if(hsqc == 1){
    res_annH <- annotation(spect_3, liste_1D = liste_1D, type = "HSQC", net = net,
                           ppm1Tol=tolPpm1, ppm2Tol=tolPpm2C)
  }

  if(cosy == 1 & tocsy == 0){
    return(list(COSY=res_annC$liste_resultat, HSQC=res_annH$liste_resultat))
  }
  else if(cosy == 0 & tocsy == 1){
    return(list(TOCSY=res_annT$liste_resultat, HSQC=res_annH$liste_resultat))
  }
  else if(cosy == 1 & tocsy == 1){
    return(list(COSY=res_annC$liste_resultat, TOCSY=res_annT$liste_resultat, HSQC=res_ann
H$liste_resultat))
  }
}
```

Tableau des scores

```
Tab_scores <- function(List_ann, Liste_1D, cosy = 1, tocsy = 1, hsqc = 1){
  if(cosy == 1){
    Metabolites_COSY <- List_ann$COSY$Metabolite
    Scores_COSY <- List_ann$COSY$score
    datNC <- data.frame(Metabolites_COSY, Scores_COSY)
    datNC <- subset(datNC, !duplicated(Metabolites_COSY))
  }
  if(tocsy == 1){
    Metabolites_TOCSY <- List_ann$TOCSY$Metabolite
    Scores_TOCSY <- List_ann$TOCSY$score
    datNT <- data.frame(Metabolites_TOCSY, Scores_TOCSY)
    datNT <- subset(datNT, !duplicated(Metabolites_TOCSY))
  }
  if(hsqc == 1){
    Metabolites_HSQC <- List_ann$HSQC$Metabolite
    Scores_HSQC <- List_ann$HSQC$score
    datNH <- data.frame(Metabolites_HSQC, Scores_HSQC)
    datNH <- subset(datNH, !duplicated(Metabolites_HSQC))
  }

  Score_COSY <- rep(0, length(Liste_1D))
  Score_TOCSY <- rep(0, length(Liste_1D))
  Score_HSQC <- rep(0, length(Liste_1D))
  Score_moyen <- rep(0, length(Liste_1D))
  Score_max <- rep(0, length(Liste_1D))

  if(cosy == 1 & tocsy == 0){
    for(i in 1:length(Liste_1D)){
      if(Liste_1D[i] %in% unique(Metabolites_COSY)){
        Score_COSY[i] <- datNC$Scores_COSY[which(datNC$Metabolites_COSY == Liste_1D[i])]
      }
      if(ListeN[i] %in% unique(Metabolites_HSQC)){
        Score_HSQC[i] <- datNH$Scores_HSQC[which(datNH$Metabolites_HSQC == Liste_1D[i])]
      }
      Score_moyen[i] <- mean(c(Score_COSY[i], Score_HSQC[i]))
      Score_max[i] <- max(c(Score_COSY[i], Score_HSQC[i]))
    }
    Tab_10 <- data.frame(Metabolites = Liste_1D, Score_COSY, Score_HSQC, Score_moyen, Sco
re_max)
    return(Tab_10)
  }
  else if(cosy == 0 & tocsy == 1){
    for(i in 1:length(Liste_1D)){
      if(Liste_1D[i] %in% unique(Metabolites_TOCSY)){
        Score_TOCSY[i] <- datNT$Scores_TOCSY[which(datNT$Metabolites_TOCSY == Liste_1D[i]
)]
      }
      if(ListeN[i] %in% unique(Metabolites_HSQC)){
        Score_HSQC[i] <- datNH$Scores_HSQC[which(datNH$Metabolites_HSQC == Liste_1D[i])]
      }
      Score_moyen[i] <- mean(c(Score_TOCSY[i], Score_HSQC[i]))
```

```
      Score_max[i] <- max(c(Score_TOCSY[i], Score_HSQC[i]))
    }
    Tab_01 <- data.frame(Metabolites = Liste_1D, Score_TOCSY, Score_HSQC, Score_moyen, Sc
ore_max)
    return(Tab_01)
  }
  else if(cosy == 1 & tocsy == 1){
    for(i in 1:length(Liste_1D)){
      if(Liste_1D[i] %in% unique(Metabolites_COSY)){
        Score_COSY[i] <- datNC$Scores_COSY[which(datNC$Metabolites_COSY == Liste_1D[i])]
      }
      if(Liste_1D[i] %in% unique(Metabolites_TOCSY)){
        Score_TOCSY[i] <- datNT$Scores_TOCSY[which(datNT$Metabolites_TOCSY == Liste_1D[i]
)]
      }
      if(ListeN[i] %in% unique(Metabolites_HSQC)){
        Score_HSQC[i] <- datNH$Scores_HSQC[which(datNH$Metabolites_HSQC == Liste_1D[i])]
      }
      Score_moyen[i] <- mean(c(Score_COSY[i], Score_TOCSY[i], Score_HSQC[i]))
      Score_max[i] <- max(c(Score_COSY[i], Score_TOCSY[i], Score_HSQC[i]))
    }
    Tab_11 <- data.frame(Metabolites = Liste_1D, Score_COSY, Score_TOCSY, Score_HSQC, Sco
re_moyen, Score_max)
    return(Tab_11)
  }
}
```

Combinaison des séquences avec le score maximal et le seuil fixé à 0.1

```
annotation_globale <- function(spect_COSY = NULL, spect_TOCSY = NULL, spect_HSQC,
                               liste_1D, cosy = 1, tocsy = 1, hsqc = 1, seuil,
                                tolPpm1 = 0.01, tolPpm2C = 0.5,
                               shift = FALSE, net = TRUE){


  if(missing(seuil)){seuil = 0.1}


  List_ann <- annotationGlob(spect_1 = spect_COSY, spect_2 = spect_TOCSY,
                             spect_3 = spect_HSQC, liste_1D, cosy = cosy,
                             tocsy = ocsy, hsqc = hsqc, tolPpm1 = tolPpm1,
                             tolPpm2C = tolPpm2C, shift = shift, net = net)

  tab <- Tab_scores(List_ann, Liste_1D, cosy = cosy, tocsy = ocsy, hsqc = hsqc)

  quant <- tab$Metabolites[which(tab$Score_max > seuil)]

  return(quant)
}
```

Calculs des vrais et faux positifs après annotation

```
metabo_1D <- function(Vrais, quantif){
  L <- quantif[which(quantif %in% Vrais)]
  u <- quantif[-which(quantif %in% Vrais)]
  return(list(detect = unique(quantif), VP = unique(L), FP = unique(u)))
}

metabo_2D <- function(Vrais, liste_1D, quantif){
  L <- quantif[which(quantif %in% Vrais)]
  u <- quantif[-which(quantif %in% Vrais)]
  nd <- liste_1D[-which(liste_1D %in% quantif)]
  fn <- nd[which(nd %in% Vrais)]
  vn <- nd[-which(nd %in% Vrais)]
  return(list(detect = unique(quantif), VP = unique(L), FP = unique(u), VN = unique(vn),
FN = unique(fn)))
}
```