Découverte de modifications chimiques portées par les protéines : identification rapide de jeux de spectres de masse sans filtre de masse

Matthieu David

Découverte de modifications chimiques portées par les protéines : identification rapide de jeux de spectres de masse sans filtre de masse

Matthieu David

UNIVERSITE BRETAGNE LOIRE MATHSTIC

UNIVERSITÉ DE NANTES

THÈSE DE DOCTORAT DE

L'UNIVERSITE DE NANTES
COMUE **U**NIVERSITE **B**RETAGNE **L**OIRE

Ecole Doctorale N°601
Mathèmatique et Sciences et Technologies de l'Information et de la Communication
Spécialité : *Informatique*

Par

« Matthieu DAVID »

« Découverte de modifications chimiques portées par les protéines : identification rapide de jeux de spectres de masse sans filtre de masse »

« »

Thèse présentée et soutenue à NANTES , le 24/10/2019
Unité de recherche : LS2N (Université de Nantes), BIA (Institut National de Recherche Agronomique
Thèse N° :

Rapporteurs avant soutenance :

Jean-François GIBRAT, Directeur de Recherche, Institut National de Recherche Agronomique

Frédérique LISACEK, Directrice de Recherche, Swiss Institute of Bioinformatics

Composition du jury :

Attention, en cas d'absence d'un des membres du Jury le jour de la soutenance, la composition ne comprend que les membres présents

Président :

Examinateurs : Alain DENISE, Professeur, Université Paris-Sud

Yves VANDENBROUCK, Principal Investigator, Commissariat à l'Energie Atomique et aux Energies Alternatives

Dir. de thèse : Guillaume FERTIN, Professeur, Université de Nantes

Co-dir. de thèse : Dominique TESSIER, Ingénieure de Recherche, Institut National de Recherche Agronomique

Invité(s)

Hélène ROGNIAUX, Ingénieure de Recherche, Institut National de Recherche Agronomique

Remerciements

Réaliser une thèse est un travail de longue haleine, qui ne s'achève parfois pas avec le départ du doctorant. C'est en revanche toujours un travail qui, non content de durer quelques trois années, commence bien en amont. Thématique, financement, supervision...

Pour toutes ces heures d'investissement professionnel, et parfois personnel, je tiens à remercier du fond du cœur Dominique et Guillaume, mes encadrants. Les remercier aussi de m'avoir supporté, de temps à autre porté, pour la réussite de cette thèse. Merci aussi pour nos échanges d'idées enrichissants, ainsi que les moments quotidiens conviviaux autour d'un café.

Au titre de ces deux derniers points, je tiens également a exprimer toute ma gratitude aux membres des deux laboratoires d'accueil qu'ont été le LS2N et l'unité BIA, dont les membres ont contribué à faire de ces quelques années de travail des années de vie communautaire. Remerciements tout particuliers à leurs secrétaires, qui réalisent, toujours avec le sourire, un travail de titan.

Merci à tous les membres et organisateurs des évènements EUBIC, qui m'ont permis de nombreux et précieux moments d'échange et de découverte. Merci aux membres de GRIOTE, pour avoir cru en ce projet, et sans qui cette thèse n'aurait sûrement pas vu le jour.

Je tiens à remercier également Stéphane VIALETTE et Yves VANDENBROUCK pour avoir accepté de jeter un œil à mon travail ne de multiples occasions et d'avoir fourni de précieux conseils et idées d'amélioration. Un immense merci à tous les membres du jury pour le temps investi à diverses étapes de la validation de ce travail : Jean-François GIBRAT, Frédérique LISACEK, Alain DENISE et Yves VANDENBROUCK. Je remercie très chaleureusement Hélène, qui à plusieurs occasions a réussi à ménager pour moi du temps libre dans son agenda de ministre, effort qui m'a profité professionnellement, et touché personnellement.

Cette thèse a également été pour moi l'occasion d'enseigner à l'Université, et je remercie les membres de l'équipe enseignante concernés. Mention toute particulière à Frederic GOUALARD et Christophe JERMANN, qui en ont fait une expérience inoubliable.

Une thèse n'est pas seulement un travail de recherche et d'enseignement, mais aussi une période de découverte personnelle et de travail sur soi. Merci à Tim URBAN, qui ne lira probablement jamais ces lignes, car il a bien des choses plus importantes à faire, et est vraisemblablement en train d'en faire d'autres. Merci d'avoir formalisé pour moi des concepts importants.

Je ne peux clore cette section sans nommer deux personnes supplémentaires. Remerciements très chaleureux à Camille, qui a partagé mon bureau au LS2N. Nombre de nos échanges me manquent, et je n'ai pu reproduire notre cérémonie du thé quotidienne. Merci à ma maman, qui a manifestement accompli haut la main la tache, parfois ingrate, de m'élever, et qui est toujours là dans les moments difficiles ou importants.

Enfin, merci infiniment à ma moitié, qui a très certainement profité autant, sinon plus que moi, des hauts et bas, déprimes, et moments Eurêka de ce travail.

Table des matières

1

Introduction

Les *protéines* sont des molécules de grande taille, aussi appelées macromolécules, dont la découverte est actée pour la première fois au 19^{e} siècle [101]. Les protéines sont constituées de composés organiques nommés *acides aminés*, seconds constituants des êtres humains derrière l'eau [138]. Les protéines sont présentes dans la totalité des cellules et des tissus des organismes vivants. Au sein de ces cellules et tissus, elles remplissent une grande variété de fonctions indispensables à la survie et au développement de l'organisme. Certaines protéines réalisent par exemple la catalyse de réactions chimiques et permettent ainsi d'assurer le métabolisme des cellules par la synthèse et la dégradation de composés chimiques (rôle enzymatique rempli par exemple par l'exokinase dans la réaction de glycolyse [83]). D'autres protéines remplissent un rôle de maintien structurel des cellules et des tissus (par exemple l'actine [112]) et permettent une organisation spatiale de l'organisme. D'autres encore assurent des fonctions de mobilité de l'organisme (par exemple la myosine [112], que l'on retrouve principalement dans les muscles où elle participe à la contraction de ces derniers), ou bien de stockage d'acides aminés destinés à la synthèse de protéines (comme c'est par exemple le cas de l'ovalbumine, protéine majoritaire du blanc d'œuf [18]). Ces exemples ne représentent qu'une petite partie des fonctions remplies par les protéines, lesquelles s'imposent donc en participants indispensables de la machinerie du vivant. Les fonctions de ces protéines, si elles dépendent majoritairement de leur séquence en acides aminés, peuvent aussi être dépendantes de modifications chimiques qui se fixent sur elles. C'est par exemple le cas de l'hème, un groupement chimique qui se fixe à l'hémoglobine et permet à cette protéine de transporter l'oxygène dans le sang [56]. Certaines fonctions peuvent également être remplies non pas par une unique protéine, mais par un *complexe protéique*, lequel résulte de l'assemblage de plusieurs protéines. L'étude et la compréhension des protéines et des modifications chimiques qui peuvent s'y fixer revêtent donc une importance cruciale pour rendre intelligible le fonctionnement des organismes vivants dans ses détails.

Par analogie avec la génomique qui étudie les gènes, l'étude des protéines est appelée protéomique. Ce domaine d'étude qui vise à identifier, quantifier et caractériser les protéines à grande échelle s'avère cependant beaucoup plus complexe. En effet, contrairement à l'information génomique, le protéome d'un organisme est dynamique : il évolue continuellement au cours du temps. De plus, un même gène peut coder pour plusieurs protéines et une très grande variété de modifications est susceptible d'enrichir dynamiquement ce répertoire déjà très large. Enfin, contrairement à la génomique, la protéomique ne dispose pas de technique expérimentale de duplication des protéines, ce qui rend l'accès aux protéines minoritaires très compliqué.

La protéomique peut s'appuyer sur différentes techniques d'analyse comme par exemple la cristallogra-

phie aux rayons X ou la résonance magnétique nucléaire. Toutefois, l'approche protéomique dite "bottom-up" basée sur de la chromatographie liquide couplée à de la spectrométrie de masse en tandem est actuellement la technique la plus utilisée car elle permet un grand débit d'analyse. Dans cette approche, les protéines sont d'abord découpées en molécules plus petites (les peptides) grâce à l'action d'une enzyme. Ces peptides sont ensuite élués avant d'entrer dans un spectromètre de masse où ils sont soumis à deux étapes de mesures de masse (appelées MS et MS/MS) séparées par une étape de fragmentation. Les spectres expérimentaux, construits à partir des intensités et des masses mesurées sur les fragments peptidiques en MS/MS, sont utilisés pour identifier les peptides à l'origine des spectres, puis pour inférer les protéines présentes dans le mélange analysé. Dans le cadre de cette thèse, nous nous sommes focalisés sur cette dernière étape d'interprétation des spectres.

Déterminer la séquence d'acides aminés associée à un spectre de masse a tout d'abord été une tâche manuelle complexe et fastidieuse, réalisée par des experts. L'introduction de l'outil informatique dans le domaine de l'analyse des données produites par les spectromètres de masse a permis d'accélérer significativement (de parfois plusieurs mois à quelques minutes) le temps de traitement des données. Cependant, malgré des spectromètres de masse de plus en plus performants, à l'issue de l'étape d'interprétation des données, il reste encore aujourd'hui une forte proportion de spectres (parfois plus de la moitié) qui n'ont pas été interprétés de manière fiable sous la forme d'une séquence d'acides aminés. La complexité algorithmique de l'interprétation provient de la combinaison de plusieurs facteurs : la volumétrie des données à traiter (plusieurs dizaines de milliers de spectres sont générés dans une analyse de quelques heures), la présence des modifications chimiques portées par les peptides et *a priori* inconnues qui modifient leurs masses, l'existence de spectres portant simultanément les informations de plusieurs peptides, la difficulté de l'étape d'inférence des protéines liée aux peptides partagés. Notons de plus que dans le cas des organismes végétaux, sujet d'intérêt pour l'INRA, les séquences d'acides aminés qui constituent les protéines ne sont pas toujours parfaitement connues.

Avec l'accroissement récent de la puissance de calcul des ordinateurs et l'amélioration des techniques de spectrométrie de masse, il nous a paru possible de développer de nouvelles approches algorithmiques moins sensibles à la présence de modifications et qui permettent ainsi une meilleure interprétation des spectres. Nous proposons dans ce manuscrit un nouvel algorithme pour l'identification de spectres issus d'expériences de spectrométrie de masse en comparant ces spectres avec des collections de spectres "de référence" modélisés à partir des protéines connues. Pour concevoir cette nouvelle méthode, nous avons posé un ensemble de contraintes à respecter : (i) la méthode développée devrait être capable de traiter des spectres qui correspondent à des peptides comportant une ou plusieurs modifications chimiques, y compris rares ou inconnues, et ce sans la connaissance préalable de la présence de telles modifications, (ii) elle devrait fournir des informations supplémentaires telles que la localisation des modifications dans les séquences d'acides aminés, mais aussi une vue globale des modifications détectées lors de l'analyse, dans le but d'aider l'interprétation des données par les biochimistes, (iii) enfin nous souhaitons que cette méthode demeure accessible aux utilisateurs des laboratoires de spectrométrie de masse, à la fois en terme de simplicité d'utilisation et d'exigence en ressources informatiques. Ce dernier point implique que la solution proposée fonctionne sur une station de travail standard en un temps acceptable pour les utilisateurs.

Ce manuscrit est organisé de la manière suivante. Nous proposons dans le Chapitre 2 un rappel général de notions concernant la protéomique avec un bref récapitulatif concernant les protéines : importance, composition et synthèse. Le Chapitre 3 poursuit avec une description détaillée du protocole de spectrométrie de masse dans le cadre de l'identification de protéines, tel qu'il est le plus souvent mis en œuvre, depuis l'acquisition des données jusqu'aux méthodes pour les interpréter. Nous exposons dans le Chapitre 4 les difficultés rencontrées par les approches algorithmiques d'identification des peptides par comparaison de spectres. Nous y discutons également un certain nombre de solutions proposées ces dernières années pour tenter d'y remédier, puis nous présentons les grandes caractéristiques de la solution que nous avons déve-

loppée. Le Chapitre 5 contient une présentation détaillée de la solution que nous proposons : tout d'abord, une description précise de la structure de données que nous avons conçue, suivie de l'intégralité du processus d'exploitation de cette structure. Notre méthode couvre ainsi le prétraitement des spectres jusqu'à la restitution des résultats. Ce chapitre se clôt avec des explications détaillées concernant la manière dont notre solution est implémentée. Une évaluation de la solution que nous proposons est effectuée dans le Chapitre 6, concernant principalement les temps d'exécution et la consommation mémoire en fonction des paramètres qui ont le plus d'impact sur ces derniers. Enfin, le Chapitre 7 contient une analyse de l'influence des principaux paramètres sur les ensembles de peptides identifiés, puis positionne notre solution en terme de performance d'identification. Nous y proposons également des exemples qui illustrent le potentiel de cette nouvelle approche en ce qui concerne le traitement des modifications non renseignées a priori. Nous terminons ce document par un rappel des principales avancées apportées par notre approche, proposons un ensemble de pistes pour l'améliorer, puis nous terminons par quelques perspectives ouvertes par notre contribution.

Protéomique

2

Les protéines, constituants essentiels de la machinerie du vivant

2.1 Rôle et origine des protéines

Les *protéines* sont des macromolécules (ou molécules de grande taille) qui furent décrites pour la première fois par le scientifique Néerlandais Mulder [101], et plus tard nommées protéines par le scientifique Suédois Berzelius en 1838 [63]. Les protéines se retrouvent chez tous les organismes vivants, où elles réalisent un vaste ensemble de fonctions cellulaires indispensables à leur survie et à leur bon développement.

2.1.1 Rôle des protéines

Au sein des cellules des organismes vivants, les protéines contribuent à la quasi-totalité des fonctions cellulaires. Ces fonctions sont indispensables au développement et à la survie de l'organisme : constitution de la structure cellulaire, transport de molécules, régulation du métabolisme, réception des stimuli et transport de signaux, motricité de l'organisme (contraction musculaire), défense immunitaire, stockage (mise en réserve d'acides aminés), catalyse de réactions chimiques, etc.

Les protéines peuvent remplir ce vaste ensemble de fonctions grâce à leur capacité à se lier à d'autres molécules présentes dans les cellules vivantes. Elles peuvent en particulier se lier à d'autres protéines, permettant la formation de complexes protéiques, modifiant ainsi le comportement des protéines qui y participent. Les protéines peuvent également subir des modifications chimiques qui peuvent entraîner une variation de leurs propriétés, changeant par la même occasion leurs fonctions. Ces modifications de fonctions peuvent être partie intégrante du cycle de fonctionnement normal de la cellule, mais peuvent aussi être dues à un dysfonctionnement cellulaire et générer un impact important sur l'organisme.

Au sein d'une cellule, une même protéine peut être présente de quelques exemplaires jusqu'à plusieurs dizaines de millions de copies, mais on ne retrouve pas nécessairement dans une cellule toutes les protéines qui peuvent potentiellement être produites par un organisme. La quantité de protéines présente au sein de chaque organisme varie grandement, allant de quelques dizaines de milliers de protéines par cellule chez les organismes simples [69] à quelques milliards par cellule chez l'être humain [12]. La concentration des différentes protéines au sein d'une même cellule peut être très inégale [140], rendant l'analyse des *protéines minoritaires* difficile. Les protéines survivent au sein de ces cellules de quelques minutes à quelques

années, mais sont le plus souvent dégradées par l'organisme après un délai de l'ordre d'une journée en moyenne [117].

2.1.2 Synthèse des protéines

Les protéines sont le produit de l'expression des *gènes* chez les organismes vivants. Les gènes sont l'unité de base de l'hérédité qui déterminent le phénotype d'un organisme, et sont composés d'*acide désoxyribonucléique* (ou *ADN*). L'ADN est ainsi nommé support de l'information génétique, et est composé de deux chaînes de polymères, aussi appelées *brins*, enroulées pour former une structure en double hélice. Chacun des brins est porteur d'une séquence de *nucléotides*, chaque nucléotide étant lui-même formé d'une *base nucléique*, d'un sucre et d'un groupement phosphate. Les bases des nucléotides composant les brins de l'ADN sont connectées par des liaisons hydrogène avec leur base complémentaire, comme illustré sur la Figure 2.1 : la cytosine est complémentaire à la guanine, et l'adénine est complémentaire à la thymine.

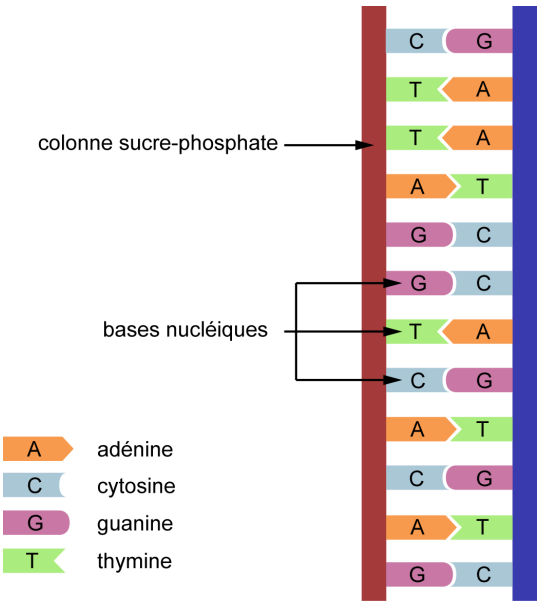


FIGURE 2.1 : Structure générale de l'ADN (vue planaire) illustrant le principe des bases nucléiques complémentaires (*illustration originale*).

L'information génétique stockée par les brins d'ADN peut donner lieu à la production de protéines à l'issue du processus d'expression des gènes. On dit dans ce cas que ce gène *encode* une protéine. Ce processus commence par la *transcription* de l'un des deux brins de l'ADN par une *enzyme*, l'*acide ribonucléique* (ou ARN) polymérase. La transcription consiste en la synthèse d'une autre séquence de nucléotides appelé ARN, à partir d'un brin d'ADN, comme présenté sur la Figure 2.2. La séquence d'ARN produite à partir d'un brin d'ADN est complémentaire de ce dernier. On notera néanmoins que dans le cas des brins d'ARN, la base nucléique thymine est remplacée lors de la transcription par une autre base nucléique, l'uracile. L'ARN produit lorsque le gène qui encode une protéine est transcrit se nomme *ARN messager*. Ce dernier va migrer dans la cellule vers les *ribosomes*, des complexes cellulaires responsables de la *traduction* des ARN messagers en protéines.

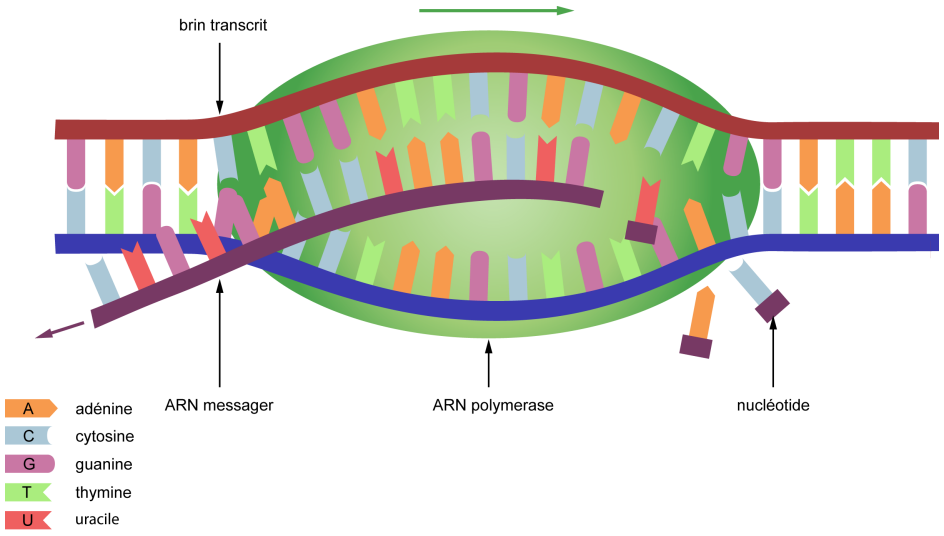


FIGURE 2.2 : Transcription d'un brin d'ADN en ARN messager par un ARN polymérase (*illustration originale*).

Les ribosomes sont des complexes moléculaires ayant la faculté de se lier aux ARN messagers et de lire leurs séquences de bases nucléiques. Cette lecture s'effectue par groupes de trois bases nucléiques, triplets appelés *codons*. A chaque anti-codon de l'ARN correspond un codon dit complémentaire (c'est-à-dire dont les trois bases nucléiques sont complémentaires) porté par une autre molécule présente dans la cellule, l'*ARN de transfert*. Ce sont ces ARN de transfert avec lesquels le ribosome va interagir pour produire les protéines lors du processus de traduction.

Chaque ARN de transfert peut se lier à un composant organique précis, appelé *acide aminé*, qui dépend du codon porté par l'ARN. Au fur et à mesure que les ARN de transfert sont associés au brin d'ARN messager grâce à la complémentarité des codons et des anti-codons, puis décrochés, les acides aminés associés aux ARN de transfert se lient les uns aux autres pour former une chaîne d'acides aminés. Ce processus est illustré sur la Figure 2.3. Cette chaîne d'acides aminés est appelée *peptide* (ou *polypeptide*, si elle dépasse une cinquantaine d'acides aminés).

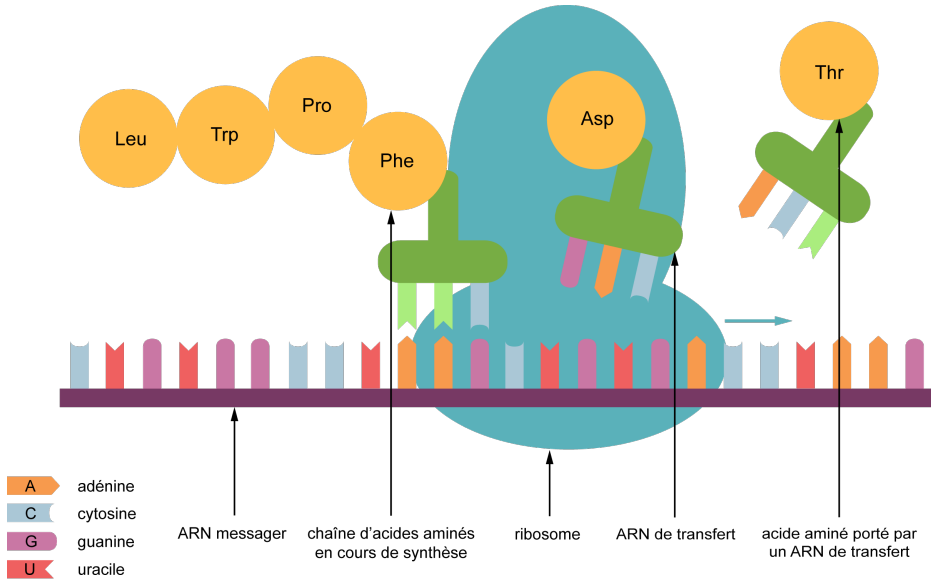


FIGURE 2.3 : Production d'une chaîne d'acides aminés par un ribosome lors du processus de traduction (*illustration originale*).

Une fois la traduction achevée, c'est-à-dire lorsque le ribosome rencontre un codon spécifique, appelé *codon stop*, la chaîne d'acides aminés est libérée et se replie sur elle-même, acquérant une structure tridimensionnelle. Cette chaîne de polypeptides forme une protéine, dont les fonctions sont définies par sa séquence d'acides aminés, mais également par sa conformation dans l'espace.

2.2 Structure des protéines

Les protéines sont constituées comme nous venons de le voir de longues chaînes d'acides aminés. Un acide aminé est un composant organique comportant un *groupe carboné*, une *fonction carboxylique* (COOH), une *fonction amine* (NH_2) ainsi qu'une *chaîne latérale* qui lui est spécifique (voir Figure 2.4). Les acides aminés sont principalement composés de carbone (C), d'hydrogène (H), d'oxygène (O) et d'azote (N), et plus rarement d'autres composés atomiques (par exemple le sélénium (Se) dans l'acide aminé sélénocystéine ou le tellure (Te) dans l'acide aminé tellurocystéine).

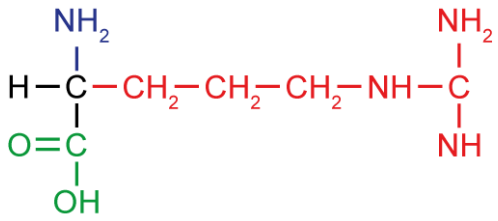


FIGURE 2.4 : Structure de l'acide aminé arginine (Arg) sous sa formule semi-développée (*image originale*). Le groupe carboné (en noir), la fonction amine (en bleu) et la fonction carboxylique (en vert) sont communs à tous les acides aminés. La chaîne latérale (en rouge) est en revanche spécifique de chaque acide aminé.

Il existe naturellement plus de 500 acides aminés différents, mais les organismes vivants n'en utilisent que 20, dits acides aminés *canoniques*. On notera les deux exceptions que sont la pyrrolysine et la sélénocystéine, des acides aminés non canoniques que l'on retrouve chez certains être vivants dans des proportions restreintes. Chaque acide aminé possède une *masse* exprimée en *Dalton* (noté Da). Un Dalton est équivalent à $1/12^e$ de la masse d'un carbone 12 (l'isotope de carbone le plus fréquent), soit $1Da \approx 1.661 \times 10^{-27} kg$. La masse totale de l'acide aminé correspond à la somme des masses des atomes qui le composent, en Dalton.

La Table 2.1 présente les différents acides aminés canoniques et pour chacun d'entre eux leur notation sous forme abrégée, leur composition et leur masse totale (mesurée en Dalton).

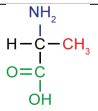
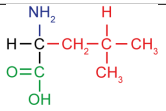
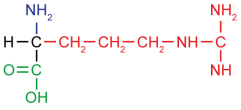
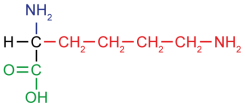
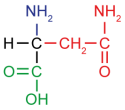
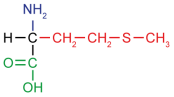
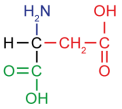
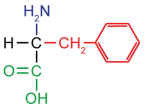
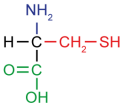
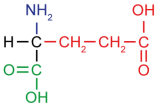
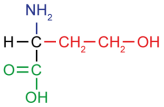
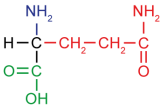
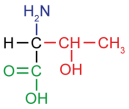
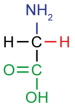
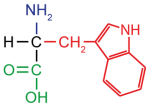
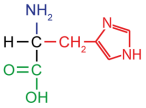
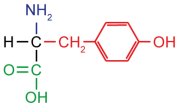
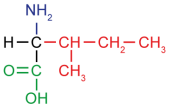
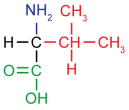
Acide aminé	Abrév.	Masse (Da)	Représentation	Acide aminé	Abrév.	Masse (Da)	Représentation
alanine	A	89.094		leucine	L	131.175	
arginine	R	174.203		lysine	K	146.189	
asparagine	N	132.119		méthionine	M	149.208	
aspartate	D	133.104		phénylalanine	F	165.192	
cystéine	C	121.154		proline	P	115.132	
glutamate	E	147.131		sérine	S	105.093	
glutamine	Q	146.146		thréonine	T	119.119	
glycine	G	75.067		tryptophane	W	204.228	
histidine	H	155.156		tyrosine	Y	181.191	
isoleucine	I	131.175		valine	V	117.148	

TABLE 2.1 : Liste des 20 acides aminés canoniques. Chaque acide aminé est accompagné de sa notation abrégée, de sa masse totale exprimée en Dalton arbitrairement arrondie à 3 décimales et de sa représentation en formule semi-développée.

La notation abrégée est utilisée pour représenter la séquence d'acides aminés qui composent un peptide, un polypeptide ou une protéine. Ces séquences d'acides aminés possèdent une *extrémité amino-terminale* (aussi appelée N-terminale) qui comprend la fonction amine et une *extrémité carboxy-terminale* (aussi appelée C-terminale) qui comprend la fonction carboxylique. Par convention, l'extrémité N-terminale est considérée comme le début de la séquence et l'extrémité C-terminale comme sa fin. Dans le cas de deux acides aminés liés entre eux, le premier a perdu un atome d'oxygène et un atome d'hydrogène de la fonction carboxylique tandis le second a perdu un atome d'hydrogène de la fonction amine, le tout formant une molécule d'eau (appelée perte d'eau). La Figure 2.5 illustre la liaison de deux acides aminés, aussi appelée *liaison peptidique*. Cet assemblage est le même pour tous les acides aminés liés au sein d'une séquence

d'acides aminés, ce qui permet de calculer la masse totale de cette séquence en additionnant les masses de chaque acide aminé qui la compose et en soustrayant la masse des pertes d'eau.

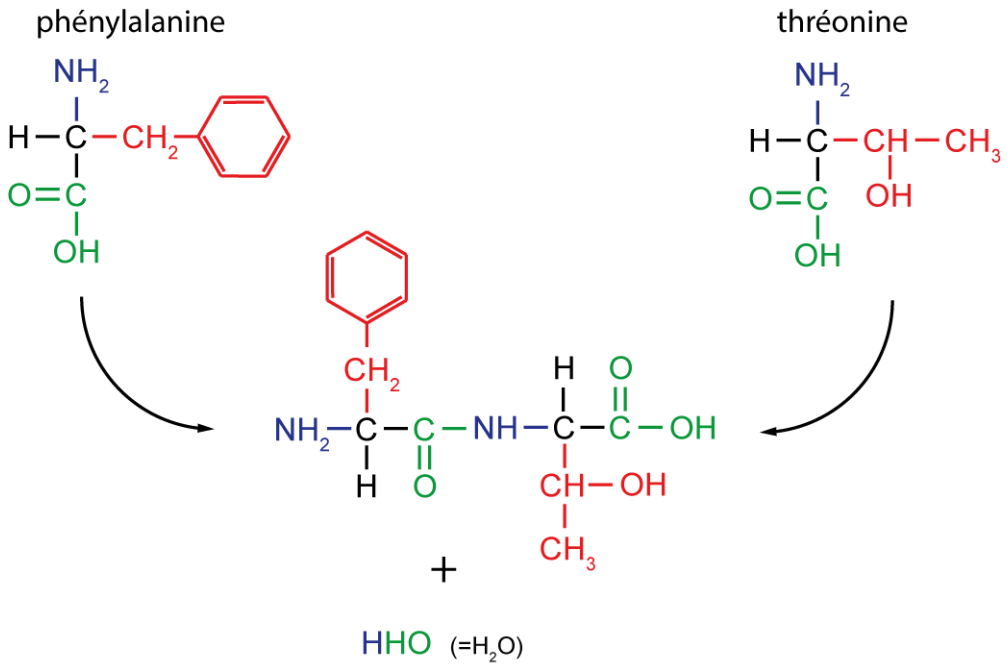


FIGURE 2.5 : Illustration de la liaison entre deux acides aminés (phénylalanine et thréonine) (*illustration originale*). L'acide aminé phénylalanine a perdu un atome d'hydrogène et un atome d'oxygène de la fonction carboxylique, lesquels se sont liés à l'atome d'hydrogène perdu par la fonction amine de la thréonine pour former une molécule d'eau. La liaison entre les deux acides aminés s'effectue aux emplacements de ces pertes d'atomes entre la fonction carboxylique de la phénylalanine et la fonction amine de la thréonine.

2.3 Modifications

La production de protéines au travers de la seule expression des gènes, par les processus de transcription et traduction que nous venons de présenter, ne suffit pas à expliquer l'immense diversité des protéines chez les organismes vivants. Plusieurs phénomènes supplémentaires viennent enrichir le répertoire des protéines existantes, parmi lesquels les *mutations* et les *modifications post-traductionnelles*.

2.3.1 Mutations

Les mutations sont des modifications permanentes d'une séquence d'ADN, résultant de la modification d'une ou plusieurs de ses bases nucléiques présentes. Les mutations peuvent être d'origines variées (mutations spontanées, erreurs lors de la réplication de l'ADN, mutations induites par des agents mutagènes, ou bien simplement variations individuelles). Parce que la séquence d'une protéine est en grande partie définie par la séquence de l'ADN au travers des processus de transcription et de traduction, une variation au niveau de l'ADN entraîne une différence au niveau de la séquence d'acides aminés de la protéine synthétisée. Cette différence peut avoir un impact important, pouvant aller jusqu'à un dysfonctionnement et à une dégradation accélérée de la protéine synthétisée.

Les trois grands types de mutations que l'ont peut rencontrer sont l'insertion d'un ou plusieurs acides aminés supplémentaires dans la séquence d'une protéine, la suppression d'un ou plusieurs acides aminés dans la séquence de la protéine, ou bien le remplacement d'un ou plusieurs acides aminés par un autre.

2.3.2 Modifications post-traductionnelles

Les modifications post-traductionnelles (ou PTMs) sont des modifications chimiques survenant après la synthèse de la protéine et potentiellement tout au long de son cycle de vie. Ces modifications sont le plus souvent enzymatiques.

Les modifications post-traductionnelles sont localisées sur les chaînes latérales des acides aminés ou bien sur l'extrémité N-terminale ou C-terminale de la protéine. La composition atomique de l'acide aminé ou des extrémités affectés s'en trouve modifiée (et donc également sa masse), ainsi que bien souvent la fonction de la protéine qui contient l'acide aminé portant la modification [98, 76].

Les modifications post-traductionnelles ne se manifestent pas toutes avec la même fréquence, et certaines sont spécifiques de certains acides aminés, tandis que d'autres peuvent se produire sur un grand nombre d'acides aminés différents. Tous les acides aminés ne sont pas sujets à des modifications post-traductionnelles en proportions équivalentes. Des bases de données répertorient les modifications post-traductionnelles (et autres modifications) et les informations connues à leur propos, parmi lesquelles *UniMod* [29] qui contient près de 1500 modifications dont de nombreuses PTMs.

L'analyse des modifications post-traductionnelles revêt un grand intérêt biologique en raison de la capacité de ces modifications à influer sur les fonctions des protéines et parfois d'entraîner des dysfonctionnement importants pouvant conduire à des maladies [37]. En outre, les modifications post-traductionnelles augmentent la variété des protéines qui peuvent être utilisées par les organismes vivants. Elles génèrent néanmoins une complexité supplémentaire importante lors de l'analyse des protéines.

2.4 Protéome et diversité des protéines

On appelle *protéome* l'ensemble des protéines présentes dans un organisme à un instant donné. Il existe plusieurs bases de données de protéines desquelles il est possible d'extraire tout ou partie du protéome d'une ou de plusieurs espèces vivantes. Ces bases de données peuvent être déduites de l'information génomique par la réplication du processus de synthèse des protéines (traduction des séquences d'ADN en suite d'acides aminés), pour en déduire quelles sont les protéines synthétisées par cet organisme. Aujourd'hui, la plus utilisée de ces banques de protéines est *UniProt* [136], créée en 2004 dans le cadre du projet Universal Protein knowledgebase. UniProt est découpée en deux sous-banques qui sont SwissProt et TrEMBL, la première contenant uniquement des protéines manuellement contrôlées par des experts tandis que la seconde contient des traductions automatiques réalisées depuis des génomes connus. En 2018, Uniprot-TrEMBL contient environ 111 millions de protéines (Figure 2.6) tandis que Uniprot-SwissProt en contient près de 500 fois moins (environ 500 milliers), ce qui illustre la présence d'une forte quantité de données non validées.

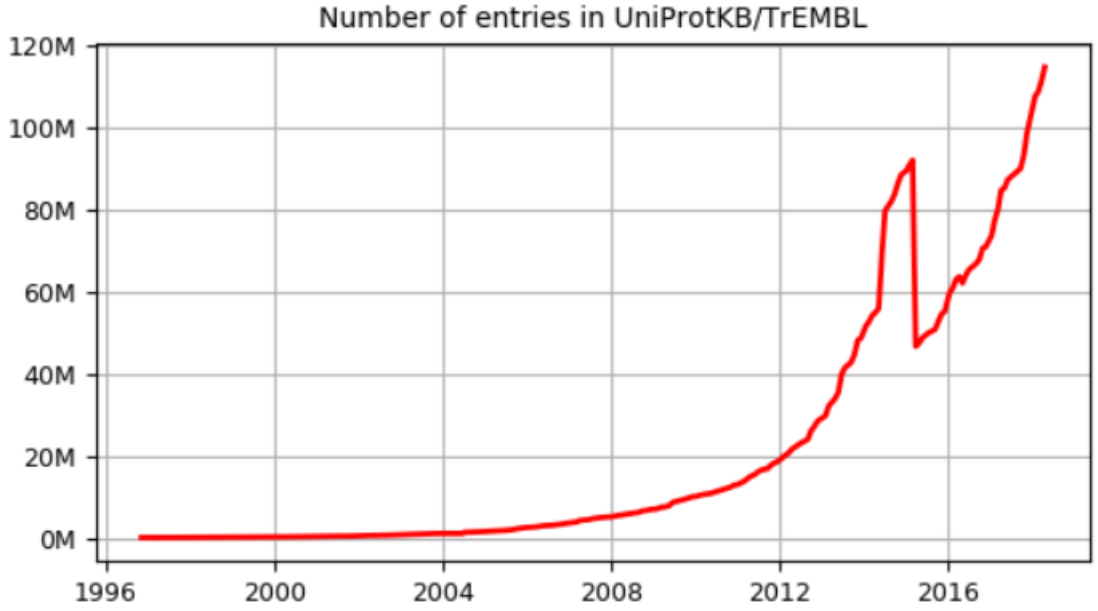


FIGURE 2.6 : Evolution du nombre de protéines répertoriées dans UNIPROT-TrEMBL depuis 1997. On observe en 2015 une chute brutale du nombre d'entrées, due à une opération de réduction du nombre de protéines dupliquées. (https ://www.uniprot.org/news/2015/04/01/release). Source : European Bioinformatics Institute (EBI), http ://www.ebi.ac.uk/uniprot/TrEMBLstats/

Les données des banques de protéines peuvent généralement être récupérées sous la forme d'un fichier informatique qui contient des métadonnées concernant chaque protéine de l'organisme sélectionné, ainsi que la séquence complète de ces protéines.

L'étude d'un protéome est nommée *protéomique*. La richesse du protéome d'un organisme est étroitement corrélée avec la complexité de l'organisme en question, la synthèse des protéines dépendant essentiellement du *génome* de l'organisme, c'est-à-dire de l'ensemble de ses gènes. On estime le nombre de gènes codants pour des protéines chez l'être humain aux environs de 20 000 [24]. Ces gènes codants produisent de 50 000 à 100 000 ARN messagers différents, qui après traduction et en présence de modifications des protéines conduisent à un répertoire de plus de cinq millions de protéines. L'importante quantité et variabilité des protéines est l'une des difficultés que rencontre la protéomique. Les modifications chimiques qui affectent les protéines ne sont très vraisemblablement pas toutes connues aujourd'hui. En outre, les protéines ne sont pas toutes présentes dans chaque protéome, leur représentation variant au cours du temps selon les fonctions à assurer au sein de l'organisme. Leur concentration au sein d'un même tissu peut de plus grandement varier, ce qui rend l'analyse des protéines minoritaires très complexe. Parmi les techniques de protéomique, la *spectrométrie de masse* est aujourd'hui la plus répandue et s'adosse à la bioinformatique pour réaliser l'analyse des résultats.

Nous allons dans le prochain chapitre décrire le processus d'acquisition des données par spectrométrie de masse en tandem tel qu'il est le plus souvent suivi dans un objectif d'identification des protéines présentes dans un mélange a priori inconnu. Puis, nous présenterons les méthodes utilisées pour analyser les données produites lors de l'analyse de spectrométrie de masse.

3

Analyse des protéines par spectrométrie de masse en tandem

La *spectrométrie de masse* [128] est une technique qui permet d'analyser des *ions*, c'est-à-dire des molécules électriquement chargées sous forme gazeuse. Elle permet de mesurer le *ratio masse sur charge* d'un ion (ce ratio est souvent simplement appelé abusivement masse), c'est-à-dire la somme des masses (en Dalton) des atomes qui composent cet ion, divisée par le nombre de charges électriques portées par cet ion. En protéomique, la connaissance de la masse des composants d'un ion permet de reconstituer la séquence du peptide qui a produit cet ion. En dépit de l'existence d'autres techniques d'analyse des protéines, la spectrométrie de masse s'est imposée petit à petit lors des trente dernières années comme la technique principale pour l'analyse des protéines et des modifications chimiques qui les affectent.

Les premières expérimentations relatives à la spectrométrie de masse datent de la fin du XIX^e siècle. Elles font suite à la découverte par Wien [38] de la capacité qu'ont les champs magnétiques à séparer les constituants gazeux selon leur ratio masse sur charge. En 1918 et 1919 respectivement, Dempster et Aston posent les fondements de la spectrométrie de masse moderne [34, 127], mais il faudra attendre la fin des années 1980 pour qu'elle se démocratise grâce au développement de nouvelles méthodes d'*ionisation* [141, 79]. Ces techniques vont élever la spectrométrie de masse au rang de technique de choix pour l'étude et la caractérisation des protéines en permettant l'analyse de molécules de masses importantes.

Aujourd'hui, la spectrométrie de masse permet d'*identifier* et de *quantifier* les protéines d'un échantillon inconnu, de caractériser les modifications post-traductionnelles qui affectent les protéines, mais aussi de caractériser la conformation spatiale des protéines ou les complexes que plusieurs protéines peuvent former. Dans le cadre de ce manuscrit, nous nous intéressons à son application pour l'identification de protéines et la caractérisation des modifications post-traductionnelles. A l'origine effectué à la main, le complexe processus d'analyse des données issues de spectrométrie de masse nécessite aujourd'hui l'utilisation d'outils informatiques adaptés pour traiter l'important volume de données générées.

Ce chapitre présente dans un premier temps une vue d'ensemble d'une analyse protéomique traditionnelle, en partant d'un mélange de protéines inconnu à analyser, et en allant jusqu'aux résultats de l'analyse informatique des données générées. Dans un second temps, chaque étape est développée plus en détail pour apporter les informations nécessaires à la compréhension des problématiques liées aux algorithmes actuels d'analyse des protéines par spectrométrie de masse.

3.1 Acquisition des données par spectrométrie de masse

L'acquisition des données par spectrométrie de masse comprend la préparation des échantillons et l'acquisition des mesures de masse par l'appareil. Il existe de nombreuses variations de ce processus, en fonction à la fois de la nature des échantillons à analyser, du matériel utilisé pour préparer ces échantillons et du modèle de spectromètre de masse. Nous nous contentons ici de décrire le processus dans sa forme utilisée de manière routinière par les laboratoires de spectrométrie de masse. L'essentiel des informations exposées dans cette section et sauf indication contraire sont issues de l'ouvrage "Computational methods for mass spectrometry proteomics" [39].

3.1.1 Vue d'ensemble d'une analyse protéomique

La Figure 3.1 donne une vue d'ensemble des différentes étapes d'un processus classique de *spectrométrie de masse en tandem* dans le cadre de l'identification de protéines. Ce processus se découpe en trois principales étapes que sont la préparation des échantillons, l'acquisition des données et enfin l'analyse de ces données. Le processus de spectrométrie de masse fait intervenir un appareil nommé spectromètre de masse. Les spectromètres de masse sont des *appareils analytiques* comportant un ou plusieurs *analyseurs de masses* (aussi appelés cellules de détection), qui peuvent déduire le ratio masse sur charge d'un ion ou séparer des ions en utilisant les propriétés électroniques de ces ions. En protéomique, les ions sont des peptides à l'état ionisé, c'est-à-dire sous forme gazeuse et électriquement chargés.

Les protéines du mélange à analyser sont souvent séparées en différents échantillons à l'aide d'une électrophorèse (a). Pour chaque échantillon d'intérêt, une *digestion enzymatique* des protéines (ou hydrolyse) permet d'obtenir un ensemble de peptides (b), qui sont ensuite séparés (c), et ionisés pour être introduits dans le *spectromètre de masse* (d). Le spectromètre de masse permet de réaliser une première mesure (dite *mesure MS* ou *MS1*) des ions introduits (e), avant d'en *sélectionner* certains pour une étape de *fragmentation* (f). Les fragments obtenus peuvent ensuite être mesurés à leur tour (g), c'est la *mesure MS/MS* (aussi appelée *MS2*). A la sortie de l'appareil, le signal est retraité et les informations obtenues sont restituées sous forme de *spectres de masse* pour être analysées (h). Après acquisition des données par le spectromètre de masse, chaque spectre de masse doit être associé à la séquence d'acides aminés (le peptide) qui a généré l'ion à partir duquel le spectre a été obtenu par l'appareil (i). Une fois reconstitué l'ensemble des peptides supposés introduits dans l'appareil, les séquences de ces peptides sont utilisées pour reconstituer les protéines ayant le plus probablement produit cet ensemble de peptides par digestion enzymatique (j).

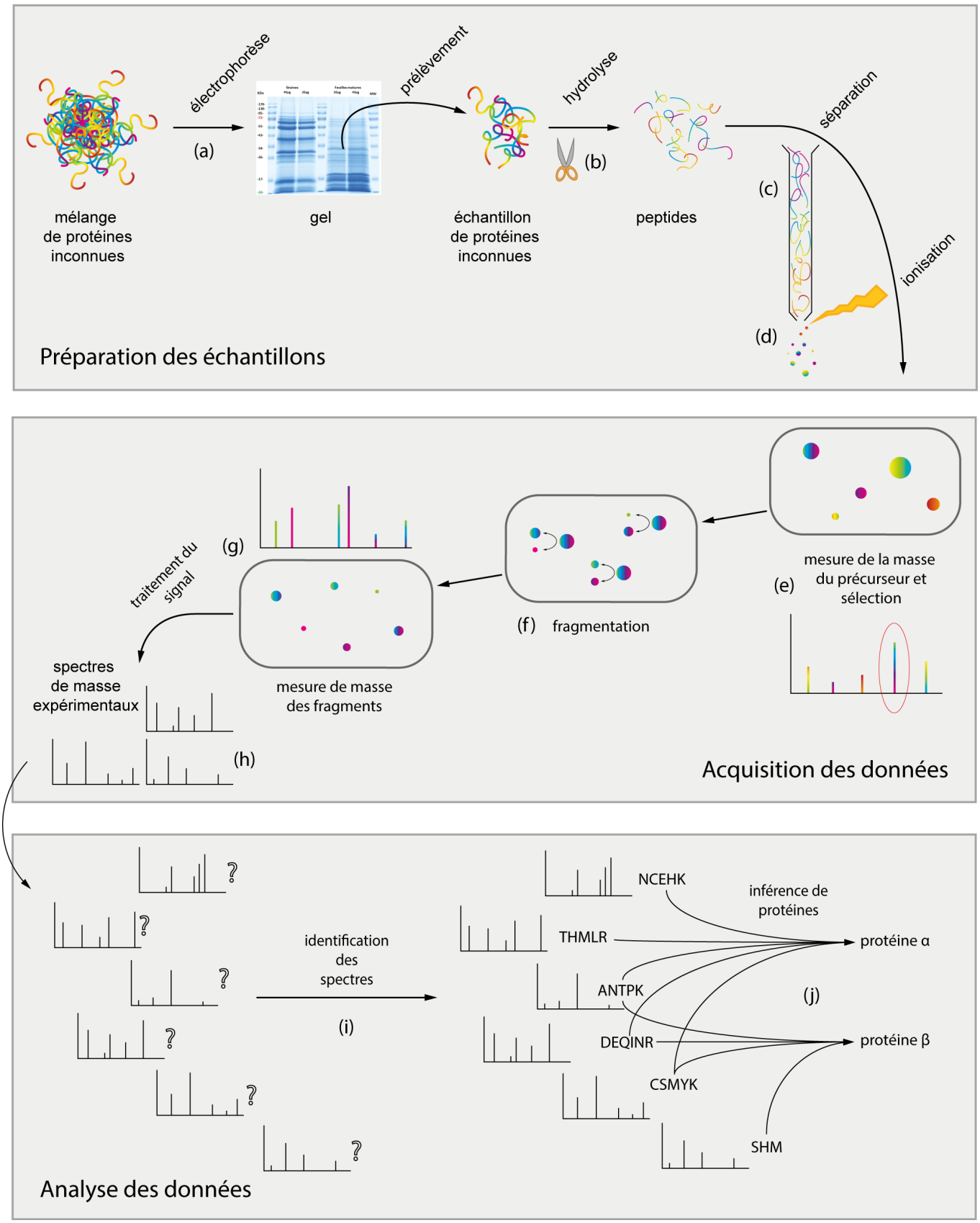


FIGURE 3.1 : Représentation schématique du processus général d'identification de protéines par spectrométrie de masse en tandem.

3.1.2 Préparation des échantillons

Lors d'une expérience de spectrométrie de masse, il est souhaitable de réduire la complexité de l'échantillon à analyser. L'objectif est de limiter autant que possible la complexité des étapes d'analyse qui suivent l'acquisition des données par le spectromètre, d'accroître la détection des protéines minoritaires et d'augmenter la capacité de discrimination des protéines à l'issue du processus.

Pour réduire la complexité du mélange, les protéines sont séparées en plusieurs échantillons qui seront analysés indépendamment. Cette étape de séparation des protéines permet de réduire la diversité en protéines de chaque échantillon analysé. On augmente par conséquent les chances de détection des protéines minoritaires, tout en diminuant la complexité d'analyse de chaque échantillon.

Séparation des protéines

Aujourd'hui, la technique la plus fréquemment utilisée en spectrométrie de masse pour réaliser l'étape préparatoire de séparation des protéines est l'*électrophorèse* sur gel.

L'électrophorèse est une technique qui permet de séparer des molécules en fonction de leur facilité de migration au sein d'un support comportant des pores, le gel. En appliquant un champ électrique à ce gel empli d'une phase aqueuse saline (cette solution joue le rôle d'électrolyte et conduit le courant), on fait migrer les molécules présentes dans le support. Cette migration s'effectue principalement en fonction de la masse et de la *charge* de ces molécules (leur conformation tri-dimensionnelle peut néanmoins aussi impacter leur capacité à se mouvoir au travers des pores).

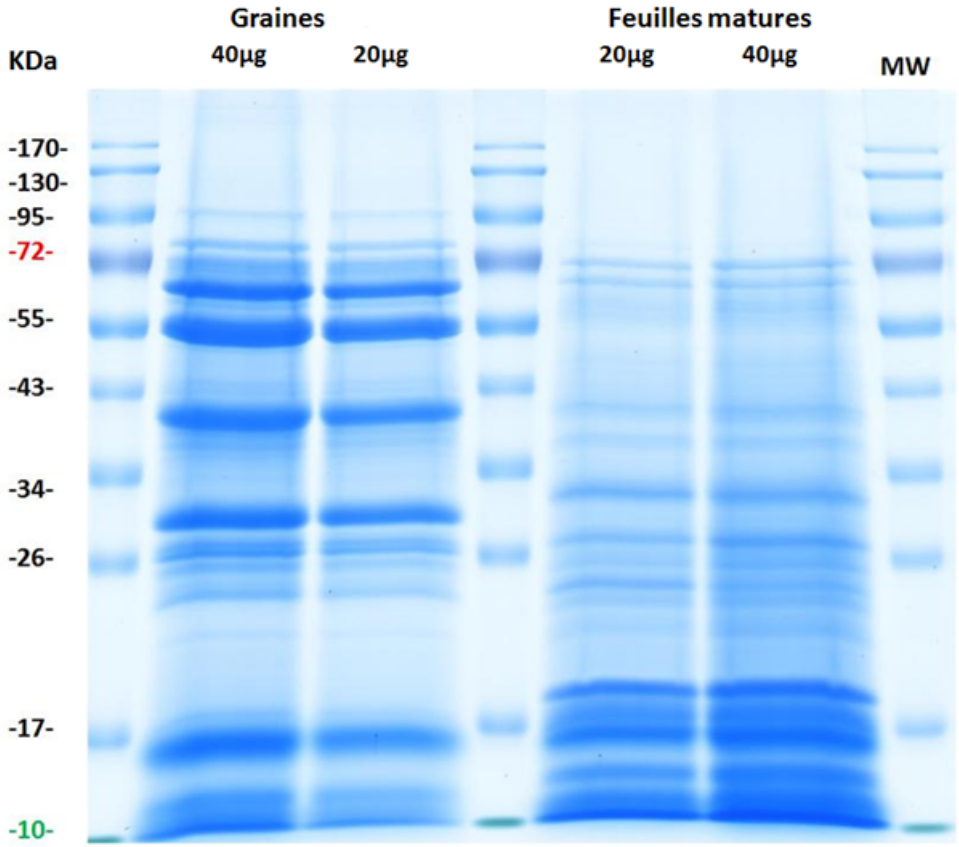


FIGURE 3.2 : Image d'un gel obtenu à l'issue d'une séparation par électrophorèse de protéines végétales. La masse indiquée sur l'axe des ordonnées est fournie en KiloDaltons. Source : Institut National de Recherche en Agronomie (INRA, Colette Larre).

Effectuer une séparation des protéines au travers d'une électrophorèse 1D (pour une dimension, la technique de séparation actuellement la plus courante) permet d'obtenir un gel comportant de multiples bandes (un exemple est fourni en Figure 3.2). Chaque bande est un emplacement du gel qui comporte un ensemble de protéines possédant une masse et une charge relativement proches, les autres ayant été séparées vers d'autres bandes pendant la migration. Chaque bande peut faire l'objet d'un prélèvement qui sera utilisé comme échantillon de protéines à analyser, dont le nombre et la variabilité des protéines se trouvent ainsi réduits.

Hydrolyse des protéines

Pour une utilisation en spectrométrie de masse, l'hydrolyse des protéines en peptides (aussi parfois appelée digestion) est réalisée à l'aide d'une enzyme de digestion. Cette enzyme de digestion est un catalyseur présent chez les organismes vivants, qui réduit le volume et la taille des molécules et facilite leur utilisation par le système, notamment en accélérant les réactions chimiques. L'enzyme de digestion agit sur les liaisons peptidiques à l'extrémité de certains acides aminés précis (le *site actif* de cette enzyme) et clive la liaison avec les acides aminés voisins. L'enzyme la plus souvent utilisée en spectrométrie de masse est la *trypsine*. Cette enzyme agit sur les liens peptidiques à l'extrémité C-terminale des acides aminés arginine (R) et lysine (K), c'est-à-dire qu'elle sectionne les protéines après les acides aminés R ou K. Une illustration d'une séquence de protéine et des sites actifs de la trypsine est fournie en Figure 3.3.

>ENSP00000357879 pep:known chromosome:GRCh37:1:151227179:151239955:1
gene:ENSG00000159352 transcript:ENST00000368884

MVLESTMVCVDNSEYMRNGDFLPTRLQAQQDAVNIVCHSKTRSNPENNVGLITLANDCEVLTTLTPDTGRILS
KLHTVQPKGKITFCTGIRVAHLALKHRQGKNHKMRIIAFVGSPVEDNEKDLVKLAKRLKKEKVNVDIINFGEEEV
NTEKLTAFVNTLNGKDGTGSHLVTVPPGPSLADALISSPILAGEGGAMLGLGASDFEFGVDPSADPELALALRV
SMEEQRQRQEEEARRAAAASAAEAGIATTGTEDSDDALLKMTISQQEFGRTGLPDLSSMTEEEQIAYAMQMS
LQGAEFGQAESADIDASSAMDTSEPAKEEDDYDVMQDPEFLQSVLENLPGVDPNNEAIRNAMGSLASQATK
DGKKDKKEEDKK

FIGURE 3.3 : Séquence de protéine au format fasta issue de la banque de protéines UniProt [136]. L'entête en bleu donne des informations sur la protéine. Les sites de coupure de la trypsine sont indiqués en rouge. Le peptide souligné en vert illustre un exemple de peptide tryptique, le peptide souligné en jaune illustre un exemple de missed-cleavage en position 14, et le peptide souligné en violet illustre un peptide semi-tryptique.

Après digestion par la trypsine, la protéine produit un ensemble de peptides se terminant par des arginines (R) ou des lysines (K), à l'exception éventuelle du dernier peptide de chaque protéine. De tels peptides sont dits *tryptiques*. Dans le cas d'une protéine comportant un nombre de k acides aminés arginine ou lysine, on s'attend à obtenir $k + 1$ peptides à l'issue de la digestion enzymatique (à l'exception du cas spécifique où une protéine se terminerait par une arginine ou une lysine, auquel cas on s'attend à obtenir k peptides). En pratique, cet exemple de digestion idéale n'est pas nécessairement vérifié : les protéines sont des molécules complexes possédant une conformation tri-dimensionnelle qui peut rendre impossible l'action d'une enzyme de digestion en masquant certains sites actifs. Il est également possible que les conditions physico-chimiques nécessaires à l'action de l'enzyme de digestion ne soient pas réunies. Il existe par exemple un débat autour de la capacité de la trypsine à cliver convenablement une protéine lorsque son site actif est suivi d'une proline [113].

L'absence de clivage par l'enzyme utilisée pour la digestion à l'emplacement de l'un de ses sites actifs est nommée *missed-cleavage*. Ce phénomène conduit à la production de peptides plus longs (voir Figure 3.3). Dans le cas de la trypsine, de tels peptides contiennent encore au moins une arginine ou une lysine supplémentaire, non située en fin de chaîne peptidique.

Il est également possible que des peptides ne demeurent pas entiers mais soient en partie dégradés. De tels peptides sont nommés *semi-tryptiques* (voir Figure 3.3). Les origines des peptides semi-tryptiques sont multiples, parmi lesquelles figurent la dégradation naturelle des chaînes d'acides aminés lors de leur cycle de vie, la modification des peptides par des conditions expérimentales spécifiques ou par le processus de spectrométrie de masse.

La prise en compte de ces cas de missed-cleavage et de peptides semi-tryptiques peut compliquer significativement l'analyse des données produites par spectrométrie de masse.

Séparation des peptides

Après la digestion enzymatique des protéines, les peptides sont analysés par le spectromètre de masse. Ils sont d'abord séparés pour être introduits dans le spectromètre de masse de manière contrôlée. Le plus souvent, la séparation des peptides est effectuée par une étape de *chromatographie*.

Le principe de la chromatographie consiste à introduire les peptides dans une colonne de chromatographie, lesquels vont être entraînés par une phase mobile (souvent liquide) au travers d'une phase stationnaire. En fonction de la nature des phases utilisées, les peptides vont migrer à une vitesse dépendant de leurs propriétés physico-chimiques. L'intervalle de temps requis pour traverser la totalité de la colonne, appelé *temps de rétention*, va différer selon les peptides et la rapidité de la traversée pourra être ajustée grâce à une variation des paramètres de la phase mobile (pression, composition...) et en adaptant la longueur de la colonne de chromatographie.

La chromatographie peut être connectée directement à l'appareil pour y introduire les peptides, ou bien les peptides obtenus en sortie de la chromatographie seront déposés sur un support spécifique appelé matrice pour une introduction dans le spectromètre.

3.1.3 Acquisition des données

Ionisation

Les spectromètres de masse mesurant le ratio masse sur charge des ions, il est nécessaire de faire passer ces peptides dans un état gazeux et de leur apporter une ou plusieurs charges électriques. Deux principales méthodes sont utilisées pour ioniser les peptides et les introduire dans le spectromètre de masse : l'*électrospray* et la *désorption-ionisation laser assistée par matrice*.

L'ionisation par électrospray requiert de connecter la sortie de la chromatographie directement au spectromètre de masse. L'invention de cette méthode en 1989 par Fenn [47] lui a valu de partager le prix Nobel de chimie en 2002. Elle consiste à soumettre le liquide à l'issue de la chromatographie à un fort voltage, créant ainsi un aérosol (suspension de fines gouttes de liquide dans un gaz) qui peut être directement introduit dans le spectromètre. Un des avantages notables de l'ionisation par électrospray est de produire des ions *multi-chargés*, c'est-à-dire comportant plus d'une charge électrique (par opposition aux ions *mono-chargés*). En effet, les spectres produits à partir d'ions comportant plus d'une charge électrique peuvent contenir une information sur leur composition en acides aminés plus riche et plus complète.

La désorption-ionisation laser assistée par matrice (MALDI) consiste à irradier les emplacements d'une matrice (sur laquelle on aura au préalable déposé les gouttelettes obtenues en sortie de la chromatographie) avec un laser pour déclencher l'ablation de cette matrice (enlèvement de matière aux dépens du relief) et la désorption (émission de molécules de gaz ou de liquide préalablement absorbées par la surface d'un solide) des molécules qu'elle contient. La vaporisation et l'ionisation des peptides, protégés de la destruction par la matrice, découle de l'utilisation du laser. Cette technique résulte d'un développement par Hillenkamp et Karas en 1985 [79] ensuite amélioré par Tanaka en 1988 [132], qui partagera le prix Nobel de chimie de 2002. Contrairement à l'électrospray, la technique MALDI produit très peu d'ions multi-chargés.

L'ionisation de peptides est un phénomène compétitif, et il est fréquent que certains peptides n'acquièrent pas du tout de charges électriques, ce qui rend impossible leur détection et leur mesure par le spectromètre de masse.

Analyseur de masse

Un spectromètre de masse contient un ou plusieurs analyseurs de masse, composants responsables de la mesure et de la sélection des ions. Il existe plusieurs types d'analyseurs de masse qui reposent sur l'exploitation de principes physiques différents pour déduire le ratio masse sur charge d'un ion, mais leur finalité reste la même : exploiter les comportements physiques des ions en interagissant avec ceux-ci et en mesurant la réponse obtenue pour réaliser indirectement une mesure de leur masse et de leur charge.

L'analyseur cyclotron à *transformée de Fourier* utilise un champ magnétique intense pour entraîner les ions en rotation dans une cavité. La période de rotation des ions dépend dans cet environnement de leur ratio masse sur charge.

L'analyseur *quadripôle* est constitué de quatre barres magnétiques, les pôles, placées en croix parallèlement les unes des autres. Deux pôles opposés sont soumis à une charge électrique, de signe inverse à la charge des deux autres pôles. Les ions suivent une trajectoire entre les quatre pôles (ou bien sont expulsés à l'extérieur) en fonction de l'intensité et des variations des charges électriques auxquelles les quatre pôles sont soumis.

L'analyseur à *trappe d'ions* permet d'emprisonner et de faire osciller les ions pendant une durée prolongée au sein de deux anneaux polarisés. En faisant varier le potentiel électrique auquel les anneaux sont soumis, il est possible d'éjecter les ions choisis en fonction de leur ratio masse sur charge en dehors de la trappe d'ions et de déduire le ratio masse sur charge des ions restants.

L'analyseur à *temps de vol* propulse les ions dans une chambre, utilise une tension électrique pour les accélérer et mesure le temps nécessaire à ces ions pour parcourir une distance donnée, ce qui permet de calculer leur ratio masse sur charge.

Les analyseurs de masse d'un spectromètre ne sont pas nécessairement du même type, et chacun de ces analyseurs possède des caractéristiques propres : temps de mesure des ions, précision de la mesure réalisée, résolution (capacité à distinguer des ions de masses très proches), etc. Par conséquent, le spectromètre de masse utilisé pour l'expérience de spectrométrie de masse a généralement un impact important sur l'analyse des données produites, et influence la configuration des logiciels utilisés lors de celle-ci.

Mesure de masse MS

Après ionisation et introduction des peptides dans le spectromètre de masse, une première mesure du ratio masse sur charge des ions est réalisée par un analyseur de masse : c'est la mesure MS. Cette première mesure va permettre de déduire le ratio masse sur charge total des acides aminés et de la (ou les) charge(s) qui compose(nt) les ions analysés. Cette mesure de masse est exprimée en Dalton. L'abondance de chacun des ions mesurés est également reportée sous la forme d'une *intensité*, dont l'unité est arbitraire. Cette intensité peut correspondre à la quantité d'ions détectés ou à un voltage en fonction du détecteur utilisé, et est corrélée à l'abondance relative des ions sans pour autant la décrire fidèlement. L'ensemble des mesures réalisées en MS par le spectromètre de masse constitue le spectre de masse MS.

La plupart du temps, la seule mesure de masse MS ne suffit pas à identifier les peptides qui ont produit les ions mesurés par le spectromètre. En effet, deux combinaisons d'acides aminés différentes (illustrées en Figure 3.4), ou bien l'interversion des mêmes acides aminés (illustrée en Figure 3.5) peuvent aboutir au même ratio masse sur charge.

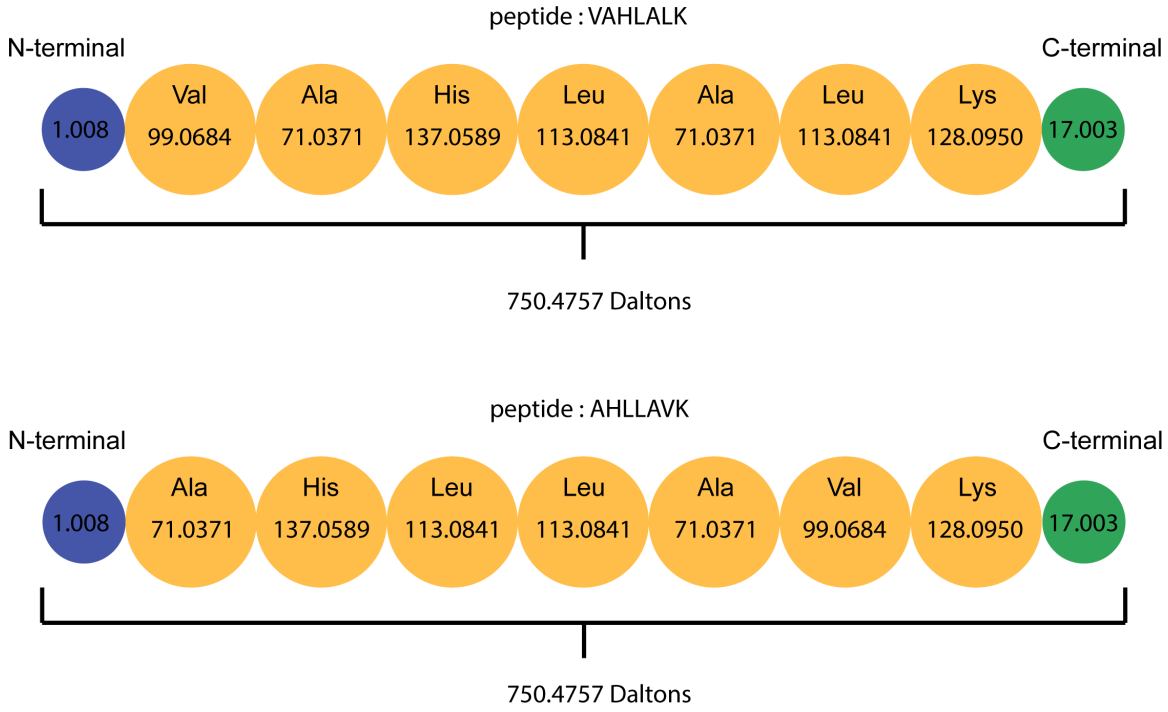


FIGURE 3.4 : Exemple de peptides de séquences différentes, correspondant à des combinaisons différentes des mêmes acides aminés, et aboutissant à la même masse.

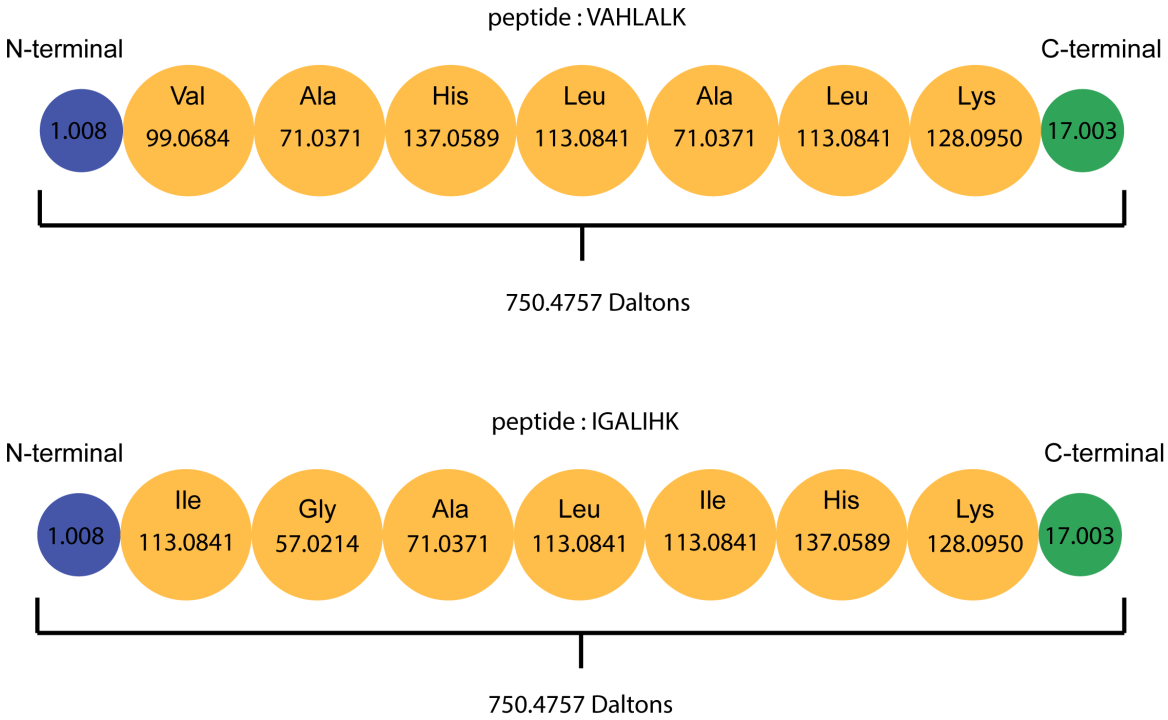


FIGURE 3.5 : Exemple de peptides de même masse ayant une composition en acides aminés différente.

Pour obtenir plus d'informations sur la composition des peptides d'intérêt, les ions correspondant à ces peptides vont être sélectionnés pour être soumis à une étape de fragmentation. Les ions sélectionnés sont appelés *ions parents* ou *ions précurseurs*. Les fragments obtenus à partir des ions parents seront ensuite soumis à une seconde mesure de masse.

Fragmentation

La fragmentation est une dissociation des ions dans un état d'énergie instable. Elle permet d'observer des motifs structurels, c'est-à-dire des valeurs de masse renseignant sur la présence d'acides aminés. Cette fragmentation peut avoir lieu au moment de l'ionisation (elle est alors nommée fragmentation in-source), ce qui est indésirable en spectrométrie de masse, ou bien provoquée dans une *chambre de collision* (qui contient le plus souvent une phase gazeuse). Il existe de multiples techniques de fragmentation, qui vont aboutir à la production de différents ions fragments dans des proportions variables.

Il existe six principaux ions fragments dûs à une fragmentation simple de la chaîne principale des acides aminés. La notation introduite en 1984 par Roepstorff et Fohlman [114] présentée en Figure 3.6 décrit ces ions fragments en fonction de l'extrémité de l'ion fragment (N-terminale ou C-terminale) et de l'emplacement de la fragmentation. Les ions portant l'extrémité N-terminale (gauche) de l'ion parent sont appelés ions *a*, *b* et *c* tandis que les ions portant l'extrémité C-terminale (droite) de l'ion parent sont appelés ions *x*, *y* et *z*. Les ions a, b et y sont généralement les ions les plus abondants.

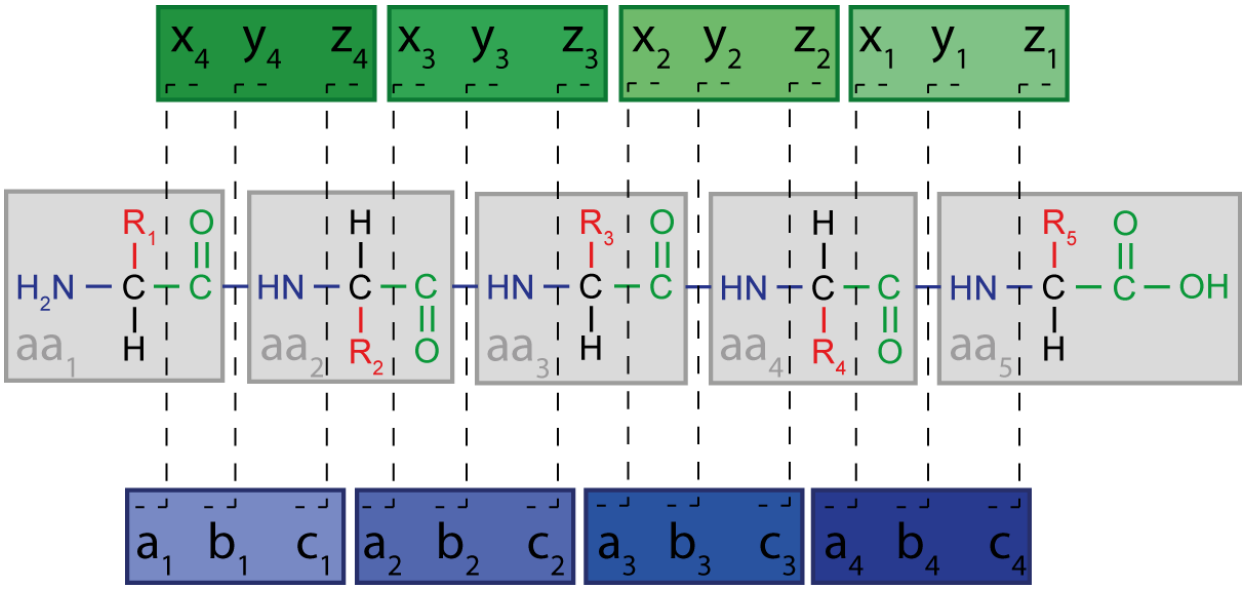


FIGURE 3.6 : Notation de la fragmentation des ions introduite par Roepstorff et Fohlman [114]. Les acides aminés sont représentés sur fond gris, leur chaîne latérale en rouge et leur parties N-terminale et C-terminale respectivement en bleu et vert. Les ions produits à partir de l'extrémité C-terminale sont représentés sur fond vert, et les ions produits à partir de l'extrémité N-terminale sur fond bleu.

Dans des conditions particulières (énergie de collision importante par exemple), il est également possible que d'autres ions se forment lors d'une double fragmentation de la chaîne principale ou d'une fragmentation de la chaîne latérale des acides aminés. Ces évènements sont généralement gênants pour l'interprétation des fragments observés.

Mesure de masse MS/MS

Lors d'une mesure de spectrométrie de masse en tandem (aussi appelée MS/MS), on met à profit la présence de deux analyseurs de masse en tandem dans un même spectromètre pour réaliser deux mesures. La première est une mesure de masse MS traditionnelle, qui permet de cibler et de sélectionner des ions (généralement les plus abondants dans un laps de temps donné) en fonction de leur masse. Les ions compris dans des *fenêtres de sélection* (la masse de l'ion plus ou moins une certaine tolérance, autour des masses sélectionnées) sont séparés des autres et soumis à une étape de fragmentation comme décrit précédemment.

Une seconde mesure de masse est alors effectuée sur les ions fragments obtenus. L'ensemble des ions fragments mesurés en MS/MS pour une fenêtre de sélection autour d'un ion parent sélectionné va constituer le spectre de masse MS/MS.

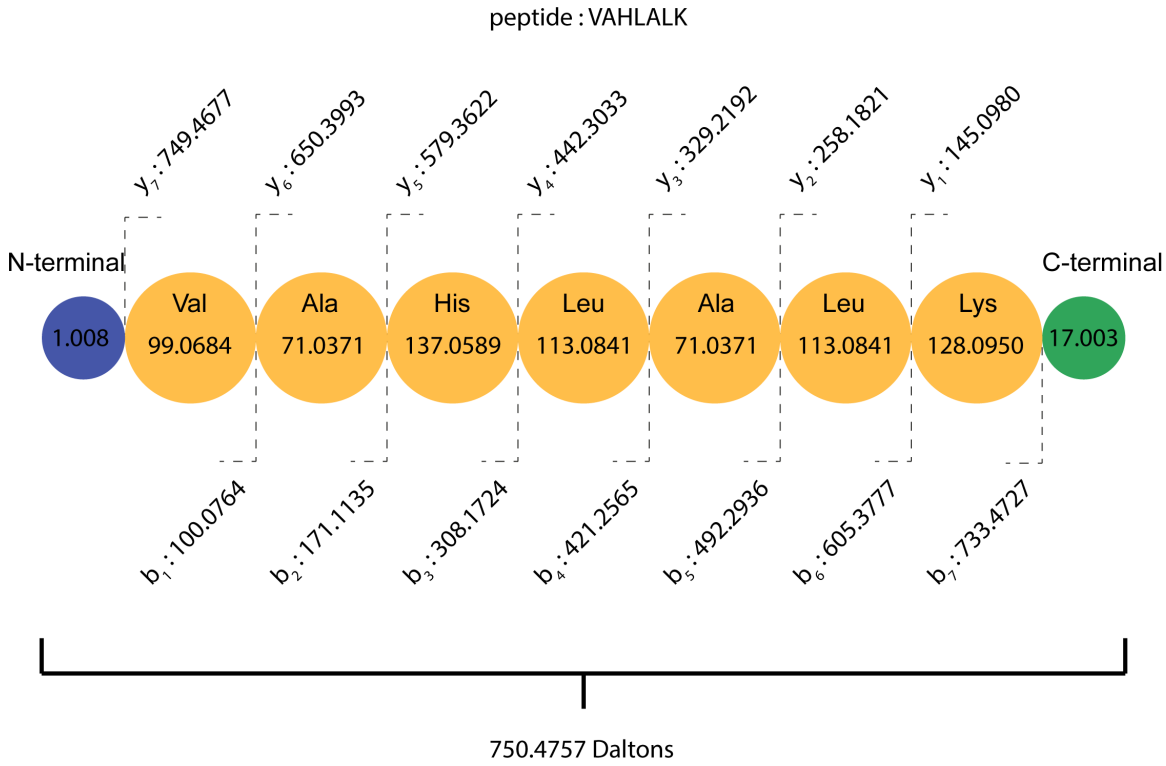


FIGURE 3.7 : Représentation des ions fragments b et y obtenus en mode MS/MS pour le peptide VAHLALK. Les différents ions fragments b et y sont représentés, accompagnés de leur numérotation usuelle, ainsi que de leur masses théoriques respectives, obtenues par fragmentation du peptide exemple VAHLALK chargé une seule fois.

Après fragmentation, la masse de chaque ion fragment porteur d'une charge électrique peut être mesurée par l'analyseur de masse. L'ensemble des mesures des ions fragments pour un même ion précurseur constitue une signature, laquelle permet de reconstituer la structure de l'ion précurseur (comme illustré sur la Figure 3.7) lorsque suffisamment d'ions fragments ont été mesurés (c'est-à-dire que le spectre MS/MS est de qualité suffisante).

3.1.4 Analyse du signal

Amélioration du signal

Les mesures de masse et d'intensité réalisées par les analyseurs des spectromètres de masse ne sont en réalité pas des valeurs discrètes. L'appareil enregistre en effet un signal continu, qui constitue un *spectre brut*. Ce spectre brut contient des informations concernant les ions mesurés (à l'aide d'une mesure MS ou MS/MS), mais également des signaux issus de *bruit*.

L'amélioration du signal brut obtenu comporte principalement la correction de la ligne de base et le lissage du signal (réduction du bruit).

Le bruit peut être de deux types différents. Le bruit chimique peut venir de contaminants introduits pendant la préparation des échantillons, de protéines ou peptides introduits par erreur, ou bien de processus de

fragmentation mal contrôlés. Ces processus de fragmentation dépendent de la méthode d'ionisation choisie et produisent des pics non désirés. Le bruit électronique est le résultat de perturbations électroniques et se manifeste par des oscillations aléatoires du signal.

La ligne de base d'un signal de spectrométrie de masse correspond à un décalage de l'intensité mesurée pour les différents ratios masse sur charge. Ce décalage est généralement dû au bruit chimique et nécessite de recaler le spectre pour que la ligne de base du signal corresponde effectivement à une intensité quasiment nulle. Ce faisant, il est nécessaire de soustraire ce décalage des intensités des ions mesurés.

Le signal brut obtenu en spectrométrie de masse est souvent dentelé, ce qui peut compliquer sensiblement l'identification de pics. Des algorithmes de lissage (moyenne pondérée sur des fenêtres glissantes, filtre Gaussien, transformées de Fourier, transformées en ondelettes) permettent d'obtenir à partir de ce signal brut une courbe plus lisse, et ainsi faciliter l'identification des pics.

Extraction des pics

Afin de pouvoir exploiter convenablement l'information produite par le spectromètre de masse, il est nécessaire de retraiter le signal pour obtenir une *peak list*, c'est-à-dire une liste de *pics* extraits du signal qui correspondent effectivement aux ions mesurés. On cherche lors de ce processus à améliorer la qualité du signal obtenu, à en extraire les pics qui correspondent aux ions mesurés et enfin à améliorer ces pics pour en extraire l'information la plus fiable et la plus complète possible.

L'extraction des pics comporte la détection des pics, la normalisation de leur intensité et leur calibration.

Détecter les pics consiste à les séparer du bruit. Idéalement, on veut obtenir une représentation de chaque fragment par une unique masse et son intensité associée. Il existe plusieurs façons de procéder : on peut par exemple sélectionner uniquement l'apex d'un pic, ou bien calculer la masse résultant du signal mesuré avec des algorithmes spécifiques, lesquels prennent en comptent les caractéristiques du spectromètre de masse utilisé. Généralement, les pics isolés et de forte intensité sont détectés sans erreur, mais les pics d'intensité faible ou les groupes de pics résultant de chevauchements de signaux sont bien plus complexes à traiter, et les erreurs bien plus fréquentes.

La normalisation de l'intensité a pour objectif d'homogénéiser l'intensité, qui peut être fortement variable d'un spectre à l'autre, et de l'exprimer relativement à l'intensité totale (le nombre total d'ions détectés) ou l'intensité maximale mesurée. Il est important que cette normalisation reflète la probabilité qu'un pic soit du bruit ou bien corresponde à un ion fragment mesuré, mais aussi qu'elle conserve les relations entre les pics. Une fois l'intensité normalisée, il est possible de s'en servir pour éliminer les pics dont l'intensité est en-dessous d'un seuil choisi, en les considérant donc comme étant du bruit.

La calibration permet de rectifier l'erreur de masse systématique que le spectromètre de masse peut faire lors de la mesure. Cette calibration est généralement réalisée grâce à une courbe de régression établie en utilisant un standard incorporé à la manipulation, en utilisant des pics caractéristiques de l'enzyme de digestion utilisée (ce sont des calibrations internes) ou bien en mesurant un standard après manipulation (calibration externe, généralement moins précise).

Pré-traitement de la peak list

A l'issue du traitement du signal, une liste de masses, appelée peak-list, est restituée en sortie du spectromètre de masse. L'étape de pré-traitement de cette peak list permet d'affiner les pics à restituer en traitant les pics dûs aux *isotopes* et en enlevant les pics douteux.

Certains éléments comme le carbone 13 (dont la masse est supérieure à celle du carbone 12, le plus courant) sont présents en quantités significatives dans les acides aminés, et vont par conséquent entraîner la présence de pics supplémentaires légèrement décalés par rapport à celui de l'isotope principal. On nomme enveloppe isotopique l'ensemble des pics produits par les ions qui contiennent des atomes isotopes. L'étape de pré-traitement va choisir ou calculer un pic représentant ces isotopes et éliminer les autres.

Il est aussi possible à cette étape de supprimer des pics qui correspondent à des *contaminants* usuels ou dont la masse ne peut pas correspondre à un ion. En effet, l'ensemble des masses que peuvent prendre les ions (sommes des masses des atomes qui les composent) n'est pas continu, et la masse de certains pics observés n'est pas cohérente avec les valeurs possibles.

3.1.5 Représentation des spectres de masse

Les spectres de masse MS et MS/MS enregistrés par le spectromètre sont restitués sous forme de listes de masses, accompagnées de valeurs d'intensité et d'informations complémentaires concernant l'ion mesuré (charge, masse de l'ion précurseur, temps de rétention, etc.). Ces spectres sont appelés *spectres expérimentaux*.

On peut représenter les spectres expérimentaux (MS ou MS/MS) par des graphes en deux dimensions comportant les ions mesurés par le spectromètre. On trouve sur l'axe des abscisses la masse et en ordonnée l'intensité associée, c'est-à-dire l'abondance relative de l'ion mesuré. La mesure de masse est exprimée en Dalton, tandis que l'intensité est exprimée en unité arbitraire. La Figure 3.8 fournit une représentation d'un spectre de masse MS/MS.

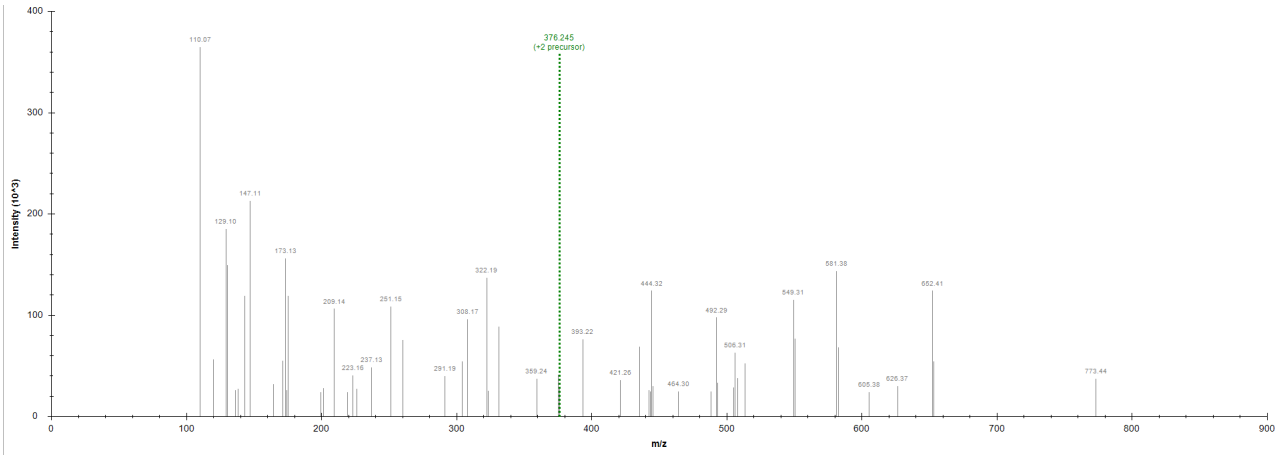


FIGURE 3.8 : Exemple de spectre de masse MS/MS produit par une expérience de spectrométrie de masse en tandem.

Il est improbable qu'un spectre de masse expérimental contienne la totalité de l'information concernant les ions fragments qui peuvent être générés par un peptide. Certains fragments peuvent être manquants parce qu'ils n'ont pas retenu de charge électrique et n'ont donc pas pu être détectés par le spectromètre, ou bien sont trop peu abondants et donc difficiles à discerner du bruit. Des difficultés liées au processus de re-traitement du signal effectué impactent également les spectres expérimentaux. Plusieurs pics peuvent avoir été discrétisés en un seul, ou bien un signal analogique étiré peut avoir été divisé en deux pics successifs. La présence d'un pic dans un spectre ne peut pas toujours être expliquée simplement, de même que l'absence d'une masse ne signifie pas systématiquement une absence de l'ion. Il est en outre possible que plusieurs ions soient sélectionnés simultanément lors de la sélection des ions précurseurs (en raison de masses très similaires) et le spectre MS/MS peut alors comporter l'empreinte des ions fragments des deux ions parents.

En conséquence, les spectres expérimentaux contiennent une information incomplète et imparfaite, ce qui complique en général leur identification.

Maintenant que nous avons décrit dans son ensemble le processus d'acquisition des spectres expérimentaux, nous allons décrire les approches algorithmiques existantes pour l'analyse de ces spectres.

3.2 Identification des spectres expérimentaux

Après la production des spectres expérimentaux par le spectromètre de masse, plusieurs étapes d'analyse sont nécessaires pour en déduire les protéines présentes dans le mélange. En effet, ces protéines ont été découpées par digestion enzymatique en peptides et ces peptides ont généré les spectres expérimentaux. Il faut donc opérer le processus inverse : retrouver le peptide qui a conduit à la génération de chaque spectre expérimental (l'identification de peptide) puis assembler ces peptides pour retrouver les protéines dont ils sont issus (l'inférence de protéines).

Formellement, identifier un spectre de masse consiste à déterminer tout ou partie de la séquence d'acides aminés la plus vraisemblablement responsable de la génération de ce spectre par un spectromètre de masse. Le résultat est retourné sous la forme d'un couple (spectre, peptide) qui sera dénommé PSM (Peptide Spectrum Match) dans la suite de ce document.

A l'origine, l'identification des spectres de masse était réalisée exclusivement manuellement : c'est un travail d'expert chronophage qui peut être particulièrement compliqué. L'outil informatique a permis d'automatiser le processus d'identification. Aujourd'hui, plusieurs méthodes informatiques coexistent pour l'analyse de spectres de masse : les approches de novo, qui travaillent à partir du spectre expérimental pour reconstituer sa séquence, et les approches par comparaison, qui comparent le spectre expérimental à d'autres spectres issus de bases de données de spectres expérimentaux ou de modèles théoriques. Une fois obtenues, les identifications de peptides sont statistiquement validées, puis utilisées dans le processus d'assemblage pour fournir la liste de protéines probablement présentes dans le mélange d'origine.

3.2.1 L'interprétation de novo d'un spectre : manuelle ou par reconstruction itérative

L'interprétation de novo d'un spectre de masse MS/MS consiste à associer des acides aminés aux ions fragments présents dans le spectre sans comparaison avec une banque de spectres. Lorsqu'elle est réalisée manuellement, cette opération nécessite l'utilisation de la représentation graphique du spectre de masse. L'objectif est d'attribuer un ou plusieurs acides aminés à certains intervalles de pics de cette représentation, lorsque l'écart de masse entre les pics correspond à la masse de la séquence d'un acide aminé ou à la masse d'une petite combinaison d'acides aminés. L'interprétation manuelle d'un spectre expérimental est un travail très long et souvent fastidieux, malgré l'existence d'un ensemble de méthodologies d'interprétation qui aident à trouver un pic non ambigu à partir duquel commencer ce travail et qui facilitent l'attribution des ions [129]. De nombreux outils ont été développés pour faciliter cette interprétation manuelle, dont beaucoup sont intégrés dans les suites logicielles du marché.

De nombreux logiciels ont été développés pour interpréter les spectres de novo en utilisant une reconstruction itérative. La plupart des méthodes reposent sur l'utilisation de graphes [48, 134, 31, 49] : chaque pic du spectre est représenté par un sommet d'un graphe ayant pour étiquette la masse du pic. Des arêtes relient deux sommets du graphe s'il existe une combinaison d'acides aminés dont la masse permet d'expliquer la différence de masse observée entre ces deux sommets. Le graphe ainsi construit est nommé graphe spectral, un concept proposé en 1990 par Bartels [11]. Les peptides qui expliquent les pics présents dans le spectre

correspondent à un chemin dans le graphe spectral. Parmi ces logiciels, SeqMS [48] a pour avantage principal d'optimiser le temps de parcours du graphe. Sherenga [31] et PepNovo [49] effectuent une évaluation probabiliste des sommets du graphe et utilisent un système de score plus élaboré pour privilégier la fiabilité des résultats obtenus. L'utilisation de la programmation dynamique est exploitée par plusieurs groupes de chercheurs [20, 92, 9, 95] pour approximer efficacement le chemin optimal dans le graphe spectral. Les techniques basées sur le graphe spectral souffrent néanmoins de l'absence de certains fragments, ce qui rend impossible le parcours du graphe du premier au dernier fragment. PepNovo est capable de travailler avec des tags pour identifier plus facilement des portions de peptides, mais cette approche a l'inconvénient de compliquer l'étape d'assemblage des protéines en rendant plus difficile la discrimination de protéines aux séquences proches. Plus récemment, pNovo [22] tire parti d'une amélioration significative de la précision sur la mesure des fragments effectuée par les spectromètres de masse pour permettre une meilleure exploitation des sommets du graphe, en diminuant les risques de confusion des pics. Sous une version améliorée, pNovo+ [21] exploite deux spectres de masse produits avec des modalités différentes pour bénéficier de renseignements complémentaires sous la forme d'ions caractéristiques. Très récemment, open-pNovo [142] a apporté à pNovo une nouvelle fonction de score pour discriminer les identifications et utilise une nouvelle méthode de programmation dynamique pour mettre en évidence les chemins d'intérêt dans le graphe spectral.

Qu'elle soit manuelle ou automatisée, la fiabilité d'une interprétation de novo repose grandement sur la qualité du spectre : si beaucoup de pics manquent, il devient alors très complexe d'identifier le motif de fragmentation et de lui attribuer des acides aminés. De même, si le spectre comprend trop de bruit, le risque d'interpréter un pic provenant du bruit comme l'une des extrémités d'un intervalle est augmenté. Cette stratégie limite donc malheureusement le nombre de peptides identifiables.

3.2.2 L'interprétation à l'aide de spectres déjà identifiés

Plutôt que de chercher à retrouver la séquence à partir de l'observation du spectre généré par le spectromètre de masse, il est possible de comparer directement le spectre expérimental avec d'autres modèles de spectres, lesquels sont déjà associés à une identification. Les spectres modèles utilisés peuvent provenir d'une collection de spectres expérimentaux préalablement identifiés (les bibliothèques spectrales) ou bien d'une collection de spectres générés à partir de collections de peptides. Pour chaque spectre expérimental, l'objectif de l'interprétation est alors de trouver parmi les spectres modèles celui qui lui ressemble le plus.

Mesure de distance entre spectres

Pour quantifier la ressemblance entre deux spectres, on calcule un score en utilisant une fonction d'évaluation. Il existe de nombreuses fonctions d'évaluation différentes qui permettent de mesurer la ressemblance entre deux spectres, et toutes comportent des spécificités en fonction de leur contexte d'utilisation. La conception et l'amélioration de ces fonctions d'évaluation sont des tâches complexes [99, 8, 65].

La méthode la plus simple et la plus rapide pour calculer le score de similarité entre deux spectres est de compter le nombre de pics en commun (appelé aussi Shared Peak Count). Cette méthode consiste simplement à compter le nombre de pics de même masse : plus ce nombre est grand, plus les deux spectres sont supposés similaires. Deux pics sont considérés communs si leur masse est la même, modulo une petite tolérance qui correspond à la prise en compte de l'incertitude de mesure du spectromètre de masse utilisé. Ce score est souvent critiqué en raison de la prise en compte équivalente de tous les pics [90, 104]. Le calcul du score peut être enrichi par des informations supplémentaires, telles l'intensité des pics, leur séquentialité (comme par exemple le score calculé par le logiciel XTandem ! [27]), leur complémentarité de fragmentation, etc.

D'autres systèmes de score plus complexes font intervenir des évaluations probabilistes concernant les chances que des fragments présents dans les spectres de masse soient partagés entre deux spectres par hasard, comme c'est par exemple le cas du très connu logiciel Mascot [107], basé sur la fonction de score MOWSE [106]. SCOPE [8] est un autre bon exemple de logiciel qui utilise un modèle de score probabiliste : une étape d'évaluation stochastique en deux passes est effectuée pour chaque paire de spectres à évaluer, qui prend en compte la probabilité d'occurrence des fragments d'ions, le bruit du spectre et l'erreur de mesure de l'appareil. Il est également possible, comme le fait Sequest [43], de complémenter l'utilisation d'un score basé sur le Shared Peak Count en travaillant directement avec le signal produit par le spectromètre de masse pour calculer la corrélation croisée entre les spectres, afin d'estimer la distance entre ces derniers. Cette approche a néanmoins le défaut d'être très coûteuse en temps de calcul, et était initialement réservée aux 500 spectres les plus fréquemment analysés. Toutefois, cette fonction a dû être améliorée à plusieurs reprises pour faire face à l'augmentation du volume de données manipulées en protéomique [42, 36].

Comparaison avec les bibliothèques spectrales

L'introduction du concept de bibliothèques spectrales en protéomique est dû à Yates et al. [144] en 1998, et est consécutive à l'observation d'une reproductibilité suffisante des spectres expérimentaux d'une expérience de spectrométrie de masse à l'autre. En 2006, Frewen et al. publient Bibliospec [51], une bibliothèque spectrale qui permet d'identifier des spectres produits par différents modèles de spectromètres de masse dans différents laboratoires. C'est également l'année du développement de X !Hunter [28] par Craig et al., un moteur de recherche de spectres pour les bibliothèques spectrales (par analogie à X !Tandem, dédié aux ensembles de spectres théoriques). En 2007, Lam et al. développent SpectraST [88], un outil de recherche de bibliothèque spectrale intégré à TransProteomic Pipeline (TPP) [81], qui permet de réaliser toutes les étapes de l'analyse de données au sein d'un framework unifié. On observe alors un effort pour créer des libraires spectrales, générer et partager les données nécessaires [28, 35]. Les bibliothèques spectrales ont depuis fait l'objet de nombreuses recherches : nouvelles fonctions d'évaluation, amélioration des algorithmes de recherche, validation statistique des identifications, etc.

Le principal avantage des bibliothèques spectrales est de contenir majoritairement des spectres dont on sait qu'ils sont observables par spectrométrie de masse en tandem. Ces spectres possèdent en outre des propriétés reproductibles sur des appareils de même technologie (en particulier l'intensité) mais dont les mécanismes sont encore mal compris et que l'on peut donc difficilement anticiper. Ces bibliothèques spectrales se prêtent bien à la fouille de données et permettent par exemple à d'autres approches d'identification de tenter de simuler l'intensité des pics [40, 77] en se basant sur les données existantes, ou bien de réaliser de la prédiction de fragmentation [7, 150, 149]. Pour ces raisons, et parce que les spectres présents dans les bibliothèques sont aussi limités à des spectres observables, les bibliothèques spectrales permettent d'effectuer des recherches qui produisent moins de résultats dus au hasard [28].

Les bibliothèques spectrales comportent néanmoins des inconvénients significatifs. Tout d'abord, leur construction est relativement complexe, car elles nécessitent l'accès à un nombre important de spectres convenablement identifiés. Elles sont sujettes à la propagation d'erreurs lorsque des spectres erronés ou fortement contaminés (comme lorsque les fragments de plusieurs ions parents figurent dans le même spectre) sont inclus dans la bibliothèque [119]. De plus, la taille de ces bibliothèques tend à croître rapidement : outre l'augmentation du risque d'inclusion d'identifications erronées, l'accroissement du volume des bibliothèques pose d'importantes difficultés de chargement en mémoire (volume et temps d'accès au disque) et restreint l'efficacité des algorithmes de recherche. Enfin, ces bibliothèques ne sont pas bien adaptées à la recherche de peptides qui ne sont pas aisément détectables par spectrométrie de masse (peptides minoritaires) ou bien de modifications chimiques rares. De même, elles ne peuvent pas être utilisées sur des organismes très peu étudiés car les données nécessaires à la construction de ces bibliothèques sont alors généralement manquantes.

3.2.3 Interprétation à l'aide de spectres théoriques

La comparaison des spectres expérimentaux avec des modèles appelés spectres théoriques est certainement aujourd'hui la méthode la plus utilisée pour identifier les spectres générés. En effet, il est possible de construire in silico un modèle de fragmentation à partir de n'importe quel peptide. On utilise pour cela la séquence d'acides aminés de ce peptide, à partir de laquelle on construit un ensemble de masses. Un tel modèle de fragmentation est appelé spectre théorique : c'est une simulation de la fragmentation idéale du peptide.

Les spectres théoriques sont construits à partir des séquences de peptides provenant de banques de protéines ou de données provenant d'autres méthodes d'analyse de protéines. Généralement, on choisit un ensemble de protéines qui correspond le plus possible à l'ensemble de protéines que l'on suppose présentes dans le mélange analysé, d'une part pour que le nombre de spectres à construire reste raisonnable, et d'autre part pour éviter un grand nombre de résultats dûs au hasard. On rajoute généralement à ces protéines choisies un ensemble de contaminants connus (comme par exemple ceux fournis sur le dépôt common Repository of Adventitious Proteins du GPM (ftp://ftp.thegpm.org/fasta/cRAP/crap.fasta) que l'on retrouve souvent lors des expériences de spectrométrie de masse.

Pour construire les spectres théoriques, on réplique dans un premier temps l'action de l'enzyme de digestion utilisée lors de l'expérience de spectrométrie de masse. Les protéines sont découpées in silico en peptides théoriques, en fonction des sites actifs que leur séquence contient et qui correspondent à cette enzyme. L'ensemble de peptides théoriques obtenu peut alors être filtré pour retirer certains candidats (typiquement des séquences peptidiques trop courtes ou trop longues), et peut également servir à générer des candidats supplémentaires (par exemple pour simuler la présence de missed-cleavages). Pour chaque peptide de l'ensemble obtenu après filtrage, il est alors possible de construire un spectre théorique en appliquant les règles connues de fragmentation. Il existe des méthodes qui permettent de réaliser la comparaison de spectres à l'aide de ces principes sans répliquer la digestion enzymatique, mais elles souffrent d'une lenteur accrue [93].

Construction de spectres théoriques

Le spectre théorique contient des masses qui correspondent aux ions associés à chaque fragment potentiel, produit à partir de la séquence d'acides aminés, lors du processus de fragmentation. En fonction des besoins, tous les ions ne sont pas modélisés dans un spectre théorique : on peut par exemple décider de ne modéliser que les ions de type b et y, ou bien de réaliser une modélisation pour un peptide avec un nombre de charges choisi. Ce spectre étant vu comme un modèle idéal, on considère néanmoins que pour la configuration choisie, l'intégralité des masses associées aux ions sont présentes (pas de masses manquantes), et que le spectre théorique ne comporte aucun bruit.

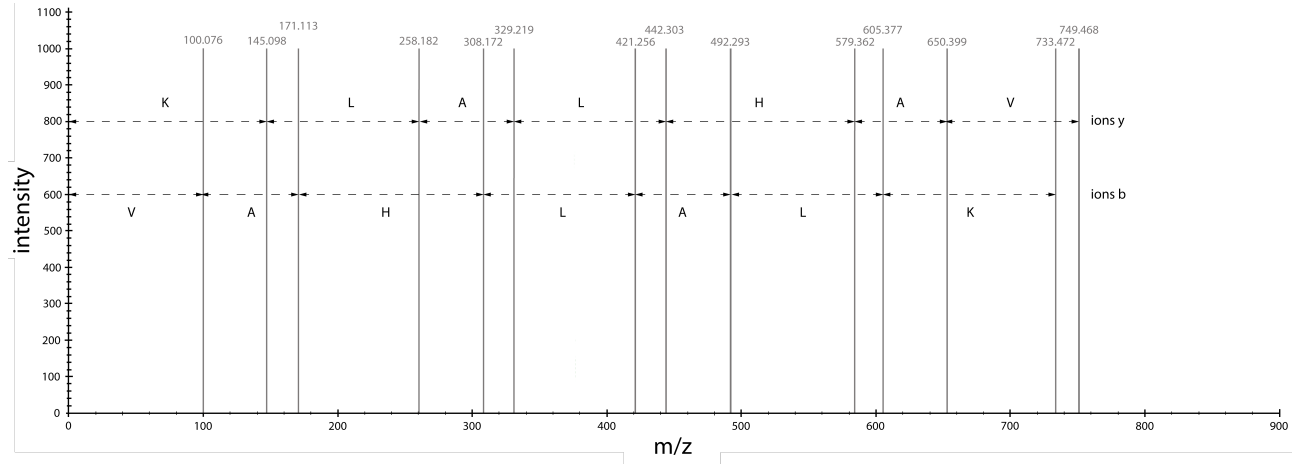


FIGURE 3.9 : Spectre théorique construit *in silico* en utilisant les règles de fragmentation des ions pour la séquence VAHLALK. Les ions simulés sont les ions b et y chargés une fois.

Les spectres théoriques ne sont pas des représentations réalistes des spectres de masse, comme on peut le voir sur la Figure 3.9. Il leur manque en effet plusieurs caractéristiques des spectres expérimentaux produits par un spectromètre de masse. Outre l'absence de bruit et la non modélisation des fragments manquants, ils ne comportent pas non plus de simulation de l'intensité dans la grande majorité des cas, même s'il existe des tentatives pour pallier ce problème [40, 77]. En effet, les mécanismes à l'origine de l'abondance des fragments ne sont pas encore complètement bien compris, et les pics sont de ce fait représentés avec une intensité arbitraire.

Comparaison avec la banque de spectres théoriques

Comparés aux spectres issus des bibliothèques spectrales, les spectres théoriques produits à partir de banques de protéines apportent moins d'information pour l'identification, car ils ne comportent pas l'intensité des pics. En revanche, utiliser des banques de protéines permet de générer des spectres théoriques correspondant à des peptides qui n'ont jamais été identifiés par le passé. En outre les banques de protéines ne reproduisent pas les erreurs des bibliothèques spectrales liées à l'incorporation d'identifications erronées. Les bases de protéines ne sont néanmoins pas à l'abri d'erreurs lors du processus de traduction depuis les séquences génomiques, lesquelles se répercutent sur les spectres théoriques (d'où l'intérêt de bases manuellement validées comme SwissProt https://www.uniprot.org/uniprot/?query=reviewed:yes). En outre, elles sont d'un intérêt limité dans le cas d'organismes peu séquencés. En effet, le génome de ces organismes étant alors méconnu, il n'est pas possible de l'utiliser pour générer un protéome complet fiable. Dans ces conditions, il peut souvent exister des organismes proches qui peuvent partager certaines protéines, mais pas toutes, avec l'organisme d'intérêt. Finalement, les spectres théoriques peuvent être construits à la volée lors de leur utilisation et ne sont donc pas stockés (seules les séquences des peptides utilisés le sont), ce qui représente un avantage au regard de la volumétrie des bibliothèques spectrales.

Il est possible de combiner les approches utilisant les deux types de données (bibliothèques spectrales et banques de protéines) pour en gommer les inconvénients. Par exemple, dans [2], les résultats obtenus après recherche dans une banque de protéines avec le logiciel commercial Phenyx [25] sont complémentés par une seconde recherche dans une bibliothèque spectrale effectuée avec SpectraST [87]. Cette combinaison reste néanmoins compliquée à intégrer de manière reproductible dans un pipeline d'exécution [119].

3.2.4 L'interprétation à l'aide de méthodes hybrides

Des méthodes combinant à la fois l'interprétation de novo et la comparaison de spectres ont également été mises au point. L'énumération exhaustive [116, 60] consiste à produire l'ensemble des peptides qui peuvent

correspondre au spectre de masse à identifier. Il est en effet en théorie possible de construire in silico toutes les séquences de peptides qui correspondent à la masse observée de l'ion parent à l'origine du spectre. Le spectre théorique qui obtient le meilleur score est en pratique celui qui a le plus de chance de correspondre au peptide dont est issu l'ion parent.

Cette méthode rencontre rapidement des difficultés lors de sa mise en pratique. Elle doit effectivement faire face à une combinatoire énorme, les peptides pouvant avoir une masse variant d'un millier de Daltons à quelques milliers de Daltons (soit quelques dizaines d'acides aminés). Même à l'aide de l'outil informatique, il devient rapidement prohibitif d'énumérer toutes les combinaisons de peptides possibles pour une masse totale donnée. Les programmes qui utilisent cette approche se tournent donc vers des méthodes réduisant la complexité combinatoire en évitant l'énumération totale des candidats. Parmi les critères de filtre exploités, on peut noter l'utilisation des ions immonium, des ions marqueurs qui témoignent de la présence de certains acides aminés dans la séquence sans pour autant donner leur position. Ce filtre permet de réduire la combinatoire totale en connaissant à l'avance certains acides aminés qui se trouvent dans les séquences des peptides. Il est aussi possible d'utiliser des heuristiques (comme par exemple des algorithmes génétiques) pour diriger la génération de candidats d'intérêt, qui seront ensuite évalués par une fonction de score [68], ou bien de générer des candidats qui ont une forte probabilité d'exhiber un bon score après évaluation, comme le fait le logiciel propriétaire PEAKS [94] en utilisant la programmation dynamique.

Le sous-séquençage [15, 74, 122, 78] est quant à lui une approche se situant à mi-chemin entre l'énumération exhaustive de peptides et la reconstruction itérative déjà présentée. Plutôt que de chercher à construire toutes les séquences possibles comme dans le cas de l'énumération exhaustive, de courtes séquences d'acides aminés (les tags) sont utilisées pour identifier quelques pics dans de petites zones du spectre. Une fois le spectre couvert par suffisamment de tags, les acides aminés manquants entre les tags sont reconstitués un à un ou bien servent de point de départ pour une recherche dans une banque de spectres. Bien qu'il apporte une solution à l'explosion combinatoire des séquences existantes pour un même ion parent, le sous-séquençage souffre d'un manque de sensibilité. En effet, des peptides dont certaines parties correspondent très bien à un ou plusieurs tags peuvent ne pas être retenus par les fonctions d'évaluation lorsque le spectre exhibe certaines parties qui contiennent très peu de fragments, un phénomène courant en spectrométrie de masse.

3.2.5 Optimisations pour traiter les grandes collections de spectres

Lors de l'identification de spectres expérimentaux par comparaison avec d'autres spectres issus d'une collection de spectres, chaque spectre expérimental peut potentiellement être comparé à plusieurs milliers de spectres théoriques ou expérimentaux. Pour évaluer la similarité entre les spectres (cf. Section 3.2.2, page 37), l'exécution de la fonction d'évaluation entre chaque spectre expérimental et tous les spectres de la collection peut être très coûteuse en temps.

Pour limiter le temps de calcul induit par ces comparaisons, différentes méthodes ont cherché à en réduire leur nombre pour chaque spectre expérimental. La méthode la plus utilisée aujourd'hui [27, 107, 43] est de filtrer les spectres de la collection pour comparer chaque spectre expérimental à un sous-ensemble de spectres uniquement, en appliquant une sélection sur les peptides qu'ils représentent. Ces peptides sélectionnés sont appelés *peptides candidats*. En règle générale, les peptides candidats sont sélectionnés en fonction de leur masse grâce au *filtre de masse*, sélection qui peut être effectuée très rapidement. Les peptides ayant une masse proche de celle du spectre expérimental à identifier sont conservés parmi les candidats. Si le spectre théorique qui correspond au spectre expérimental est contenu dans les spectres construits à partir de la banque de protéines, alors idéalement ce spectre théorique doit aussi faire partie des candidats sélectionnés. Le spectre expérimental et l'ensemble des spectres issus des peptides candidats sont alors soumis à la fonction d'évaluation. Le meilleur résultat est retourné comme étant l'identification la plus probable. Il est

également possible de filtrer les candidats dans la collection en utilisant des tags reconstruits par méthode *de novo* à partir du spectre expérimental [50, 131], mais ces méthodes sont plus coûteuses en temps de calcul.

3.2.6 Validation des identifications

Une fois la liste des PSMs (couples (spectre, peptide)) obtenue, il est nécessaire de déterminer ceux qui seront validés au regard du critère de fiabilité souhaité. Même si une p-value peut estimer la fiabilité de chaque PSM à l'aide de différentes méthodes (méthode paramétrique [125], non paramétrique [80], en utilisant des courbes d'ajustement empiriques [54, 85], méthodes de programmation dynamique [84, 5, 71]), cette mesure ne permet pas d'évaluer le pourcentage d'erreurs dans l'ensemble des résultats du fait d'un problème bien connu sous le terme de "problème des comparaisons multiples". Pour pallier cette difficulté, la notion de FDR (False Discovery Rate) a été introduite par Benjamini-Hochberg [13].

En protéomique, la FDR évalue le nombre d'erreurs d'identification parmi les PSMs. La méthode la plus utilisée pour l'évaluer se base sur une banque dite decoy [41] générée à partir de la banque de protéines d'origine. L'objectif de cette banque decoy est de produire des peptides "leurres" qui produiront des PSMs considérés comme des erreurs. La banque decoy peut être construite à partir de la banque de protéines dite target de plusieurs manières : assemblage aléatoire d'acides aminés respectant les occurrences de ces acides aminés, permutation ou inversion des séquences des protéines, modélisation par chaînes de Markov, etc. L'objectif est de préserver la composition générale de la banque d'origine tout en minimisant le nombre de séquences communes entre les deux banques.

Les banques target et decoy peuvent être agrégées avant d'effectuer l'interprétation des PSMs ou bien deux recherches indépendantes peuvent être réalisées, une dans chaque banque. Il n'existe pas de consensus autour de la méthode à utiliser pour la construction de la base decoy, mais il est en revanche admis qu'il est préférable d'agréger les deux bases avant interprétation [41]. Cette agrégation permet en effet une meilleure estimation du taux de faux positifs, par la mise en concurrence des modèles de spectres générés à partir de la base target et de la base decoy.

Les PSMs obtenus lors de l'interrogation de la banque agrégée (ou des deux banques) sont rangées par valeur de score, du plus fiable au moins fiable. A un seuil de score donné, l'évaluation de la FDR se fait en calculant le rapport du nombre de PSMs associés à la banque decoy (considérés comme des erreurs) sur le nombre de PSMs associés à la banque target (considérés comme corrects). En pratique, c'est l'utilisateur qui fixe le seuil de FDR (souvent de 1%), et les logiciels en déduisent le seuil de score de validation. Ensuite, tous les PSMs dont le score est supérieur à ce seuil sont validés : les spectres sont dits identifiés.

3.3 Inférence de protéines

L'*inférence de protéines* constitue la dernière étape du processus d'analyse, à l'issue de laquelle est reportée une liste de protéines vraisemblablement présentes dans le mélange analysé. La présence de chaque protéine est associée le plus souvent à une probabilité pour chaque protéine. Développée en 1993 [145, 75, 97, 67, 106] par plusieurs équipes simultanément, l'identification des protéines était à l'origine réalisée à partir des mesures de MS uniquement. Ces méthodes peuvent toujours servir à identifier des protéines préalablement isolées, mais ne permettent pas l'identification de protéines en mélange dans le contexte des expériences de spectrométrie de masse actuelles. Le processus d'inférence des protéines en spectrométrie de masse en tandem consiste essentiellement à assembler les peptides identifiés à l'issue des étapes précédentes en un ensemble de protéines de confiance.

Le problème d'*assemblage* des peptides en protéines se rapporte en théorie à un problème de couverture d'ensembles minimale (*Minimum Set Covering Problem*) [72] dans lequel on cherche à trouver le plus petit ensemble de protéines tel que tous les peptides identifiés lors de l'analyse soient associés à ces protéines, c'est-à-dire en appliquant le principe de parcimonie (utiliser un minimum de causes pour expliquer un phénomène). Cette formulation repose essentiellement sur deux suppositions : toutes les identifications de peptides effectuées lors des étapes précédentes sont correctes, et tous les peptides ont la même probabilité d'être détectés et identifiés. En réalité, il est très vraisemblable que la première supposition soit improbable et la seconde erronée.

En pratique, deux solutions extrêmes peuvent être mises en œuvre. La première cherche à résoudre le problème de couverture d'ensembles minimale. La seconde relaxe les suppositions sur lesquelles cette formulation du problème repose, et cherche plutôt à reporter la totalité des protéines associées aux peptides identifiés. Choisir la première solution fait courir un risque important d'ignorer des protéines bel et bien présentes dans le mélange. L'existence de protéines associées à un unique peptide (cas appelés *one-hit wonders*) ne fait que compliquer le problème, statuer sur la présence de ces protéines devenant conditionnel à la véracité d'identification de cet unique peptide. La seconde solution n'est pas non plus satisfaisante, en cela qu'elle comporte un risque de reporter des protéines qui n'étaient pas présentes dans le mélange d'origine, parce que leur séquence comprend un peptide identique à celui associé avec une protéine effectivement présente (de tels peptides sont dits *peptides dégénérés*). Les algorithmes qui réalisent l'inférence des protéines (appelés aussi algorithmes d'assemblage) ont pour objectif de trouver une ou plusieurs solutions les plus proches possibles d'une liste de protéines effectivement présentes dans le mélange, ces solutions visant un compromis entre les deux extrêmes théoriques. Ces algorithmes d'assemblage peuvent être classés en deux grandes familles [72] : les algorithmes basés sur les *graphes bipartis* et les modèles à information complémentaire.

Graphes bipartis

Le graphe biparti est un objet mathématique utilisé en théorie des graphes. Ce graphe est composé de deux ensembles de sommets S_1 et S_2, et chaque arête de ce graphe relie un sommet de S_1 à un sommet de S_2. Dans le cas de l'inférence de protéines, ce graphe est utilisé par certains algorithmes d'assemblage. L'un des ensembles de sommets est constitué des protéines, et l'autre ensemble est constitué des peptides identifiés. Deux sommets sont reliés par une arête si le peptide incident est produit par digestion enzymatique à partir de la protéine incidente à cette même arête.

Des algorithmes (DBParser [143], IDPicker [148], MassSieve [124] et LDFA [6]) appliquent le *principe de parcimonie* et cherchent à identifier à l'aide d'heuristiques le sous-ensemble minimal de protéines associées aux peptides identifiés (c'est-à-dire résoudre le problème de couverture d'ensembles minimale). D'autres méthodes utilisent une approche statistique (Qscore [100] ou PROT_ROBE [115]), qui peut être guidée ou non par des paramètres (EBP [108], PANORAMICS [46], MSBayesPro [91]). ProteinProphet [105], qui est probablement aujourd'hui l'algorithme d'assemblage le plus utilisé, appartient à cette dernière catégorie. Il est aussi possible d'opter pour une approche dite optimiste, qui consiste à reporter l'ensemble des protéines pour lesquelles il existe une quantité estimée suffisante de peptides associés (DTASelect [130]).

Information complémentaire

Les modèles à information complémentaire utilisent des données supplémentaires pour fournir une solution au problème d'inférence de protéines qui met l'accent sur la qualité de la couverture des protéines. Cette information peut provenir de données générées par l'expérience de spectrométrie de masse, ou bien de sources extérieures.

On trouve parmi les données générées par la spectrométrie de masse et exploitées par les algorithmes d'assemblage les données de la mesure MS, les données de la mesure MS/MS et le profil d'expression des peptides. Les données MS/MS brutes sont utilisées pour valider des suppositions faites lors de l'assemblage (HMS [120]) ou exploitées dans des modèles basés sur un graphe tripartite (spectres, peptides, protéines) pour nourrir un algorithme d'apprentissage (Barista [126]). Ce même modèle peut être utilisé pour assigner à chaque spectre des groupes de peptides en fonction des protéines identifiées (Scaffold [118]). La combinaison avec les données de la mesure MS permet d'améliorer l'information générée par les one-hit wonders. Enfin, le profil d'expression des peptides, généralement utilisé en quantification, peut être étendu à l'identification (PIPER [109]).

Ces informations obtenues par spectrométrie de masse peuvent également être corrélées avec des données biologiques de source externe, comme par exemple les réseaux d'interactions entre protéines (CEA [89], MSNet [110]) en se basant sur le postulat que des protéines qui interagissent lors de processus biologiques ont de fortes chances d'être présentes simultanément dans les mêmes échantillons. Des données moins directement liées peuvent aussi être utilisées, tels les ARN messagers (MSpresso [111]) ou des données génétiques (PeptideClassifier [58], MIPGEM [55]).

En dépit de nombreux efforts de recherche, le problème de l'inférence de protéines par spectrométrie de masse en tandem est loin d'être résolu. C'est un processus complexe comportant de nombreux compromis, dont la bonne qualité et les résultats dépendent étroitement de l'ensemble des étapes qui précèdent, et en particulier de l'identification des peptides. Des identifications de peptides manquantes, ou obtenues avec une faible confiance, conduisent à une plus faible quantité de protéines identifiées et à des ambiguïtés dans l'identification, notamment en raison de la présence de peptides dégénérés. Aujourd'hui, les algorithmes d'identification de peptides souffrent de plusieurs lacunes, en particulier lorsqu'il s'agit d'identifier des peptides portant des modifications chimiques non anticipées. Nous revenons plus en détail sur ces difficultés d'identification dans le prochain chapitre, où nous exposons également des développements alternatifs qui tentent de résoudre ces difficultés, avant d'avancer notre proposition.

4

Développement d'une approche sans filtre de masse à priori

4.1 Les limites des méthodes d'identification traditionnelles

Ces dernières années, la recherche en spectrométrie de masse a cherché à comprendre pourquoi tant de spectres demeurent non interprétés avec fiabilité à l'issue de l'étape d'identification des peptides. On estime en effet que jusqu'à 75% des spectres de masse produits lors d'une expérience de spectrométrie de masse en tandem peuvent demeurer non analysés [23]. Il semble néanmoins que ces spectres de masse soient souvent de bonne qualité et proviennent effectivement bien de la mesure d'ions issus de peptides [57]. Les récentes améliorations des spectromètres de masse, notamment en terme de précision sur la mesure des ions fragments, devraient permettre l'identification de ces spectres de masse. Ces spectres restent néanmoins non analysés, ce pour plusieurs raisons [19] : leur charge peut être mal déterminée, la banque de protéines recherchée incomplète, ou bien les paramètres de recherche excluent le bon peptide candidat. Le cas complexe des modifications variables appartient à cette dernière catégorie : une modification non anticipée qui altère un peptide réduit grandement les chances d'identifier convenablement ce peptide à l'aide des algorithmes d'identification actuels. Malheureusement, on estime que pour chaque peptide présent après digestion, il y aurait 8 à 12 formes modifiées de ce même peptide [44]. Cette importante quantité de spectres non identifiés expliquerait en grande partie la difficulté rencontrée par les approches algorithmiques traditionnelles [123].

Les approches traditionnelles sont souvent qualifiées de "recherches restreintes" (restricted search) en raison de la présélection de candidats à la comparaison lors de l'identification de spectres expérimentaux (à l'aide d'un filtre de masse ou de tags). De nombreux efforts de recherche ont été faits pour améliorer la sensibilité de ces méthodes et permettre notamment une meilleure identification des modifications non anticipées affectant les peptides analysés. Malgré ces efforts, les résultats ne produisent pas des avancées aussi prononcées qu'espéré.

4.2 Les origines des pertes d'identifications

Deux grandes raisons pour lesquelles des pertes d'identification sont observées en spetrométrie de masse semblent se dégager : les spectres peuvent être de qualité insuffisante, ou bien comporter des modifications. Il n'est évidemment pas exclu que ces deux aspects soient combinés pour certains spectres.

Les spectres de moindre qualité

Comme nous l'avons déjà exposé auparavant dans ce manuscrit, les spectres produits par les spectromètres de masse sont rarement de parfaite qualité et ils nécessitent plusieurs étapes de retraitement algorithmique pour pouvoir être exploités. En dépit de l'amélioration du signal qui est effectuée en amont, les étapes d'extraction des pics et de pré-traitement de la peak list ne sont pas toujours suffisantes pour permettre l'identification des spectres.

Parmi les défauts qui peuvent persister dans les spectres expérimentaux à l'issue de ces étapes de retraitement, on peut notamment noter l'absence d'une ou plusieurs masses. Parce que l'ensemble des fonctions d'évaluation utilisent les masses partagées comme base de calcul, l'absence de certaines masses a un impact important sur la capacité à identifier des spectres. Il est aussi possible que quelques pics très intenses masquent les autres en fonction de la manière dont l'intensité est normalisée et le bruit lissé. Ce phénomène peut conduire à ignorer certaines masses qui devraient en réalité être prises en compte. Enfin, la discrétisation de pics proches et notamment des isotopes peut entraîner des erreurs d'attribution des masses, ou l'algorithme de prédiction de la charge peut proposer une charge erronée.

4.2.1 L'impact des modifications sur les spectres de peptides modifiés

Les peptides peuvent être sujets à des modifications chimiques qui peuvent être d'origine naturelle (modifications post-traductionnelles) et avoir un intérêt biologique important, ou bien êtres dues à des artefacts de manipulation liés à la préparation des échantillons. Les modifications chimiques peuvent être localisées sur les extrémités N-terminale ou C-terminale du peptide, ou n'importe lequel des acides aminés qui le composent. Il est également possible que la séquence du peptide comporte des modifications dues à des mutations, se traduisant par une modification de sa séquence d'acides aminés. En terme d'identification, ces mutations se comportent de la même manière que les modifications chimiques. La présence d'une modification ou d'une mutation dans un peptide complexifie la bonne identification d'un peptide. En effet, la valeur d'une partie des masses va être modifiée, ainsi que la masse totale du peptide.

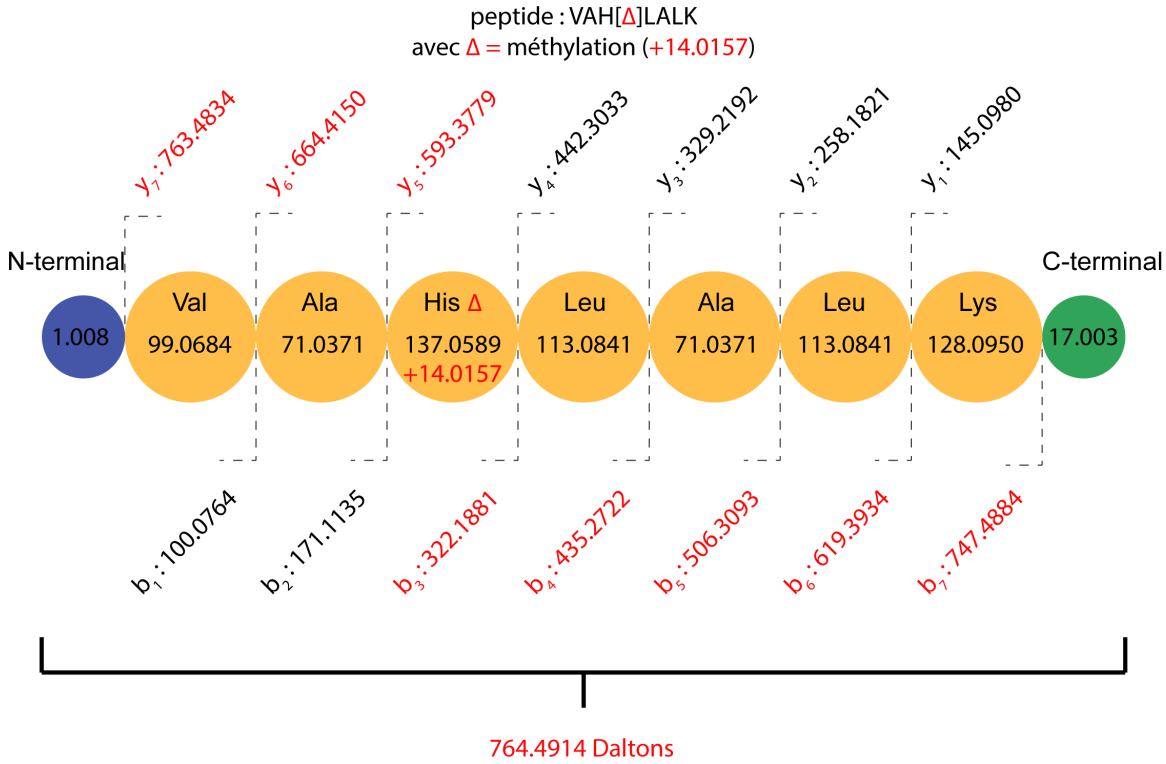


FIGURE 4.1 : Mise en évidence des fragments modifiés par la présence d'une modification post-traductionnelle en position 3 du peptide VAHLALK (méthylation de l'histidine). Les fragments dont la masse est modifiée sont notifiés en rouge. La masse totale est également modifiée.

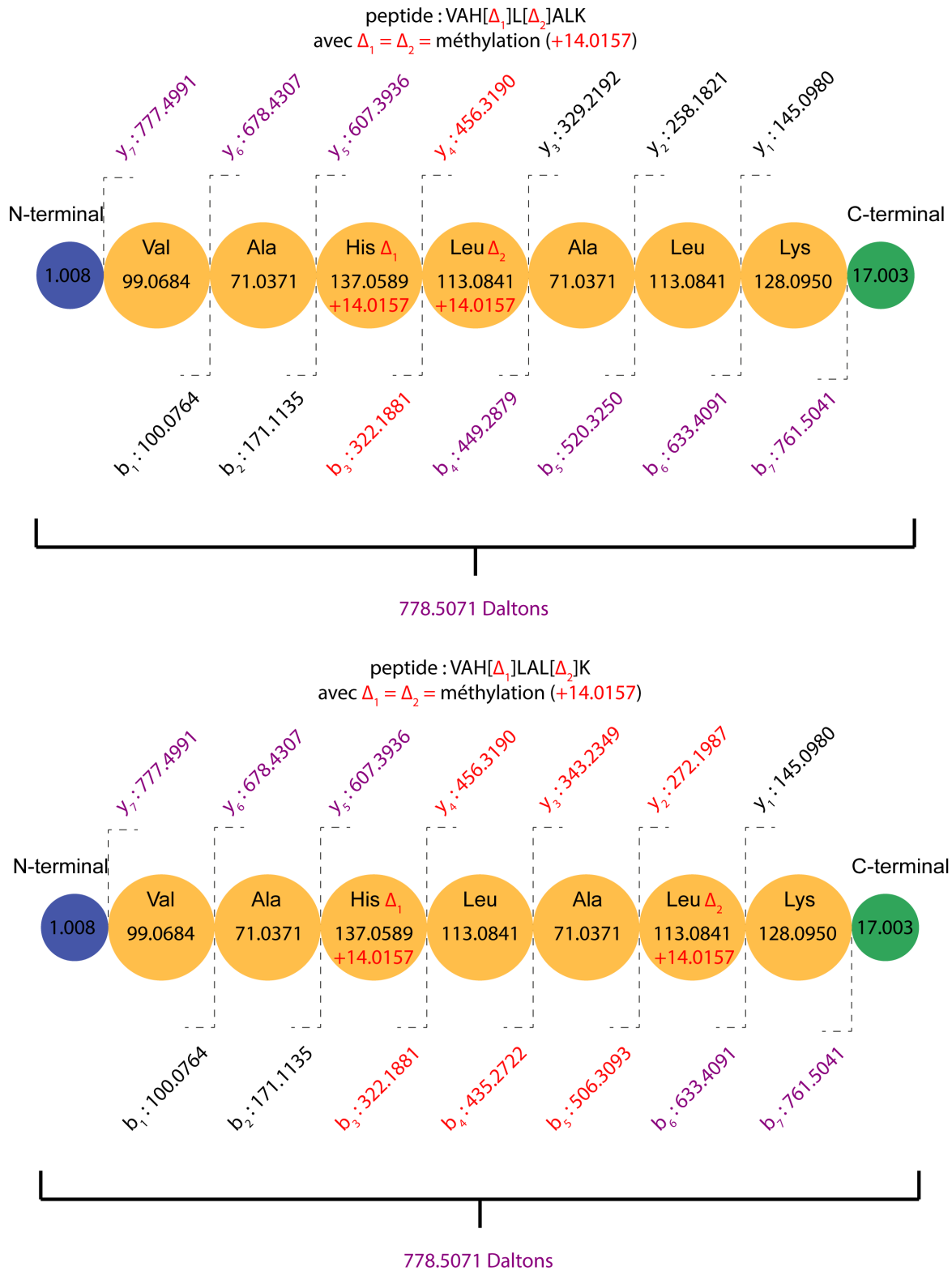


FIGURE 4.2 : Mise en évidence des fragments modifiés par la présence de deux modifications post-traductionnelles dans le peptide VAHLALK. Les fragments dont la masse est modifiée par une seule modification sont notifiés en rouge, ceux dont la masse est modifiée par deux modifications en violet. La masse totale est également modifiée.

La présence d'une modification modifie le spectre obtenu lors de l'expérience de spectrométrie de masse en tandem : non seulement l'intensité des pics peut être différente, mais le motif de fragmentation va surtout varier en fonction de la position de la modification dans la séquence d'acides aminés du peptide. Les fragments suivant la modification (pour les fragments N-terminaux) ou précédant la modification (pour les

fragments C-terminaux) verront de fait leur masse modifiée de manière proportionnelle à la masse de la modification. En effet, lorsqu'une modification affecte un acide aminé, la masse de ce dernier est modifiée. Cette modification se propage sur la valeur des masses suivantes (pour les fragments N-terminaux) ou précédentes (pour les fragments C-terminaux) et modifie en cascade les masses totales de ces fragments. Finalement, pour une unique modification, ce sont 50% des masses du peptide qui sont modifiées. Ce phénomène est illustré Figure 4.1. Plusieurs modifications chimiques peuvent se cumuler dans un unique peptide et affecter des acides aminés différents. Un tel cumul modifie les masses du spectre produit par l'appareil de manière plus marquée que lorsque le peptide contient une unique modification variable, comme illustré en Figure 4.2.

La Figure 4.3 illustre les positions des fragments dans le spectre dans trois cas : sans modification, avec une unique modification, et avec deux modifications. Si l'on observe le spectre sans modifications (a), on peut constater que la majorité des masses du spectre expérimental correspond à une masse dans le spectre théorique, et le nombre de masses en commun s'élève à 11 dans cet exemple. Dès lors que le spectre expérimental comporte une unique modification (b), on peut constater que de nombreuses masses se retrouvent modifiées dans le spectre expérimental et le nombre de correspondances avec le spectre théorique diminue. Enfin, lorsque le spectre expérimental comporte deux modifications (c), ce phénomène s'amplifie et très peu de masses sont encore partagées entre le spectre expérimental et le spectre théorique. Cette diminution est d'autant plus prononcée que les deux modifications sont espacées dans le spectre expérimental (d). Ainsi, un spectre expérimental fortement modifié devient très difficile à identifier si la méthodologie d'identification s'appuie sur le Shared Peak Count.

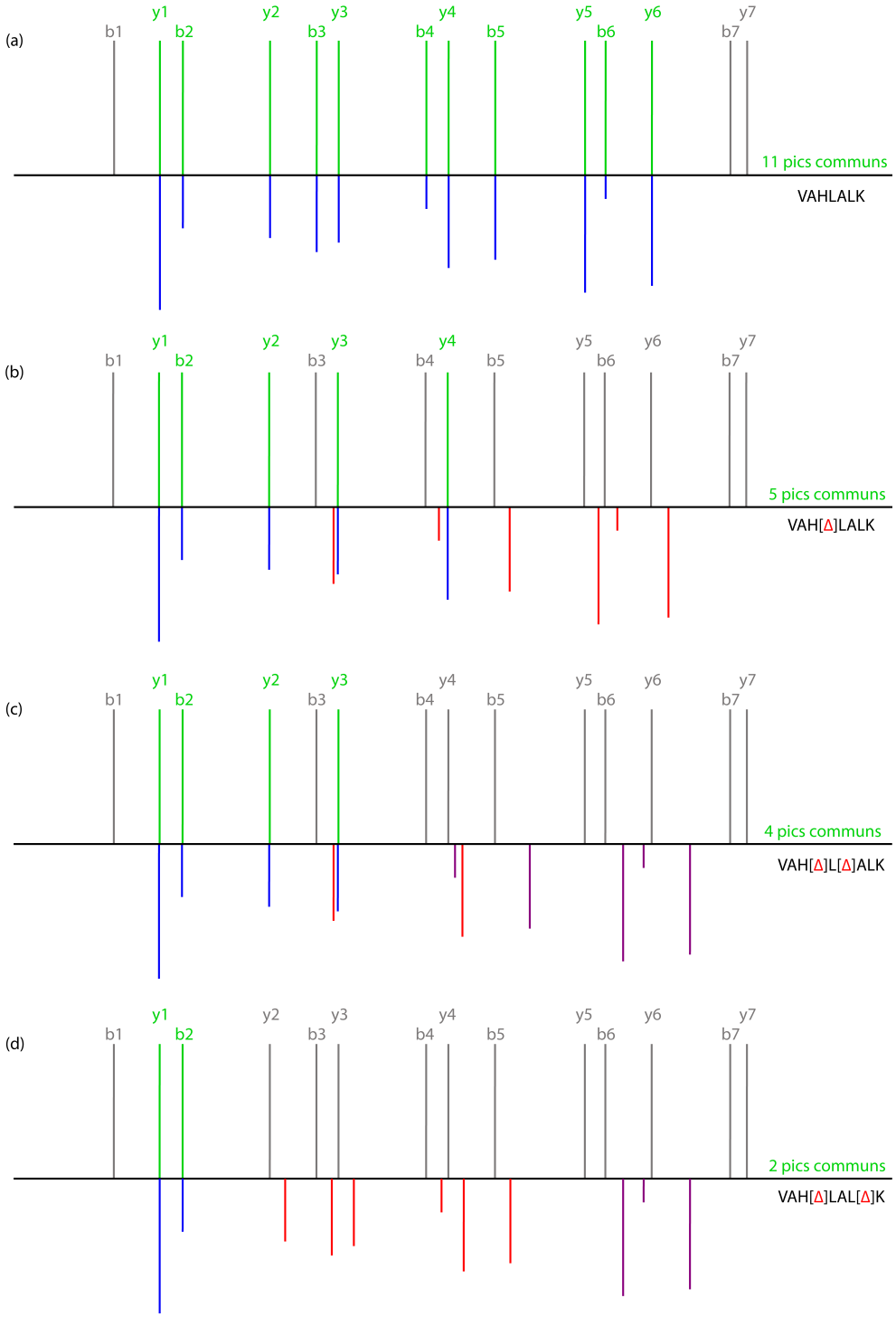


FIGURE 4.3 : Illustration de la modification des positions des masses d'un spectre de masse expérimental en fonction du nombre de modifications et de leur position. Le spectre au-dessus de l'axe des abscisses correspond au spectre théorique du peptide VAHLALK dans lequel les pics en gris correspondent à une absence du fragment et ceux en vert à un fragment détecté. Le spectre en miroir représente les fragments correspondants dans le spectre expérimental, modifiés comme il se doit le cas échéant. Les pics bleus ne sont pas affectés par la modification, les pics rouges sont affectés par une unique modification et les pics violets par deux modifications. Comme on peut le constater, la présence d'une modification entraîne en théorie la disparition de la moitié des pics plus un (sauf collision due au hasard). Lorsque deux modifications sont présentes dans le même peptide, leur impact va dépendre de leur éloignement : si elles sont très proches l'une de l'autre, le nombre de pics ayant une masse modifiée sera proche du cas avec une unique modification. Plus elles seront espacées, plus il y aura de pics déplacés et donc non détectés.

4.3 Conséquences pour les algorithmes d'identification de peptides

4.3.1 Le filtre de masse

La modification de masse totale liée à la présence d'une ou plusieurs modifications chimiques implique un écart de masse entre la séquence du peptide modifié et cette même séquence qui ne porte pas de modification chimique. Par conséquent, la masse totale du peptide théorique, qui ne comprend pas la ou les modifications chimiques, ne peut correspondre à la masse du spectre expérimental mesuré. Ceci est vrai même si la séquence à partir de laquelle le spectre théorique est construit correspond bien à celle du peptide analysé par le spectromètre. Dans ce cas précis, le spectre théorique, dont la masse ne correspond pas à celle du spectre expérimental, ne peut pas faire partie des candidats retenus pour l'identification.

Les modifications post-traductionnelles posent en particulier problème dans le cadre des approches avec filtre de masse et on estime qu'elles ont impact important sur le taux d'identification atteint. Dans le cas de ces recherches restreintes, c'est donc le filtre opéré sur les candidats à la comparaison par la fonction d'évaluation qui est suspecté être l'une des raisons de la sensibilité limitée. Il serait de fait responsable de la défausse de potentiels candidats corrects et générateur d'erreurs d'identifications. La moitié des faux négatifs pourraient en réalité être ainsi dûs à des peptides modifiés [16].

4.3.2 Introduction de modifications variables

Pour pallier l'exclusion de peptides théoriques de l'ensemble des candidats dans le cas de spectres qui correspondent à des peptides modifiés, les approches restreintes proposent d'anticiper les modifications chimiques. Pour ce faire, les modifications à rechercher doivent être expressément spécifiées par l'utilisateur avant la recherche sous la forme de modifications variables. Lors de la recherche, pour chaque modification chimique à anticiper, on ajoutera à l'ensemble des peptides candidats sélectionnés par le filtre de masse les peptides dont la masse totale est proche de la masse du spectre adjoint à la masse de cette modification. L'inclusion d'un trop grand nombre de modifications entraîne par conséquent une explosion du temps de recherche, en particulier si les modifications ajoutées ne sont pas spécifiques de certains acides aminés. Dans ce dernier cas la combinatoire devient en effet très importante. On observe également avec l'augmentation du nombre de modifications variables une augmentation du nombre de faux positifs [3]. Outre ces difficultés, la manipulation des méthodes de validation par FDR se retrouve complexifiée [52] et peut conduire à des validations statistiques dont la fiabilité est questionable.

En pratique, seules quelques modifications sont ajoutées alors qu'il existe une très grande variété de modifications différentes [139]. A titre d'exemple, la base de référence Unimod [29] en contient plus d'un millier. En conséquence, beaucoup de spectres modifiés ne peuvent pas être analysés par les méthodes restreintes. Les approches de filtrage par tags [133, 131] ont moins de difficulté à identifier des portions de spectres, mais elles se heurtent au problème dit de "torture du peptide" : avec suffisamment de portions non identifiées entre les tags, il est possible de trouver de multiples peptides qui correspondent aux tags identifiés.

Ces difficultés se retrouvent fortement exacerbées lorsque plus de deux modifications affectent le même peptide : il est plus difficile de les anticiper, et donc de prévoir à l'avance la masse correcte du peptide. De plus, les fragments sont déplacés de manière moins prévisible, comme illustré sur la Figure 4.2. Il n'est pas surprenant que les méthodes restreintes affichent une sensibilité réduite vu l'impact de ces modifications sur les spectres expérimentaux et l'augmentation de l'espace de recherche qui découle de l'anticipation de nombreuses modifications variables. En outre, toutes les modifications ne sont vraisemblablement pas connues, et les bases qui les référencent ne garantissent en aucun cas l'exhaustivité. Cette incomplétude rend presque impossible la découverte de nouvelles modifications ou la recherche de modifications rares

par les approches restreintes.

4.3.3 Les recherches en plusieurs étapes

Les algorithmes de recherche traditionnels cherchent à contourner ces difficultés en effectuant une recherche en plusieurs étapes [135] (multi-passes). Une étape de recherche restreinte avec filtre de masse est généralement effectuée dans un premier temps en conservant un nombre de modifications variables très limité. Elle est ensuite suivie d'une ou plusieurs étapes de recherche successives pour lesquelles on ajoute généralement une ou plusieurs modifications supplémentaires, ou par exemple des peptides avec missed-cleavages dans la banque de protéines. En revanche, chaque étape de recherche supplémentaire s'accompagne aussi d'une restriction de l'espace de recherche qui prend la forme de contraintes sur la banque de protéines : de manière générale, on ne conserve dans la banque que les protéines identifiées avec succès lors de la première passe de recherche.

Mascot [30] utilise cette approche dans sa méthodologie de recherche tolérante : une première passe de recherche classique est réalisée en spécifiant quelques modifications. Une seconde passe suit, pour laquelle seules les protéines identifiées lors de la première passe sont considérées. Mascot tolère alors la présence de missed-cleavages, la substitution d'acides aminés, et recherche les peptides exhibant une combinaison de plusieurs modifications parmi celles utilisées dans la première passe auxquelles peut être ajoutée une modification supplémentaire issue d'Unimod. PeaksPTM [62], X !Tandem [26] et Phenyx [25] permettent une recherche multi-passes qui infère dans un premier temps un ensemble de protéines présentes dans le mélange. Une seconde recherche qui considère d'autres modifications provenant d'Uniprot est ensuite effectuée sur cet ensemble restreint de protéines en considérant une unique modification par peptide, suivie d'une passe de recherche qui combine plusieurs des modifications courantes identifiées lors des passes préalables. Enfin, Paragon [121] propose dans une alternative intéressante de déterminer des sous-séquences de protéines à examiner intensivement en utilisant des tags et conduit ensuite une recherche restreinte avec une liste complète de modifications.

Cette approche présente néanmoins deux principaux défauts [45, 135, 14] qui diminuent son intérêt. D'abord, elle présuppose qu'il existe pour chaque peptide au moins un spectre non modifié dans le mélange, qui permet d'inférer et de conserver dans la banque les protéines effectivement présentes dans le mélange. Ensuite, l'estimation de la FDR est plus difficile à réaliser et le nombre de faux positifs peut être sous-estimé en raison de l'utilisation d'une banque decoy de taille réduite lors des passes successives.

4.3.4 Les recherches en cascade

En 2013 Huang et Al. proposent ISPTM [73], une méthode dite de recherche en cascade. Le principe est similaire aux recherches multi-passes en cela que plusieurs recherches successives sont effectuées à l'aide d'un logiciel d'identification de peptides traditionnel. En revanche, plutôt que de restreindre la banque de protéines après chaque passe, la recherche en cascade réduit l'ensemble de spectres à analyser. Le principe consiste à réaliser une première passe de recherche avec l'ensemble des spectres sans modification variable pour identifier les peptides non modifiés. Les spectres analysés qui identifient des peptides avec confiance sont retirés de l'ensemble des spectres à analyser. Les spectres restants font l'objet de plusieurs passes de recherche supplémentaires, lesquelles testent des combinaisons d'une ou plusieurs modifications prises dans une liste fournie au logiciel.

Si cette technique a l'avantage de ne pas nécessiter la présence d'un peptide non modifié dans le mélange comme support pour filtrer les protéines inférées lors de la première passe à conserver, elle possède néanmoins ses propres défauts. Les passes de recherche successives proposées par les auteurs sont très intensives en terme de ressources informatiques. En pratique, l'explosion combinatoire associée contraint

l'usage de courtes listes de modifications et l'exploitation de relativement peu de combinaisons différentes. En outre, la mise en oeuvre de cette méthode est lourde et nécessite le développement par l'utilisateur de programmes adaptés. Finalement, cette méthodologie complique le calcul de la FDR, laquelle doit être établie pour chacun des groupes de peptides identifiés lors des recherches successives [82].

4.3.5 Les méthodes OMS

Plusieurs méthodes dites Open Modification Search ont été développées ces dernières années. L'objectif principal de ces méthodes est d'apporter une réponse aux limitations des approches par filtre de masse lorsqu'il s'agit d'identifier des peptides modifiés. Les méthodes OMS sont aussi appelées "unrestricted search" (sous-entendu sans filtrage de candidats) mais en pratique le terme "méthodes hybrides" est plus approprié. Il n'existe en effet à notre connaissance pas de méthode OMS qui ne réalise absolument aucune présélection des candidats en raison de l'explosion combinatoire qui conduit à un espace de recherche trop important.

Les méthodes OMS ont dans un premier temps été utilisées dans le cadre de l'utilisation de bibliothèques spectrales. QuickMod [4] (plus tard optimisé et parallélisé en tant que MzMod [70]) utilise une combinaison d'apprentissage et de fonction d'évaluation adaptée avec une fenêtre de tolérance élargie à ± 200 Da pour identifier des peptides modifiés. Un autre groupe de chercheurs [19] propose d'exploiter la conservation de 50% des fragments entre un peptide non modifié et la même séquence modifiée pour identifier des peptides porteurs de modifications. Par ailleurs, pMatch [146] est un outil qui utilise la notion de décalage des pics observés dans les spectres en utilisant une fonction de score adaptée et une fenêtre de tolérance élargie à ± 300 Da. Cette méthode a ensuite été améliorée par un nouveau système de score [96] basé sur une analyse statistique pour la validation des deltas de masse observés lors d'une recherche en plusieurs passes.

En comparaison, assez peu de développements ont été réalisés en ce qui concerne les approches OMS dans le cadre de l'utilisation de banques de protéines. DeltAMP [53] se base sur une représentation des paires de spectres sous formes de vecteurs comprenant la masse et la durée de rétention des spectres. Ce logiciel, qui repose essentiellement sur une information tirée de l'ion précurseur, et non des ions fragments, part de l'hypothèse qu'une paire constituée d'un spectre modifié et de son pendant non modifié peut être détectée à l'aide d'un modèle statistique. MODa [102] repose sur une présélection de candidats à l'aide de tags suivie d'une étape de programmation dynamique pour aligner ces tags avec des peptides issus de la banque et expliquer les deltas de masse observés. Tout comme MODa, InsPect [137] fait usage de la programmation dynamique pour rechercher les modifications sans les connaître au préalable et implémente une fonction de score qui prend en compte non seulement la présence mais également l'absence d'ions fragments. PIPI [147] repose sur un encodage des peptides sous forme de vecteurs booléens, témoins de présence de triplets d'acides aminés dans ce peptide. Les spectres expérimentaux à identifier sont eux aussi encodés en utilisant les différences de masses entre des triplets de pics intenses qui correspondent à des acides aminés. Le vecteur booléen qui correspond au spectre expérimental est alors évalué par le biais d'une corrélation croisée avec les vecteurs booléens des peptides de la banque pour lesquels la différence de masse totale est comprise dans une fourchette de ± 250 Da. Enfin, MSFragger [86] utilise une répartition des fragments des spectres théoriques en différents ensembles. On peut ainsi rapidement déterminer quels sont les fragments partagés entre les spectres et de minimiser le temps de recherche en restreignant les ensembles analysés grâce à une fenêtre de tolérance réglable autour de la masse du précurseur.

Une façon simple d'effectuer une recherche OMS avec un algorithme par filtre de masse est d'utiliser une fenêtre de tolérance large autour des ions précurseurs sélectionnés et en s'autorisant plus de temps et de puissance de calcul. Cette idée a déjà été utilisée par [17] pour affiner la recherche sur les spectres non identifiés lors d'une première recherche avec filtre de masse en sélectionnant une tolérance de précurseur à ± 200 Da. Cette approche a ensuite été étendue par [23] : deux recherches successives sont effectuées à

l'aide de Sequest [36] sur un jeu de données d'un million de spectres, une recherche restreinte suivie d'une recherche avec une fenêtre de tolérance fixée à ± 500 Da. La première recherche identifie quelques 396 736 spectres, tandis que la recherche plus tolérante en identifie 510 139 (dont 325 157 en commun avec la recherche précédente). Les 184 982 spectres identifiés restants correspondent essentiellement à des peptides modifiés. Cette manipulation démontre à priori la capacité des approches OMS à identifier des peptides modifiés à condition de disposer d'une précision de mesure élevée sur les ions fragments. Cependant, la totalité des identifications obtenues ne représentent toujours qu'une petite proportion du million de spectres composant l'échantillon de départ. En particulier, les auteurs de cette étude estiment que les peptides modifiés identifiés représentent très vraisemblablement moins de 50% du total des peptides modifiés.

Toutes ces méthodes OMS sont critiquées pour les mêmes limitations. Tout d'abord, à part MSFragger sur lequel nous reviendrons (Section 7.4, page 112) ces algorithmes sont difficilement utilisables sur des stations de travail courantes, en raison de leur temps d'exécution élevé et de leur forte consommation mémoire. Alors que les étapes d'interprétation représentent déjà le goulot d'étranglement de la spectrométrie de masse en tandem pour l'identification de protéines, cette demande de performances élevées n'est pas souhaitable. De plus, la sensibilité des approches OMS est contestée : si elles sont en effet capables d'identifier des peptides modifiés inaccessibles aux autres approches, un taux élevé de faux positifs parmi les identifications oblige à abaisser le seuil de confiance pour les identifications. Il en résulte une quantité d'identifications moindre que lors de l'utilisation de méthodes par filtre de masse. Finalement, peu de ces approches ont été pensées pour un usage routinier dans les laboratoires de spectrométrie de masse qui ne disposent pas nécessairement du personnel ou du matériel pour leur mise en application.

4.4 Développement d'une approche OMS alternative

4.4.1 Principe général

FIGURE 4.4 : Illustration de la modification des positions des masses d'un spectre de masse expérimental en fonction du nombre de modifications et de leur position.

On se propose de mettre au point une nouvelle méthode d'identification de peptides entrant dans la catégorie des méthodes OMS. La Figure 4.4 illustre le principe général que l'on souhaite mettre en œuvre pour cette méthode. Pour ce faire, on souhaite ne pas recourir à un filtre de masse lors de la comparaison des spectres expérimentaux générés par spectrométrie de masse en tandem avec les spectres théoriques générés à partir d'une banque de protéines. Ce filtre de masse est à l'origine utilisé pour réduire le nombre de comparaisons à effectuer lors de l'identification des spectres en sélectionnant pour évaluation uniquement un petit sous-ensemble de spectre. Une conséquence logique de la suppression de ce filtre de masse, difficulté rencontrée par les approches OMS, est l'augmentation importante du temps de calcul et de la consommation mémoire. On cherche donc à designer cette nouvelle méthode d'identification de sorte à éviter ce problème.

Avec l'émergence de spectromètres de masse récents dont la mesure sur les ions fragments est de plus en plus précise, nous pensons pouvoir émettre l'hypothèse suivante : il est possible de se baser sur un score simple pour identifier un spectre, à savoir utiliser uniquement le nombre de masses partagées entre deux spectres. Si cette hypothèse est correcte, il est alors possible de représenter les spectres sous forme de listes de masses et de transformer le problème de comparaison de masses entre spectres en une recherche de listes

d'objets (c'est-à-dire une liste de masses) partagés par un ou plusieurs ensembles (c'est-à-dire les spectres) d'une collection. Identifier des listes d'objets communs dans une collection d'ensemble est un problème auquel s'intéresse l'algorithme A-Priori mis au point en 1994 [1]. Malheureusement, pour les collections d'ensemble de grandes tailles, ni cet algorithme ni ses variantes ultérieures n'apportent de réponse qui permette de fournir à la fois une réponse exhaustive, dans un temps raisonnable et sur une station de travail courante.

Si nous parvenons à calculer de manière efficace le nombre de masses partagées entre un spectre expérimental s_i et un spectre théorique s_j, nous pouvons alors exploiter pour chaque paire (s_i, s_j) la différence entre la masse calculée de s_j et la masse de s_i observée par le spectromètre de masse. Nous nommons par la suite $\Delta(s_i, s_j)$ cette différence de masse entre le spectre théorique s_j et le spectre expérimental s_i. A partir d'une liste de paires de spectres pour lesquelles nous avons calculé la valeur de similarité, nous souhaitons utiliser cette différence de masse pour améliorer le score et sélectionner le meilleur candidat parmi les multiples identifications possibles.

4.4.2 Calcul de la similarité entre spectres

Dans la suite de ce document, le score utilisé pour évaluer la ressemblance entre un spectre expérimental s_i et un spectre théorique s_j sera noté $S(s_i, s_j)$. Ce score correspond très exactement au nombre de masses partagées entre les deux spectres (ni l'intensité ni aucun autre élément n'est pris en compte une fois le retraitement du spectre expérimental effectué). Plus la valeur de $S(s_i, s_j)$ est élevée, plus la probabilité que le spectre expérimental s_i identifie le peptide à partir duquel le spectre théorique s_j a été construit est forte. La méthode proposée dans la suite de ce document permet de calculer $S(s_i, s_j)$ pour n'importe quelle paire de spectre composée d'un spectre expérimental et d'un spectre théorique sans appliquer de filtre de sélection des candidats au préalable. Nous avons conscience que cette fonction d'évaluation peut être qualifiée de simpliste, mais elle fait sens pour plusieurs raisons : elle facilite la compréhension et l'analyse des résultats, elle est une des fonctions d'évaluation les plus rapides à calculer, et enfin elle est le socle de presque toute fonction d'évaluation plus complexe. Nous émettons donc l'hypothèse que cette fonction d'évaluation est suffisante pour obtenir des résultats convaincants (même si elle peut éventuellement par la suite être complexifiée).

Nous cherchons donc une manière de calculer, pour toute paire (s_i, s_j) la valeur $S(s_i, s_j)$ dans un temps limité (de l'ordre de l'heure) sur un ordinateur standard. A notre connaissance, il n'existe aujourd'hui pas de technique qui compare un très grand nombre de spectres deux à deux en un temps acceptable (du moins pas avec une capacité de calcul utilisable de façon routinière). Les méthodes existantes qui s'approchent le plus de notre problème et qui réalisent la comparaison de spectres expérimentaux avec des spectres théoriques sont les méthodes Open Modification Search, développées à l'origine dans l'idée de résoudre les problèmes de sensibilité imputés au filtre de masse des approches par recherche restreinte.

Malheureusement, si les approches OMS actuelles sont des méthodes hybrides et peinent à atteindre des temps de calcul raisonnable, c'est principalement parce que la comparaison des paires de spectres dans un ensemble de grande volumétrie soulève avant tout des problèmes de temps d'exécution et d'espace mémoire. De fait, les collections de spectres expérimentaux issues de spectrométrie de masse contiennent aujourd'hui plusieurs dizaines de milliers de spectres, tandis qu'une banque de protéine telle celle de Homo Sapiens conduit à la production de plusieurs centaines de milliers de spectres théoriques. Au delà de la considération du simple temps de calcul, le grand nombre de paires de spectres soulève également la question de la manipulation en mémoire de ces paires. Le risque est en effet important de devoir faire de nombreuses opérations de stockage sur disque pour conserver et manipuler les distances calculées, un processus bien plus lent que le simple accès mémoire des résultats.

Bien que les approches OMS développées jusqu'à présent soient un premier pas, nous sommes faisons face ici à un problème d'accessibilité de la technique et de passage à l'échelle. Beaucoup de petits laboratoires de spectrométrie de masse ne disposent en effet pas de ressources informatiques importantes à allouer à l'identification des résultats de manipulation. Nous souhaitons par conséquent proposer une méthode permettant de résoudre ce problème sur une station de travail classique.

Nous proposons dans la partie suivante de ce manuscrit une description détaillée de la solution que nous avons développé dans un premier chapitre, suivie d'une analyse des performances de cette solution.

SpecOMS

5

Description détaillée de SpecOMS

SpecOMS est un pipeline de trois algorithmes (SpecGrowth, SpecXtract et SpecFit). Ces algorithmes sont organisés autour d'une structure de données, SpecTrees, inspirée des Factorized Prefix Trees [61]. Une étape préliminaire de traitement des données est nécessaire à son exécution : le Bucket Clustering. A l'issue de cette étape de préparation des données, l'algorithme SpecGrowth organise les données de manière à ce que la structure de données SpecTrees contienne sous forme compacte le nombre de masses en commun entre n'importe quels spectres contenus dans la structure. Ces données peuvent alors être extraites par SpecXtract, puis analysées par SpecFit, pour fournir une analyse des spectres expérimentaux produits lors de l'expérience de spectrométrie de masse en tandem. Ce chapitre décrit SpecTrees, la structure de données sous-jacente à SpecOMS, ainsi que les différents composants de SpecOMS qui l'utilisent. On se réfèrera à la Figure 5.1 pour une représentation détaillée. Chacune des étapes décrites ci-après est expliquée en détail dans la suite de ce chapitre. En partant d'une collection de spectres $s \in S$ (a), l'algorithme de Bucket Clustering (b) organise ces données selon les masses observées pour chacun des spectres (c) dans des objets appelés "buckets". L'algorithme SpecGrowth (d) utilise ensuite ces buckets pour construire la structure de données SpecTrees (e) stockée en mémoire. Lors de ce processus de construction, des compteurs sont créés et actualisés par incréments pour calculer partiellement le nombre de masses en commun entre certains groupes de spectres. A partir de la structure de données SpecTrees, l'algorithme SpecXtract (f) extrait des couples de spectres (s_i, s_j) et calcule la similarité $S(s_i, s_j)$ associée. Cette étape fournit, pour chaque spectre s, une liste de spectres s_j (g) qui partagent un nombre suffisant de masses en commun. Enfin, SpecFit (h) utilise la paire (s_i, s_j), la similarité $S(s_i, s_j)$ et la différence de masse $\Delta(s_i, s_j)$ pour améliorer si possible l'alignement des spectres, et enfin proposer pour chaque spectre s_i la paire (s_i, s_j) (i) à retourner en tant que probable identification de s_i.

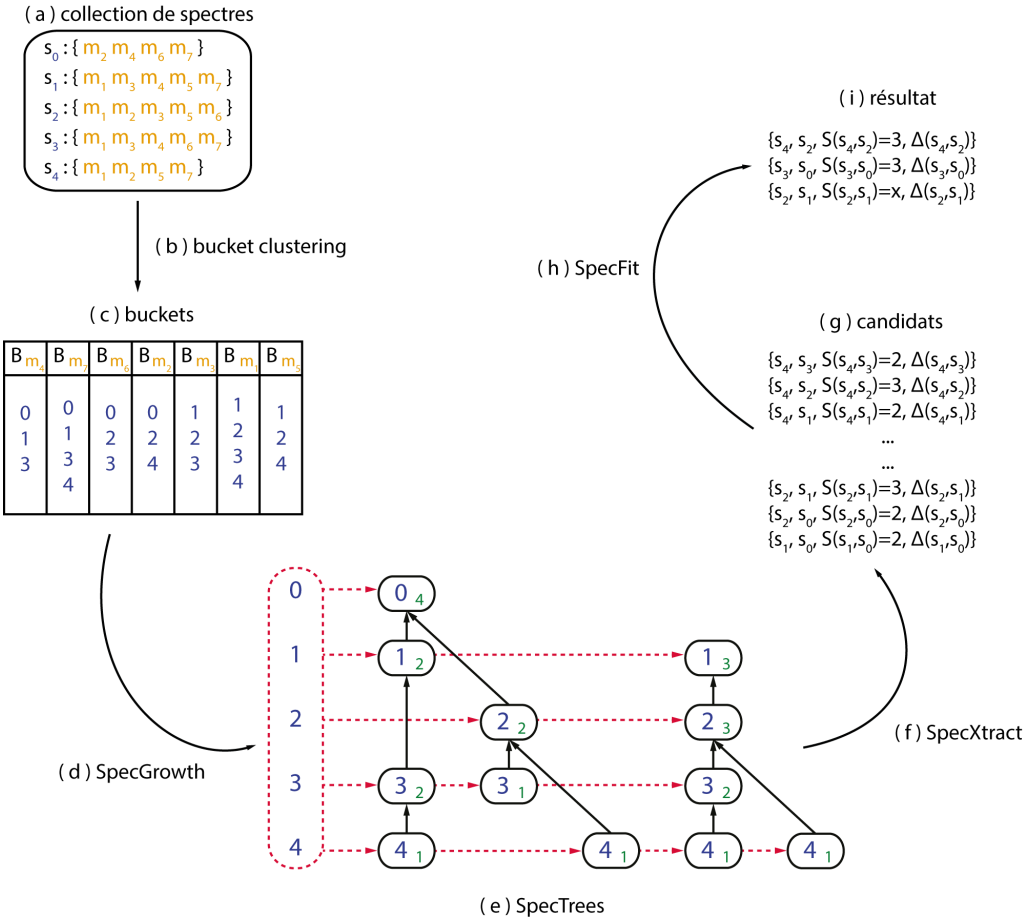


FIGURE 5.1 : Vue d'ensemble des composants de SpecOMS et des différentes étapes de traitement des données.

Dans ce manuscrit, nous présentons dans la Section 5.1 la structure de données SpecTrees et son contenu. Nous décrivons ensuite dans la Section 5.2 la préparation des spectres expérimentaux et théoriques, dans la Section 5.3 l'algorithme de Bucket Clustering utilisé pour la mise en forme des données, dans la Section 5.4 l'algorithme SpecGrowth, qui construit la structure de données SpecTrees, et enfin dans les Sections 5.5 et 5.6 les algorithmes SpecXtract et SpecFit, qui permettent l'exploitation des informations contenues dans SpecTrees. Nous précisons ensuite les différences importantes entre l'aspect conceptuel et l'implémentation technique de la structure et des différents algorithmes, en vue d'obtenir un temps d'exécution satisfaisant.

5.1 La structure de données SpecTrees

SpecTrees est la structure de données au cœur de SpecOMS. Elle permet, étant donné un ensemble de spectres S composé aussi bien de spectres théoriques que de spectres expérimentaux, de stocker de manière compacte les données numériques permettant de calculer rapidement $S(s_i, s_j)$ pour toutes les paires (s_i, s_j) de S^2, avec $i \neq j$.

Les spectres stockés dans SpecTrees peuvent être indifféremment des spectres expérimentaux ou bien des spectres théoriques, à partir du moment où ces spectres peuvent être représentés par des ensembles de masses discrètes. Lors de l'insertion des spectres dans SpecTrees par l'algorithme SpecGrowth, des calculs intermédiaires sont effectués, dont les résultats sont stockés dans la structure. Ces résultats seront ensuite utilisés par SpecXtract pour calculer la similarité $S(s_i, s_j)$ pour toutes les paires (s_i, s_j) de S^2, avec $i \neq j$.

La structure de données SpecTrees est illustrée en Figure 5.2. SpecTrees est une forêt d'arbres inversés, c'est-à-dire un ensemble d'arbres dont les arêtes sont orientées des feuilles vers la racine. Chaque nœud v de l'ensemble des nœuds V de cette forêt contient deux informations sous forme d'entiers : un identifiant de spectre s_{id} et un compteur c.

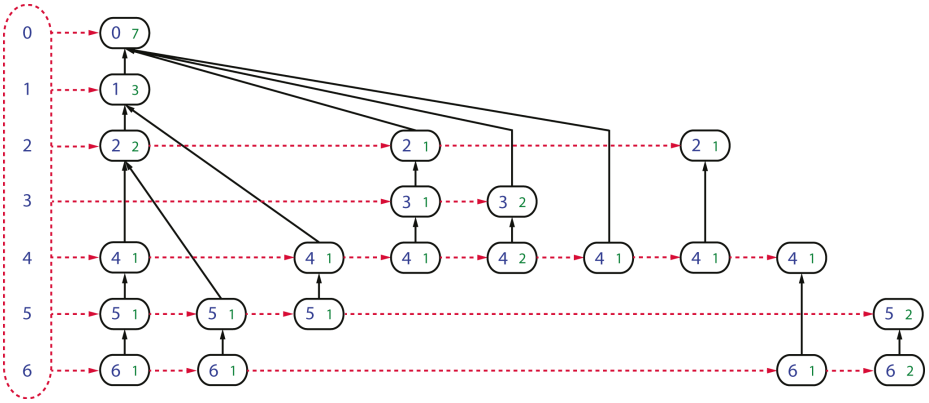


FIGURE 5.2 : Représentation conceptuelle de la structure de données SpecTrees. Les arbres sont représentés en noir. Les listes chaînées accessLists{ } utiles pour faciliter le parcours de la structure sont représentées en pointillés rouges. Les identifiants de spectres dans les nœuds et dans accessLists{ } figurent en bleu et les compteurs en vert.

Chaque identifiant de spectre s_{id} est unique (deux spectres différents ne peuvent avoir le même identifiant). Un nœud v de SpecTrees référence simplement le spectre dont il porte l'identifiant s_{id}. On définit $path(i)$ comme le chemin d'un nœud qui porte l'identifiant i à la racine de l'arbre dans lequel se trouve ce nœud. Un même identifiant s_{id} peut être répété dans plusieurs nœuds différents, mais SpecGrowth assure lors de la construction de SpecTrees que deux nœuds v_1 et v_2 de V ayant le même identifiant ne puissent être présents sur le même chemin. Deux nœuds v_1 et v_2 ayant le même identifiant peuvent néanmoins se situer dans le même arbre.

Le compteur c présent dans un nœud v quelconque représente une borne inférieure sur le nombre de masses partagées entre le spectre s_i, identifié par $s_{id}(v)$ et n'importe quel spectre s_j tel que $s_i < s_j$, identifié par $s_{id}(v')$ pour tout nœud $v' \in path(v)$.

L'extraction efficace des informations contenues dans SpecTrees nécessite de pouvoir réaliser un parcours rapide de tous les nœuds $v \in V$ ayant le même s_{id}, c'est-à-dire tous les nœuds associés au même spectre. Cette opération est réalisable grâce à un ensemble de listes chaînées nommé accessLists{ }.

L'ensemble de listes chaînées accessLists{ } est indexé par un identifiant de spectre. Une liste accessLists$_i${ } contient l'ensemble des nœuds de spectres dont l'identifiant est i. Parcourir la liste accessLists$_i${ } permet donc de parcourir tous les nœuds associés au spectre s_i.

5.2 Préparation des spectres

Avant insertion dans la structure de données SpecTrees, les spectres doivent subir des étapes de génération (pour les spectres théoriques) et de préparation.

Lors d'une expérience de spectrométrie de masse, les masses mesurées pour les ions fragments ne sont pas des valeurs discrètes exactes. Elles sont en réalité le résultat de la discrétisation (sous forme d'un nombre réel) d'un signal analogique mesuré avec une certaine précision. En conséquence, une masse m ne

correspond pas nécessairement exactement à la masse du fragment associé, lequel se trouve très vraisemblablement plutôt dans un intervalle $[m-\epsilon; m+\epsilon]$ autour de m, où ϵ est connu et correspond à l'incertitude de la mesure.

Autre difficulté en ce qui concerne la précision, les systèmes informatiques rencontrent d'importants problèmes de stabilité numérique lors d'opérations répétitives ou de comparaison des nombres flottants (nombres à virgule). Des opérations d'additions successives (nécessaires pour construire les modèles théoriques) risquent d'augmenter fortement l'incertitude sur la valeur des masses des fragments. De même, pour vérifier que deux masses m_1 et m_2 sont équivalentes, il va être nécessaire en travaillant avec des nombres flottants de vérifier que $m_1 \in [m_2-\epsilon; m_2+\epsilon]$. Il n'est en revanche pas nécessaire de vérifier également que $m_2 \in [m_1-\epsilon; m_1+\epsilon]$ car la valeur de ϵ utilisée est fixée et ne dépend pas des masses. Ce calcul double le nombre de comparaisons à effectuer, comparaisons déjà très coûteuses en temps pour nos systèmes informatiques actuels.

Pour contourner ces difficultés, des étapes supplémentaires de discrétisation des masses sont effectuées. Celles-ci diffèrent légèrement pour les spectres expérimentaux et les spectres théoriques.

5.2.1 Préparation des spectres expérimentaux

Les spectres expérimentaux sont fournis à SpecOMS sous la forme d'un fichier au format mgf ou mzxml contenant a minima pour chaque spectre expérimental : sa masse totale, sa charge et la liste de ses fragments. La Figure 5.3 présente un exemple de fichier mgf contenant un spectre expérimental.

```
1  BEGIN IONS
2  TITLE=exemple_spectre_expérimental.raw
3  SCANS=9691
4  RTINSECONDS=3078.71748
5  CHARGE=2+
6  PEPMASS=376.2445
7  110.0716 364630.1
8  147.1128 212809.6
9  129.1023 185264.4
10 173.1284 156217.3
11 130.0864 149129.3
12 581.3766 143413.4
13 322.1872 136835.4
14 421.2595 35939.2
15 493.2964 33067.2
16 164.1184 31899.8
17 445.3273 30100.9
18 626.372 29879.5
19 505.2534 28625.4
20 375.2858 28289.9
21 442.2806 26047.2
22 136.0758 25675.4
23 323.1895 25584.8
24 376.2795 24866.5
25 488.2099 24824.5
26 443.2804 24813.2
27 605.3794 24112.2
28 376.186 23537.2
29 END IONS
```

FIGURE 5.3 : Exemple de spectre expérimental au format MGF. Les lignes 1 et 29 comportent les délimiteurs d'un spectre au format MGF. Les lignes 2 à 6 comportent les informations traditionnelles pour un spectre de masse expérimental. Les lignes 7 à 28 comportent des pics sous la forme d'une paire de nombres flottants, le premier étant la masse exprimée en Dalton et le second l'intensité exprimée dans une unité arbitraire.

La première étape de préparation des spectres expérimentaux est de sélectionner pour chaque spectre expérimental les k masses des fragments les plus intenses, où k est un paramètre fixé par l'utilisateur. Pour chaque spectre expérimental, les fragments sont triés par ordre d'intensité décroissante puis les k premiers

sont sélectionnés et le reste des fragments est ignoré lors de la suite du traitement.

Pour éviter les difficultés relatives à la manipulation des nombres flottants, les masses restantes des fragments des spectres sont converties en nombres entiers. Pour ce faire, les masses des fragments sont multipliées par 10^{α}, où α est un paramètre fourni par l'utilisateur en fonction de la précision connue de son spectromètre de masse, puis sont arrondies à l'entier le plus proche. La valeur entière obtenue est celle de la masse du fragment mise à l'échelle. Par exemple, en partant de la valeur de masse 764.4914 Da et avec $\alpha = 2$, la valeur obtenue après ce processus est $76\,449$.

Pour décrire l'incertitude de mesure de l'appareil utilisé lors de l'expérimentation de spectrométrie de masse en tandem, on utilise plusieurs ensembles de masses supplémentaires. Ces masses sont ajoutées aux spectres expérimentaux et sont calculées à partir de la valeur discrétisée obtenue à l'étape précédente. Un second paramètre β également fourni par l'utilisateur permet de créer pour chaque masse m de chaque spectre un ensemble de 2β nouvelles masses $\{m-\beta, ..., m-1, m+1, ..., m+\beta\}$. Ces nouvelles masses et la masse m permettent de décrire une plage de valeurs qui seront comparées avec les masses non dupliquées des spectres théoriques. Une masse m_1 d'un spectre théorique est identique à la masse m_2 d'un spectre expérimental si elle correspond à l'un des entiers de l'ensemble $\{m_2 - \beta, ..., m_2 + \beta\}$. La combinaison des paramètres α et β permet ainsi de décrire les valeurs discrètes de l'intervalle de valeurs $[m - \epsilon; m + \epsilon]$ que peuvent prendre les masses des spectres expérimentaux dans SpecOMS. En repartant par exemple de la valeur $76\,449$ calculée dans le paragraphe précédent et avec $\beta = 2$, l'ensemble de masse créé est $\{76\,447, 76\,448, 76\,449, 76\,450, 76\,451\}$.

Ces paramètres α et β sont idéalement définis à partir de la précision du spectromètre de masse exprimée en Dalton pour la mesure des fragments. Pour une précision exprimée sous forme d'un nombre décimal, α correspond au nombre de décimales après la virgule, et β correspond à la partie fractionnelle. Par exemple avec 0.03 Dalton d'incertitude sur la mesure des fragments, α vaut 2 (deux chiffres après la virgule) et β vaut 3. Pour 0.025, α vaudrait 3 et β vaudrait 25. Cependant, comme nous le verrons plus loin, un tel paramétrage qui multiplie les masses n'est pas souhaitable.

5.2.2 Construction des spectres théoriques

La construction des spectres théoriques est réalisée selon la méthode décrite en Section 3.2.3, page 39.

Dans un premier temps, l'action d'une enzyme de digestion (le plus souvent la trypsine) est répliquée sur chacune des protéines de la banque sélectionnée et aboutit à un ensemble de peptides P. Chaque peptide $p \in P$ est alors utilisé pour construire un modèle théorique si la longueur de p est supérieure ou égale à une longueur minimale p_{min} et inférieure ou égale à une longueur maximale p_{max}. On estime en effet que des peptides trop courts comportent un risque très important d'identification due au hasard. En contrepartie, des peptides trop longs peuvent entraîner un biais lors de l'identification. En effet, ils comportent un plus grand nombre d'acides aminés, donc un plus grand nombre de fragments et le risque que des masses soient partagées par hasard est plus élevé. Les valeurs des paramètres p_{min} et p_{max} sont choisies par l'utilisateur, de même que l'enzyme de digestion. Préalablement à la digestion des protéines, l'utilisateur peut choisir de construire une banque dite *decoy* de ces protéines pour l'évaluation de la FDR (c.f. Section 3.2.6, page 42). Si l'utilisateur fait ce choix, alors les protéines de la banque decoy sont simplement fusionnées avec celle de la banque de départ, et un indicateur leur est apposé pour permettre de les distinguer dans les résultats finaux.

Une fois la digestion enzymatique de la banque effectuée, les listes de fragments théoriques correspondant à chaque peptide conservé sont construites en utilisant les masses isotopiques des acides aminés. Dans le cas des spectres théoriques, les masses isotopiques des acides aminés utilisées dans SpecOMS comprennent 5 décimales. Pour éviter des problèmes de stabilité numérique liée à l'addition de nombres

flottants, ces masses isotopiques sont d'abord multipliées par 10^5 de manière à les convertir en entiers. Ces valeurs sont utilisées pour calculer les masses des fragments, après quoi les masses calculées sont divisées par 10^5 pour finalement subir le même processus de mise à l'échelle que les masses des fragments des spectres expérimentaux (multiplication par le paramètre α et arrondi à l'entier le plus proche). Dans le cas des spectres théoriques, il n'est pas nécessaire d'effectuer de duplication des masses en utilisant le paramètre β. En effet, dans le cas de la présence d'une masse partagée, l'une des masses m du spectre théorique correspondra à l'une des masses de l'ensemble de masses $\{m-\beta, ..., m+\beta\}$ du spectre expérimental.

Les masses des fragments ainsi obtenues sont des nombres entiers calculés avec un minimum de perte de précision et peuvent être utilisées directement par l'algorithme de Bucket Clustering de SpecOMS.

5.3 Bucket Clustering

Le Bucket Clustering est une étape de répartition des spectres en différents ensembles appelés *buckets*. Cette répartition s'effectue en fonction des masses contenues dans les spectres.

Un bucket B est un ensemble d'identifiants de spectres indexé par une masse m. Soit un spectre s et $M_s = \{m_1, ..., m_p\}$ l'ensemble des masses de ce spectre, soit un bucket B_m : le spectre s appartient à B_m si et seulement si il existe $m_j \in M_s$ telle que $m_j = m$.

Pour réaliser le Bucket Clustering, les spectres à ajouter dans la structure SpecTrees sont parcourus par ordre d'identifiant de spectre croissant. Les masses de chaque spectre sont lues successivement : si il existe déjà un bucket pour une masse donnée (c'est-à-dire indexé par cette masse), l'identifiant du spectre actuellement traité est ajouté à ce bucket. Dans le cas contraire, un nouveau bucket indexé par cette masse est créé et l'identifiant du spectre traité y est ajouté. La Figure 5.4 illustre le début de la procédure de Bucket Clustering avec les spectres s_0, s_1, s_2, s_3. La Figure 5.5 illustre la poursuite du processus avec les spectres s_4, s_5, s_6.

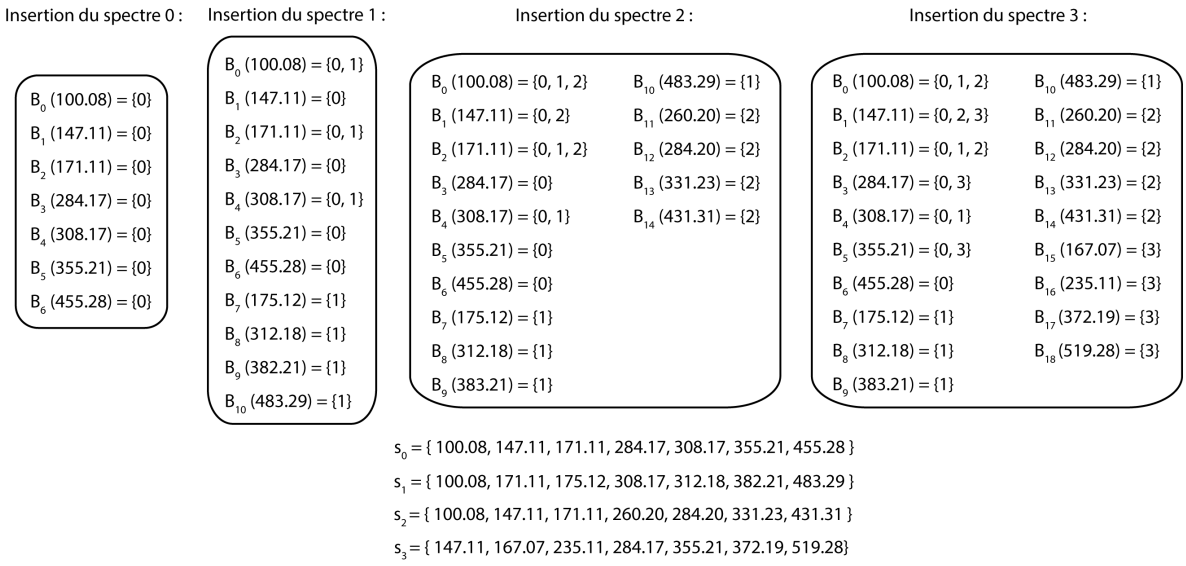


FIGURE 5.4 : Exemple de processus de Bucket Clustering (partie 1). Les spectres s_0, s_1, s_2, s_3 sont successivement insérés dans les buckets. Par exemple, lors de l'insertion du spectre s_2, la masse 100.08 est ajoutée au bucket B_0 qui contient déjà les identifiants des spectres s_0 et s_1 (car s_0 et s_1 possèdent aussi la masse 100.08). De la même manière la masse 147.11 est ajoutée à B_1 et la masse 171.11 à B_2. Les masses restantes de s_2 conduisent à la création des nouveaux buckets B_{11}, B_{12}, B_{13} et B_{14}.

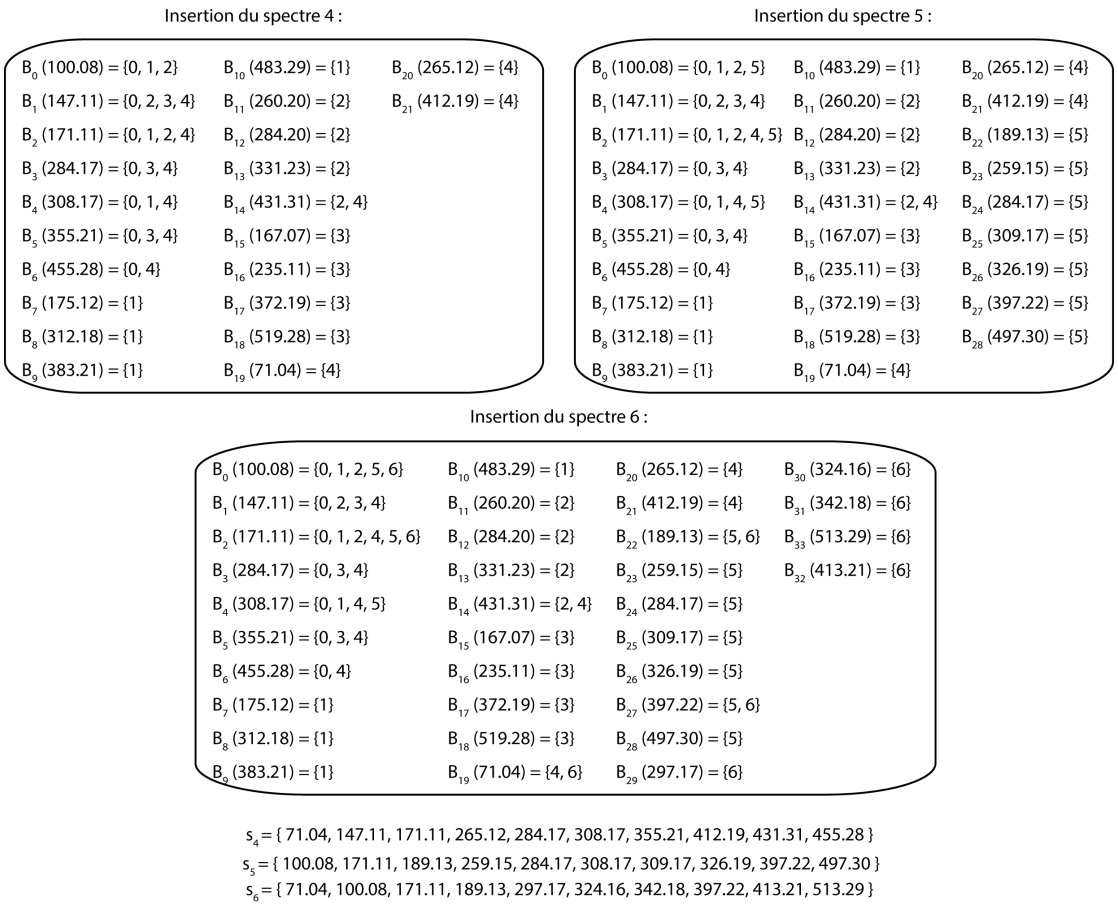


FIGURE 5.5 : Exemple de processus de Bucket clustering (partie 2). Les spectres s_4, s_5, s_6 sont successivement insérés dans les buckets.

A l'issue de ce processus, un spectre s qui possède k masses distinctes est présent dans k buckets différents. Ainsi par exemple, à une précision de 0.02 Dalton et en ayant retenu les 50 pics les plus intenses d'un spectre donné, si l'on suppose que toutes les masses obtenues sont distinctes, on obtient 250 copies de l'identifiant de ce spectre réparties dans 250 buckets différents. Tous les spectres présents dans un même bucket B_m partagent au moins la masse m en commun. Il existe cependant des buckets qui ne contiennent qu'un seul identifiant de spectre, ce qui signifie que la masse associée n'est présente que dans un seul spectre. Le but étant de calculer des masses partagées entre des paires de spectres, ces buckets peuvent être supprimés, comme illustré en Figure 5.6.

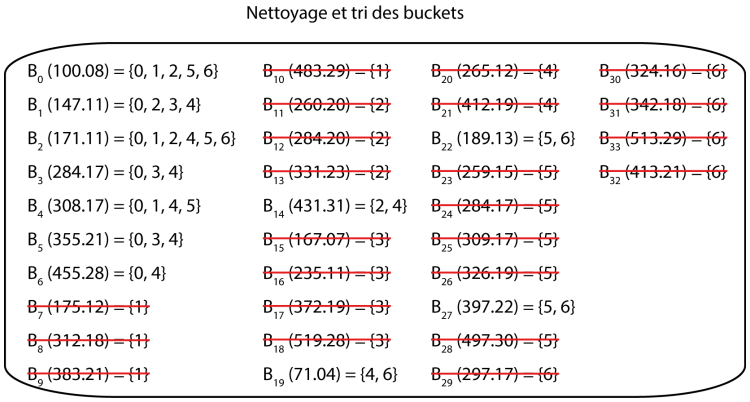


FIGURE 5.6 : Exemple de processus de Bucket clustering (partie 3). Les buckets contenant un unique identifiant de spectre sont supprimés, par exemple le bucket B_7 qui contient l'unique identifiant de spectre s_1, puis le contenu de chaque bucket restant est trié par ordre d'identifiants de spectres croissants.

Après ajout des identifiants de spectres, le contenu de chaque bucket est trié par identifiant de spectre croissant. L'ensemble des buckets est ensuite trié par ordre lexicographique en fonction du contenu des buckets. Après ces étapes de tri, illustrées en Figure 5.7, les données du Bucket Clustering peuvent être utilisées par SpecGrowth pour la construction de SpecTrees.

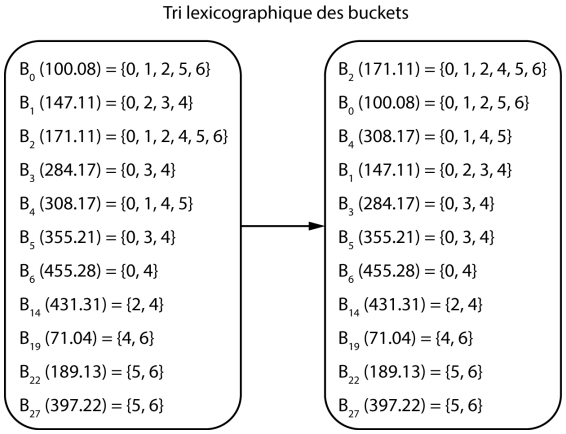


FIGURE 5.7 : Exemple de processus de Bucket clustering (partie 4). Enfin, l'ensemble des buckets est trié par ordre lexicographique.

5.4 SpecGrowth

SpecGrowth est l'algorithme qui construit la structure de données SpecTrees à partir des données mises en forme par l'algorithme de Bucket Clustering, c'est-à-dire à partir de l'ensemble des buckets triés par ordre lexicographique croissant.

La construction de SpecTrees est effectuée par insertions successives des identifiants de spectres contenus dans les buckets. En fonction des spectres précédemment insérés, de nouveaux nœuds sont ajoutés à SpecTrees lors de l'insertion des identifiants de spectres, ou bien des compteurs sont incrémentés dans des nœuds existants. Lors du processus de construction, les liens des nœuds à leurs parents et les listes chaînées de accessLists{ } sont également maintenus à jour.

5.4.1 Construction de SpecTrees

Le processus de construction de SpecTrees se fait en respectant l'ordre des buckets lexicographiquement triés lors de l'opération de Bucket Clustering : les identifiants de spectres contenus dans les buckets sont insérés un à un dans SpecTrees jusqu'à ce que tous les buckets aient été traités.

L'ajout des informations contenues dans le bucket initial se fait de la manière suivante : SpecGrowth part du premier identifiant de spectre i du bucket. Pour cet identifiant i, un premier nœud v est créé dans SpecTrees, et on assigne au nœud v l'identifiant lu dans le bucket, c'est-à-dire $s_{id}(v) = i$. Ce premier nœud v forme la racine du premier arbre de SpecTrees. Le compteur de ce nœud est initialisé à 1 et le nœud v est ajouté à la liste l_i de accessLists{ }. SpecGrowth insère ensuite les identifiants de ce premier bucket dans l'ordre des identifiants qu'il contient. Chaque insertion est très similaire à celle du premier nœud de SpecTrees à ceci près que lors de la création d'un nouveau nœud v', celui-ci est relié au nœud précédent v dans l'arbre par une relation père-fils allant du nœud fils v' vers le nœud père v. La Figure 5.8 illustre un exemple d'insertion dans SpecTrees d'un premier bucket à partir de l'exemple de Bucket Clustering fourni précédemment. SpecGrowth peut ensuite procéder à l'insertion des buckets suivants pour lesquels la procédure diffère légèrement de l'insertion des identifiants de spectres du premier bucket.

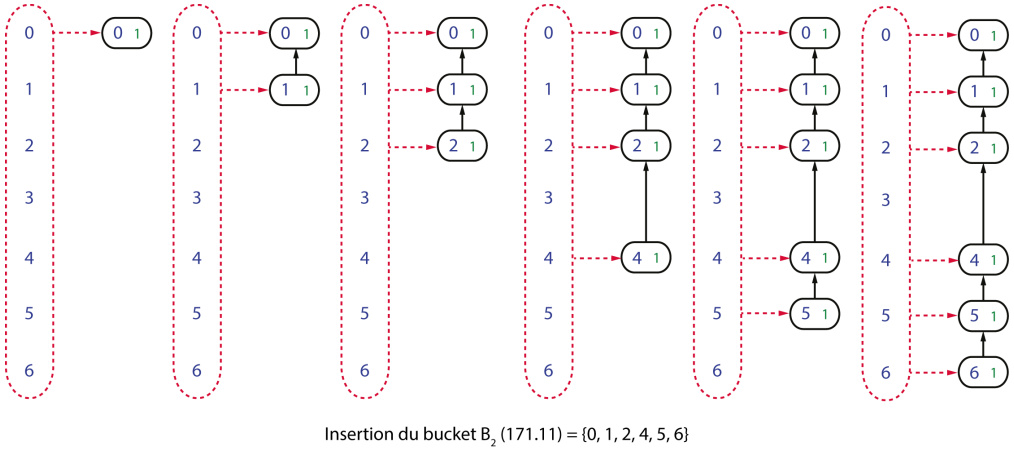


FIGURE 5.8 : Exemple de construction de SpecTrees par l'algorithme SpecGrowth (partie 1). Insertion du contenu du premier bucket : les identifiants de spectre sont insérés dans l'ordre et donnent lieu à la création de nouveaux nœuds dont les compteurs sont initialisés à 1. Chaque nouveau nœud est relié au précédent qui devient son père, et est également relié à la liste chaînée correspondante dans accessLists{ }.

Pour chaque bucket après le premier, SpecGrowth procède à l'insertion des identifiants contenus dans chaque bucket en partant de la racine de l'arbre. Si SpecTrees contient déjà plusieurs arbres, SpecGrowth travaille toujours sur l'arbre dont la racine est la plus récemment insérée dans SpecTrees. Soit r la racine de cet arbre et i le premier identifiant dans le bucket à insérer. Si $s_{id}(r) = i$ (c'est-à-dire que la racine porte le même identifiant de spectre que celui à ajouter), aucun nouveau nœud n'est inséré et le compteur $c(r)$ est incrémenté de 1. Sinon, un nouveau nœud v est créé avec $s_{id}(v) = i$ et un compteur $c(v) = 1$. Ce nouveau nœud v est ajouté à la liste l_i de accessLists{ } et forme la racine r' d'un nouvel arbre dans SpecTrees. SpecGrowth travaille désormais sur cet arbre à la place du précédent. Dans tous les cas, SpecGrowth passe à l'insertion de l'identifiant suivant du bucket dans SpecTrees jusqu'à ce que la totalité du bucket soit traité.

Lors de l'insertion de l'identifiant j qui suit dans le bucket, deux cas se présentent : soit il existe un nœud v fils du nœud r (ou r' si un nouvel arbre a été créé) tel que $s_{id}(v) = j$, soit il n'en existe pas. Si le nœud v existe, alors son compteur $c(v)$ est simplement incrémenté de 1. Sinon, un nouveau nœud v' est créé et initialisé avec $s_{id}(v') = j$ et $c(v') = 1$. Le nœud v' est relié au nœud v qui devient son père. Le nœud v' est

également ajouté à la liste l_j où il prend la place de dernier élément. SpecGrowth itère ce processus jusqu'à traitement de la totalité des identifiants du bucket. L'insertion des identifiants d'un bucket conduisant à la formation d'une fourche est illustré pas à pas en Figure 5.9. La Figure 5.10 illustre l'insertion d'un bucket conduisant à la création d'une seconde fourche ainsi que l'insertion d'un bucket conduisant à la formation d'un nouvel arbre.

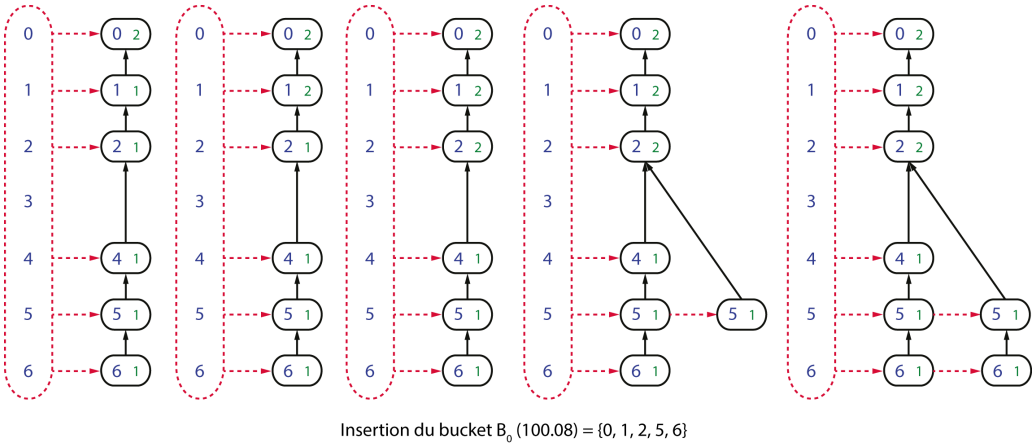


FIGURE 5.9 : Exemple de construction de SpecTrees par l'algorithme SpecGrowth (partie 2). Insertion du contenu du second bucket : en partant de la racine de l'arbre courant, tant que les identifiants des spectres correspondent à des nœuds existants, les compteurs de ces nœuds sont incrémentés de 1. Une fois qu'il n'existe plus de nœud correspondant à l'identifiant à ajouter, celui-ci et tous les identifiants suivants sont ajoutés le long d'une nouvelle branche.

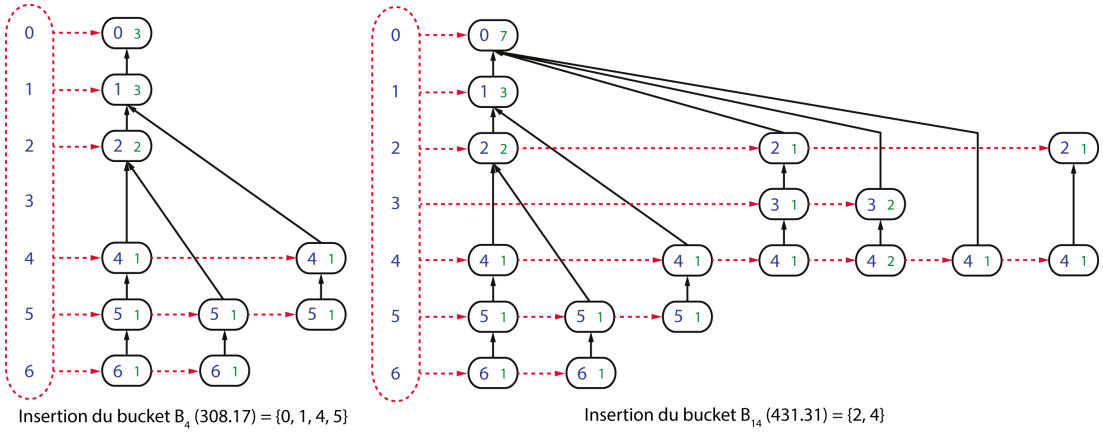


FIGURE 5.10 : Exemple de construction de SpecTrees par l'algorithme SpecGrowth (partie 3). La figure de gauche illustre l'insertion du contenu du troisième bucket, qui conduit à la création d'une nouvelle branche dans l'arbre existant, tandis que la figure de droite illustre l'insertion du contenu d'un bucket conduisant à la création d'un nouvel arbre dans SpecTrees.

On notera que dès que l'insertion d'un identifiant conduit à la création d'un nouveau nœud, l'insertion du reste des identifiants contenus dans le bucket est équivalente à l'insertion des identifiants du premier bucket : il n'y a plus de compteur à incrémenter, mais uniquement de nouveaux nœuds fils à créer. Une fois que les identifiants de spectres contenus dans le dernier bucket ont tous été ajoutés à SpecTrees, la structure de données est complète.

Pour une paire de spectres (s_i, s_j) donnée contenue dans la structure, SpecTrees contient l'information nécessaire pour calculer rapidement $S(s_i, s_j)$. Le véritable atout de cette structure de données est néanmoins

de pouvoir fournir rapidement cette information pour *toutes* les paires de spectres contenues dans SpecTrees, et ce avec une empreinte mémoire optimisée. C'est cette tâche que réalise par l'agorithme SpecXtract.

5.5 SpecXtract

SpecXtract est l'algorithme qui, pour chaque paire de spectres (s_i, s_j), extrait l'information contenue dans SpecTrees et calcule $S(s_i, s_j)$ lorsque $i > j$. SpecXtract ne peut calculer la similarité $S(s_i, s_j)$ si $i < j$, ce qui n'est pas un véritable problème en raison de la symétrie de la mesure de similarité (il suffit de calculer $S(s_j, s_i)$ qui lui est égale). Nous expliquons dans un premier temps comment extraire $S(s_i, s_j)$ pour une paire (s_i, s_j) donnée. Dans un second temps, nous généralisons la méthode et montrons comment calculer toutes les valeurs $S(s_i, s_j)$.

5.5.1 Extraction de similarité pour une paire de spectres

Soient deux spectres $s_i \in S$ et $s_j \in S$ donnés tels que $i > j$. La similarité $S(s_i, s_j)$ peut être calculée de la manière suivante dans un accumulateur entier noté $acc_{i,j}$ initialisé à 0.

SpecXtract commence par accéder au premier nœud v de la liste $l_i \in L$ dans accessLists{ } et mémorise la valeur $c(v)$. SpecXtract interroge le contenu des nœuds situés sur le chemin de v jusqu'à la racine de l'arbre. Si, lors de ce cheminement vers la racine, SpecXtract rencontre un nœud v' tel que $sid(v') = j$, alors SpecXtract ajoute à l'accumulateur la valeur mémorisée $c(v)$.

Après avoir remonté la branche à partir du nœud v, SpecXtract accède au nœud suivant w dans la liste l_i, mémorise la valeur $c(w)$ et répète le processus de remontée jusqu'à la racine. Ici encore, SpecXtract ajoute la valeur $c(w)$ à $acc_{i,j}$ si un nœud w' est rencontré lors de la remontée tel que $sid(w') = j$ (éventuellement le nœud w' peut être le même que le nœud v').

Une fois tous les nœuds de la liste l_i parcourus et la remontée à la racine effectuée pour chacun d'entre eux, l'accumulateur $acc_{i,j}$ contient la valeur $S(s_i, s_j)$. La Figure 5.11 illustre le processus d'extraction sur l'exemple utilisé dans ce document.

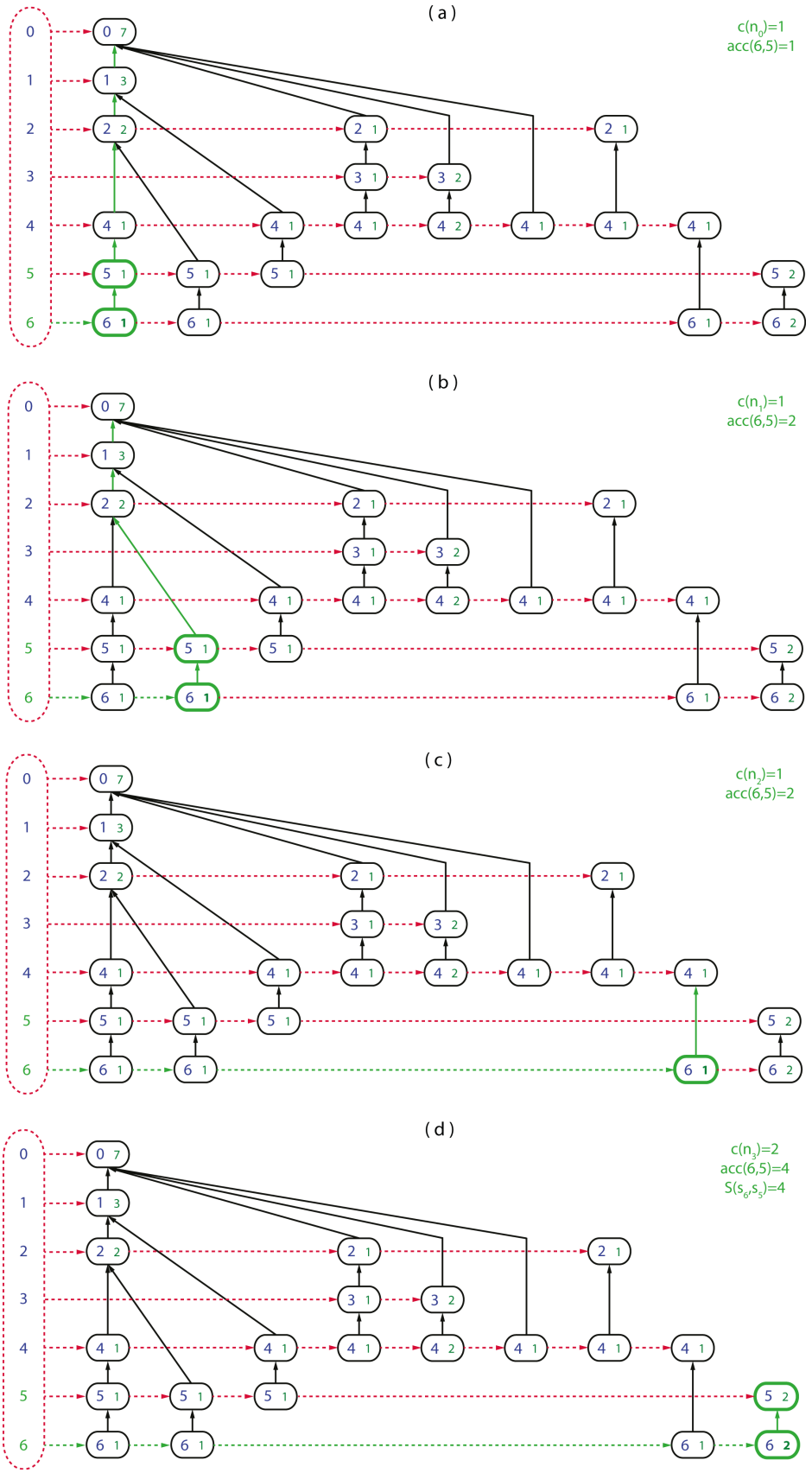


FIGURE 5.11 : Exemple de calcul de similarité $S(s_6, s_5)$ par l'algorithme SpecXtract pour la paire de spectres (s_6, s_5). Cette figure illustre le calcul de $S(s_6, s_5)$. Le parcours réalisé par SpecXtract est illustré en vert et les valeurs de travail temporaires sont indiquées en haut à droite de SpecTrees.

En pratique, SpecXtract ne calcule pas la similarité $S(i, j)$ pour une unique paire de spectres (s_i, s_j)

donnée mais pour *tous* les spectres $s_i \in S$ et $s_j \in S$ tels que $i > j$, et ce en minimisant le nombre de passes de lectures de SpecTrees. Nous présentons cette méthode dans la section suivante.

5.5.2 Généralisation à toutes les paires

Pour généraliser le calcul des valeurs de similarités, SpecXtract procède à plusieurs exécutions successives d'une procédure de lecture de SpecTrees : une pour chaque spectre expérimental dans SpecTrees. Chaque exécution de cette procédure de lecture débute par le premier nœud d'une liste chaînée l_i et est notée $read_i$. Lors de l'exécution de $read_i$, SpecXtract calcule les similarités $S(s_i, s_j)$ pour tous les spectres s_j tels que $i > j$. Pour le bon fonctionnement de cette procédure, SpecXtract n'utilise plus un unique accumulateur mais un ensemble d'accumulateurs indexés par les identifiants de spectres comme suit. Soit s_i un spectre donné et s_j tout spectre tel que $i > j$. On définit ACC_i un ensemble d'accumulateurs chacun associé à un spectre s_j et indexé par j, tous initialisés à 0.

Pour extraire $S(s_i, s_j)$ où s_i est fixé et où $0 \leq j \leq i-1$, la procédure $read_i$ réalise un parcours de SpecTrees de manière similaire à celui effectué pour une unique paire de spectres. La procédure $read_i$ parcourt successivement tous les nœuds de la liste chaînée l_i à partir du premier. Pour chaque nœud v rencontré lors du parcours de l_i, $read_i$ va mémoriser la valeur $c(v)$, puis remonter la branche dans laquelle se trouve ce nœud v jusqu'à atteindre la racine de l'arbre. Pour chaque nœud v' rencontré lors de cette remontée à la racine, $read_i$ incrémente de $c(v)$ l'accumulateur $acc_{(s_{id}(v),s_{id}(v'))}$ de ACC. Une fois tous les chemins partant de tous les nœuds de la liste l_i remontés jusqu'à la racine, les accumulateurs de l'ensemble ACC contiennent les valeurs voulues : on a $S(s_i, s_j) = acc_{(i,j)}$. La Figure 5.12 illustre ce processus sur un exemple.

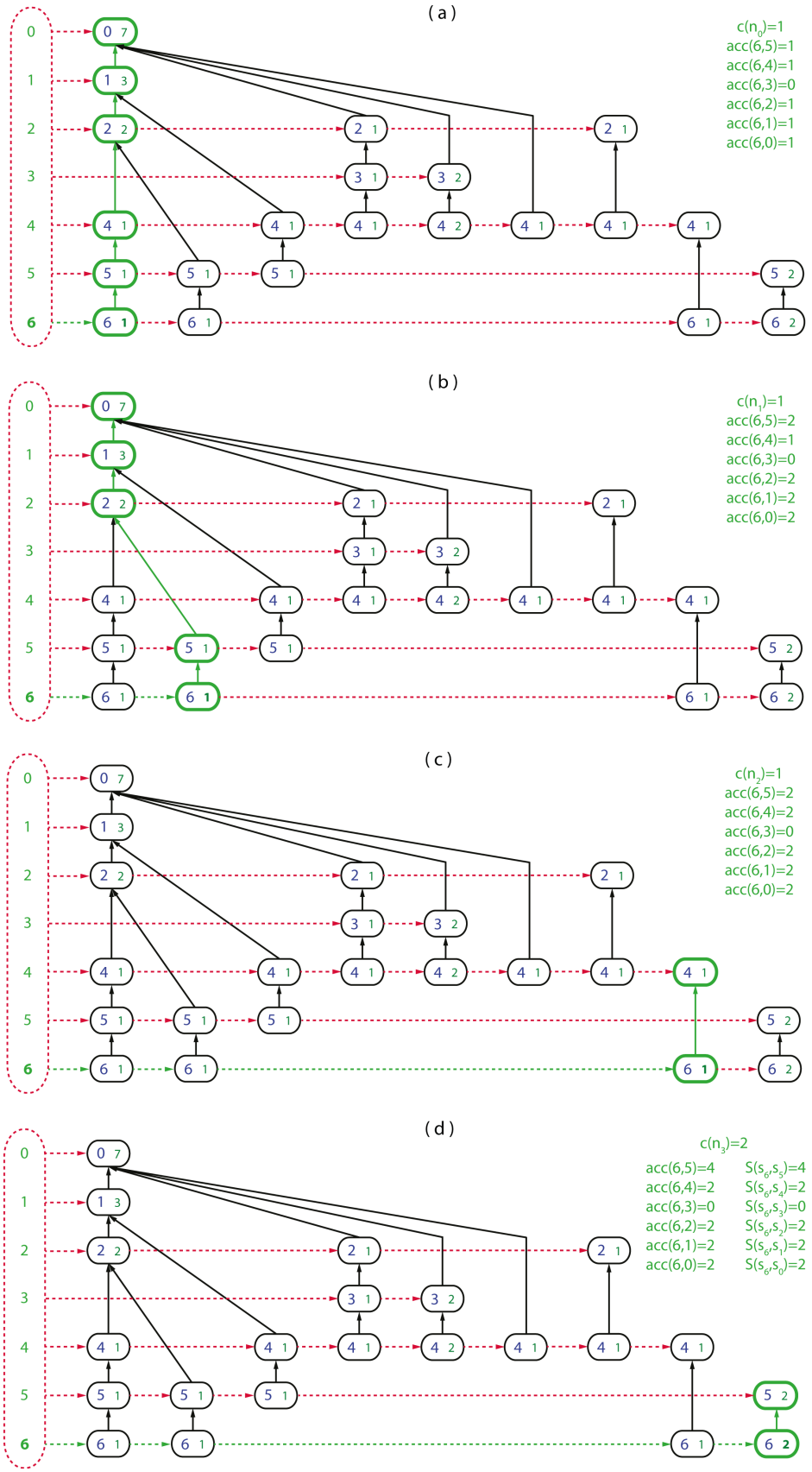


FIGURE 5.12 : Exemple de calcul de similarité par l'algorithme SpecXtract pour un spectre avec tous les autres. Cette figure illustre le calcul de $S(s_6, sj)$ pour tout s_j tel que $0 \leq j \leq 5$. Le parcours réalisé par SpecXtract est illustré en vert et les valeurs de travail temporaires sont indiquées en haut à droite de SpecTrees.

A l'issue de la procédure $read_i$, SpecXtract ne fait pas directement appel à la procédure $read_{i-1}$, afin de ne pas avoir à stocker en mémoire trop de résultats. A la place, SpecOMS fait immédiatement appel à l'algorithme suivant, SpecFit, pour analyser les similarités calculées par la procédure $read_i$. Tous les résultats calculés par $read_i$ ne sont pas utilisés : en effet, beaucoup des valeurs de similarités calculées sont faibles. Pour des raisons de temps de calcul et d'espace de stockage des résultats, on choisit d'ignorer les valeurs en dessous d'un certain seuil T, T étant un paramètre fixé par l'utilisateur. Les valeurs calculées par SpecXtract qui sont conservées sont communiquées à SpecFit sous la forme de 4-uplets qui correspondent aux PSMs, composés de deux spectres s_i et s_j, la similarité $S(s_i, s_j)$ et la différence de masse entre s_i et s_j notée $\Delta(s_i, s_j)$.

A l'issue de ce processus, SpecXtract a permis de calculer les valeurs $S(s_i, s_j)$ pour l'ensemble des paires (s_i, s_j) avec $i > j$. SpecXtract peut alors exécuter la procédure $read_j$ où s_j est le spectre de SpecTrees tel que $j = i - 1$.

5.6 SpecFit

L'algorithme SpecFit est exécuté sur les PSMs produits par SpecXtract lorsque le score $S(s_i, s_j)$ est supérieur au seuil T fixé par l'utilisateur. SpecFit est utilisé à la fois pour discriminer les nombreuses solutions d'identification proposées pour un même spectre s_i, et pour rechercher des spectres modifiés, pour lesquels le delta de masse associé au PSM n'est pas proche de zéro et doit donc être interprété. Pour un spectre s_i donné, l'algorithme SpecXtract, va retourner les similarités $S(s_i, s_j)$ pour tous les spectres s_j avec $i > j$ pour lesquelles $S(s_i, s_j) \geq T$. Il peut donc y avoir plusieurs PSMs candidats à l'identification de s_i, c'est-à-dire plusieurs paires de spectres qui contiennent le spectre s_i dans les PSMs. Pour chacun des PSMs proposés par SpecXtract, SpecFit va exécuter différentes analyses en fonction du delta de masse $\Delta(s_i, s_j)$ observé entre les deux spectres de ce PSM. Si ce delta est proche de 0 ($\Delta(s_i, s_j) \in [-\epsilon, +\epsilon]$ où ϵ est l'incertitude de mesure de l'appareil recalculée à partir des paramètres α et β fournis par l'utilisateur) et si $S(s_i, s_j) \geq S_{\Delta m=0}$, SpecFit laisse le PSM inchangé. Si $\Delta(s_i, s_j) < -\epsilon$, SpecFit va opérer une recherche de semi-tryptique (on émet l'hypothèse qu'une partie du peptide candidat est manquante dans le spectre en raison d'une digestion enzymatique partielle). Si $\Delta(s_i, s_j) > +\epsilon$, SpecFit va opérer une recherche de missed-cleavage (on émet l'hypothèse que le spectre expérimental est en réalité composé de deux peptides qui n'ont pas été convenablement séparés par l'enzyme de digestion). Dans tous les cas, si $\Delta \notin [-\epsilon; +\epsilon]$ ou $S(s_i, s_j) < S_{\Delta m=0}$, SpecFit réalise également une étape de détection des modifications pour tenter de proposer une meilleure identification. Pour un spectre donné, au fur et à mesure de chaque PSM obtenu (soit issue de SpecXtract, soit par la recherche effectuée par SpecFit), SpecFit effectue une sélection entre ce PSM et le précédent pour n'en conserver qu'un seul.

Lors de ce processus de sélection du meilleur PSM, plusieurs informations peuvent être utilisées : la valeur de similarité entre les deux spectres du PSM avant un possible alignement par SpecFit, ainsi que le delta de masse entre les deux spectres du PSM. En particulier, le delta de masse observé entre les deux spectres du PSM nous permet de favoriser les PSMs pour lesquels le delta de masse est nul. Nous jugeons en effet qu'en deçà d'un certain score, un PSM exhibant un delta de masse nul a plus de probabilité de correspondre à la bonne identification qu'un PSM exhibant une modification. Pour favoriser ce type de PSMs, nous définissons un seuil alternatif nommé $S_{\Delta m=0}$. Tout PSM reporté par SpecXtract pour lequel le delta de masse et nul et exhibant un score supérieur à $S_{\Delta m=0}$ sera considéré comme un meilleur candidat. Notons R un PSM reporté pour le spectre s_i actuellement sélectionné comme le PSM le plus probable. Les paragraphes suivants décrivent dans un premier temps le processus de sélection entre le PSM R et un nouveau PSM candidat R', à la fois dans le cas où R' est un PSM produit à partir de R par SpecFit, et dans le cas où R' est un autre PSM reporté par SpecXtract pour le spectre s_i. Dans un second temps, nous expliquons les différents processus de recherche effectués par SpecFit à partir de R qui peuvent conduire à

la production d'un PSM R'.

Sélection du PSM résultat

Lors de l'exécution de l'algorithme SpecFit, le PSM R est initialisé avec le premier PSM obtenu pour un spectre s_i tel que la similarité est supérieure au seuil T. Chaque PSM R' suivant pour ce même spectre s_i fourni par SpecXtract ou généré par SpecFit est alors comparé avec R pour savoir lequel conserver en respectant les mécanismes suivants.

Quatre cas différents peuvent se présenter :

- a) un PSM est tel que $\Delta(s_i, s_j) \in [-\epsilon, +\epsilon]$
- b) un PSM est un cas de semi-tryptique
- c) un PSM est un cas de missed-cleavage
- d) un PSM est tel que $\Delta(s_i, s_j) \notin [-\epsilon, +\epsilon]$ et n'est ni un cas de missed-cleavage, ni un cas de semi-tryptique

Les différents cas ci-dessous listent sous quelles conditions le PSM R' remplace ou non le PSM R.

- De R ou R', on conserve toujours préférentiellement un PSMs qui est dans le cas (a), sauf si l'autre PSM a un score supérieur à $S_{\Delta m=0}$.
- De R ou R', on conservera préférentiellement un PSM dans le cas (b) si l'autre correspond au cas (c) ou (d).
- De R ou R', on conservera préférentiellement un PSM dans le cas (c) si l'autre correspond au cas (d).

Dans chacune des situations décrites précédemment, si deux PSMs correspondent au même cas, on conserve celui qui a le score le plus élevé. dans le cas où $S'(s_i, s_j) = S(s_i, s_j)$ se présenterait, SpecFit choisit de conserver le PSM qui exhibe le delta de masse le plus proche de 0.

Les différents cas dans lesquels SpecFit génère de nouveaux PSMs avec pour objectif l'analyse de semi-tryptiques, de recherche de missed-cleavages et de détection de modification sont expliqués ci-dessous.

Analyse des semi-tryptiques

L'analyse des semi-tryptiques est uniquement réalisée dans le cas des PSMs pour lesquels $\Delta(s_i, s_j) < -\epsilon$, c'est-à-dire pour lesquels le spectre expérimental a une masse inférieure à celle du modèle théorique. En partant de l'extrémité gauche du peptide, les masses des acides aminés de la séquence peptidique sont soustraites à la masse du spectre théorique s_j pour simuler une séquence s'_j jusqu'à ce que $\Delta(s_i, s'_j) \geq -\epsilon$. Si le processus s'interrompt avec $\Delta(s_i, s'_j) > +\epsilon$, alors le spectre expérimental ne correspond pas à un peptide semi-tryptique basé sur cette séquence. Si en revanche $\Delta(s_i, s'_j) \in [-\epsilon; +\epsilon]$, les fragments du nouveau peptide modèle s'_j sont construits et on calcule $S(s_i, s'_j)$ pour produire un nouveau PSM R'. Le même processus est ensuite répété en partant de l'extrémité droite du peptide.

Recherche des missed-cleavages

La recherche de peptides avec missed-cleavages est uniquement réalisée dans le cas où $\Delta(s_i, s_j) > +\epsilon$, c'est-à-dire que le spectre théorique possède une masse moindre que celle du spectre expérimental. Pour un PSM constitué d'un spectre expérimental s_i et d'un spectre théorique s_j associé à un peptide p, SpecFit va aller consulter les peptides voisins de p dans toutes les protéines de la banque qui contiennent p. Notons q un de ces peptide voisins. Si $\Delta(s_i, s_j)$ est égal à la masse du peptide q, alors un nouveau peptide p', concaténation de p et de q, est construit. Un nouveau modèle de spectre théorique s'_j correspondant au peptide p' est construit et le nombre de masses partagées avec s_i est calculé pour produire un nouveau PSM R'. Ce processus est ensuite répété pour tous les peptides q voisins de p dans toutes les protéines qui contiennent le peptide p.

Détection des modifications

Dans le cas où $\Delta \notin [-\epsilon; +\epsilon]$, on peut estimer que le spectre expérimental porte vraisemblablement une ou plusieurs modifications chimiques. Si le spectre théorique s_j proposé est bien identifié par le spectre s_i, on connaît la différence de masse totale induite par la présence de ces modifications, qui est équivalente à $\Delta(s_i, s_j)$. Plusieurs modèles de spectres théoriques vont donc être construits en ajoutant successivement la masse $\Delta(s_i, s_j)$ à chacun des acides aminés de la séquence du peptide p à partir duquel s_j a été obtenu. Pour chacun des peptides modifiés p' ainsi générés, un nouveau modèle théorique s'_j est construit et le nombre de masses partagées entre s_i et s'_j est calculé. Si le peptide modifié p' nouvellement créé conduit à un spectre théorique s'_j tel que $S(s_i, s'_j) > S(s_i, s_j)$, on produit un nouveau PSM R' composée du spectre expérimental s_i et du nouveau spectre théorique s'_j. Pour un couple (s_i, s_j), plusieurs nouveaux PSMs peuvent donc être générés en fonction de l'emplacement auquel la modification est testée dans la séquence. Ces nouveaux PSMs sont comparés au fur et à mesure de leur génération avec le PSM R. Un exemple est fourni en Figure 5.13.

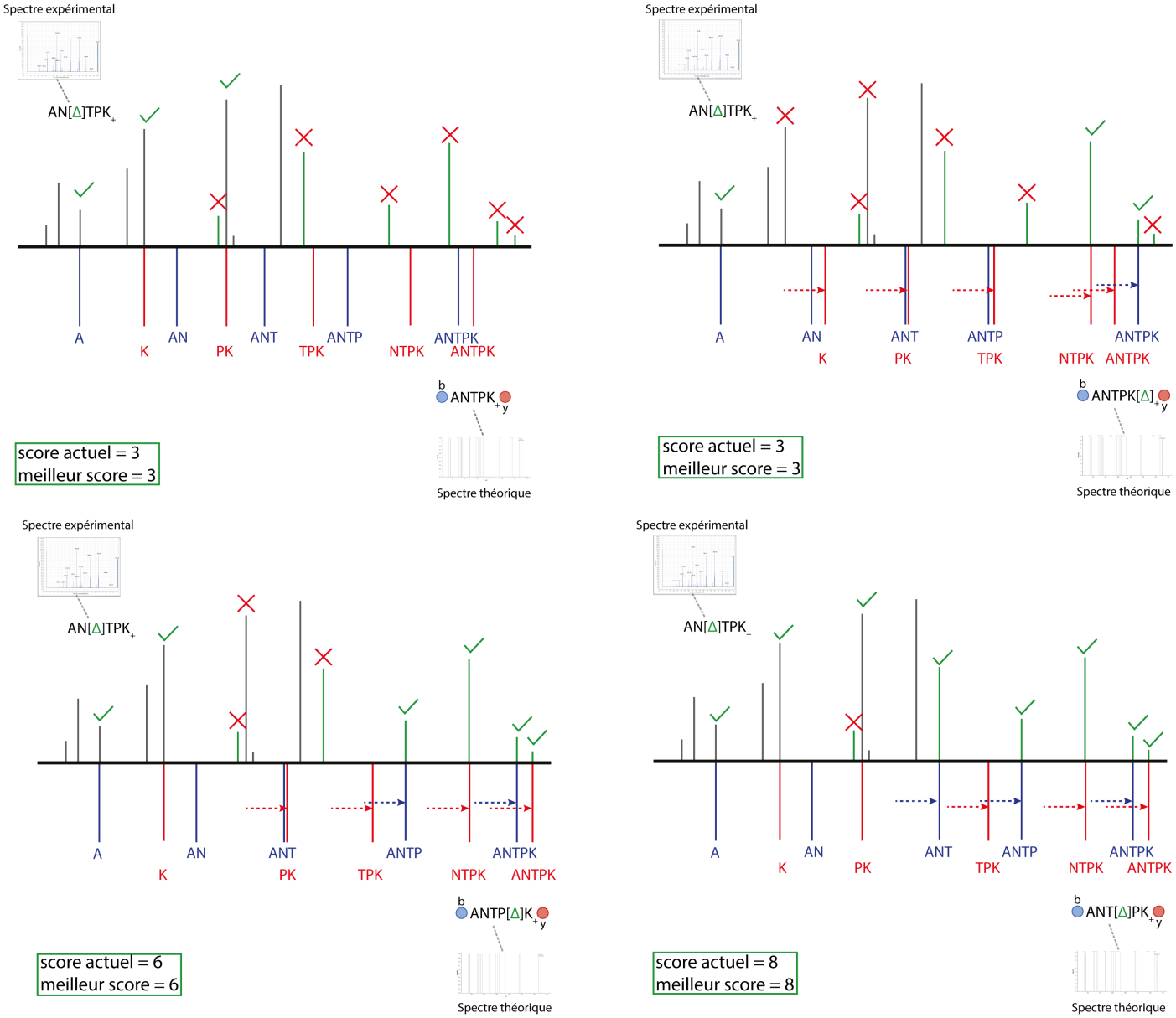


FIGURE 5.13 : Détection des modifications effectuée par SpecFit.
Ce schéma illustre quatre positions différentes d'une modification testée par SpecFit dans le peptide ANTPK et le déplacement des pics correspondant dans le spectre théorique. Au-dessus des axes des abscisses se trouve une représentation du spectre théorique avec une indication des masses partagées pour chaque masse. En-dessous de l'axe des abscisses se trouvent les masses calculées des spectres théoriques modifiés. Le score total est rappelé dans le cadre vert pour chaque position testée : le meilleur PSM est celle suggéré lorsque la modification porte sur l'acide aminé T.

Une fois les PSMs candidats calculés par SpecXtract, puis éventuellement générés et enfin filtrés par SpecFit, SpecOMS sauvegarde les résultats dans un fichier de type csv sous la forme d'un enregistrement comportant les champs suivants.

- L'identifiant du spectre expérimental s_i
- La séquence du peptide p ayant conduit à la construction du spectre théorique s_j
- La valeur $S(s_i, s_j)$ observée avant exécution de SpecFit
- La valeur $S(s_i, s'_j)$ observée après exécution de SpecFit

- Le delta de masse $\Delta(s_i, s'_j)$
- La meilleure position pour la modification liée à cette différence de masse dans la séquence du peptide
- L'indication de la présence d'un éventuel missed-cleavage ou semi-tryptique et l'extrémité du peptide concernée. Dans le cas d'un missed-cleavage, SpecFit reporte également la séquence du peptide voisin.
- L'identifiant de la protéine à partir de laquelle a été obtenu le peptide (éventuellement plusieurs dans le cas d'un peptide présent dans plusieurs protéines) et son origine (banque target ou banque decoy)

5.7 Détails d'implémentation

La section précédente fournit une description fonctionnelle "haut niveau" de la structure de données SpecTrees. L'implémentation de SpecTrees respecte cette description fonctionnelle mais diffère dans les structures de données mises à contribution. Nous explicitons ci-dessous quelques concepts d'implémentation en nous focalisant sur ceux indispensables au fonctionnement efficace en temps et en mémoire de SpecOMS.

5.7.1 Implementation SpecTrees

Les arbres de la structure de données SpecTrees sont appelés à contenir de très nombreux nœuds. Leur implémentation par le biais de données structurées en mémoire comme des objets implique une importante consommation mémoire. Étant donné la volumétrie des données que l'on souhaite manipuler, cette consommation mémoire devient rapidement un handicap.

En outre, l'algorithme SpecXtract va réaliser des tâches peu coûteuses mais répétitives, faisant intervenir de nombreux accès mémoire. Il est préférable que les données accédées successivement lors des opérations d'extraction soient placées en mémoire de manière continue. On évite en effet ainsi une dégradation significative des performances dues aux mécanismes de fonctionnement des processeurs modernes (mise en cache). Cette contiguïté ne peut pas être assurée efficacement si chaque nœud de SpecTrees est implémenté exactement comme dans la description fonctionnelle.

Toutes ces remarques portent sur les noeuds de SpecTrees mais s'appliquent également à accessLists{ } : les listes chaînées ne garantissent en général pas la contiguïté mémoire de l'information qu'elles contiennent et impliquent une consommation mémoire supplémentaire, ce qui est dans notre cas un handicap conséquent.

Pour contourner les difficultés liées à l'utilisation de données structurées, SpecTrees est implémenté en utilisant un ensemble de tableaux de type primitif, c'est-à-dire des tableaux stockant des types de base disponibles dans le langage utilisé (ici, principalement des entiers). Afin de pouvoir respecter la description fonctionnelle de SpecTrees en utilisant cette implémentation tabulaire, il faut attribuer à chaque nœud v de SpecTrees un identifiant unique. Cet identifiant sera par la suite noté n_{id}.

L'indexation des nœuds est séquentielle et se fait au fur et à mesure de la création de la structure et de l'insertion de ces nœuds par l'algorithme SpecGrowth. Pour chaque nœud v, son identifiant $n_{id}(v)$ coïncide avec l'index auquel les données de ce nœud v sont stockées dans les différents tableaux. La Figure 5.14 illustre l'indexation des nœuds. Cinq tableaux d'entiers sont nécessaires pour recréer le comportement décrit pour SpecTrees. Ces entiers sont positifs, nuls ou égaux à -1, cette dernière valeur étant réservée pour représenter une absence de donnée.

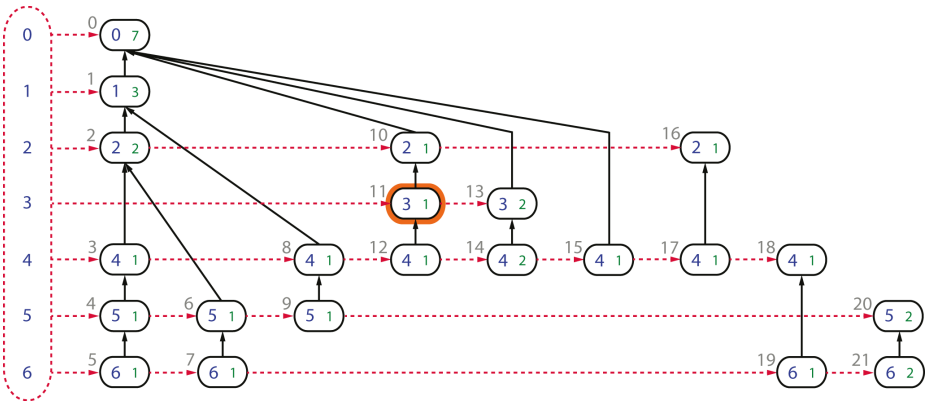


FIGURE 5.14 : Représentation conceptuelle de la structure de données SpecTrees avec ses nœuds indexés par des identifiants. Les identifiants des nœuds ont été rajoutés en gris en haut à droite de chaque nœud. Le nœud numéro 11 bordé d'orange est pris pour exemple dans la suite.

Le tableau specID[ ] contient les identifiants de spectres s_{id} présents dans les nœuds. Pour un nœud v donné, la valeur $s_{id}(v)$ est stockée à la position $n_{id}(v)$ dans specID[ ]. Une valeur de -1 dans ce tableau est utilisée pour représenter une absence de nœud et donc un identifiant inexistant.

Le tableau counter[ ] contient les compteurs présents dans les nœuds. Pour un nœud v donné, la valeur $c(v)$ est stockée à la position $n_{id}(v)$ dans counter[ ]. Une valeur de -1 dans ce tableau est utilisée pour représenter une absence de nœud et donc un compteur inexistant.

Le tableau parent[ ] permet de conserver la relation entre un nœud et son parent dans la structure arborescente. Soient deux nœuds v et v' dans SpecTrees pour lesquels v' est le nœud père de v. Le tableau parent[ ] contient à la position $s_{id}()$ la valeur $s_{id}(v')$. Une valeur de -1 signifie que le nœud v est à la racine d'un arbre.

Le tableau next[ ] permet de simuler le passage d'un nœud au suivant en utilisant une liste chaînée de accessLists{ }. Soient deux nœuds v et v' d'une liste chaînée l tels que v' est le nœud immédiatement après v dans l. Le tableau next[ ] contient à la position $s_{id}(v)$ la valeur $s_{id}(v')$. Une valeur de -1 signifie que le nœud v est le dernier de la liste l.

Enfin, le tableau first[ ] permet d'avoir accès au premier nœud de chacune des listes chaînées de accessLists{ }. Pour un identifiant de spectre quelconque égal à i, le tableau first[ ] contient à la position i le premier nœud v de la liste chaînée associée à cet identifiant de spectre. Une valeur de -1 signifie que la structure ne contient aucun spectre avec cet identifiant, ce qui est possible uniquement dans le cas où ce spectre ne partagerait aucun pic en commun avec aucun autre spectre.

La Figure 5.15 illustre l'utilisation de ces tableaux pour stocker les données en obtenant un comportement identique à celui obtenu en utilisant des arbres et des listes chaînées.

5.7.2 Implémentation de SpecGrowth

Tout comme l'implémentation concrète de SpecTrees diffère de sa description formelle, SpecGrowth possède plusieurs subtilités d'implémentation qui permettent un accès et un remplissage rapide des tableaux qui composent la structure de donnée SpecTrees, et sans lesquelles le volume de données à traiter devient un facteur limitant.

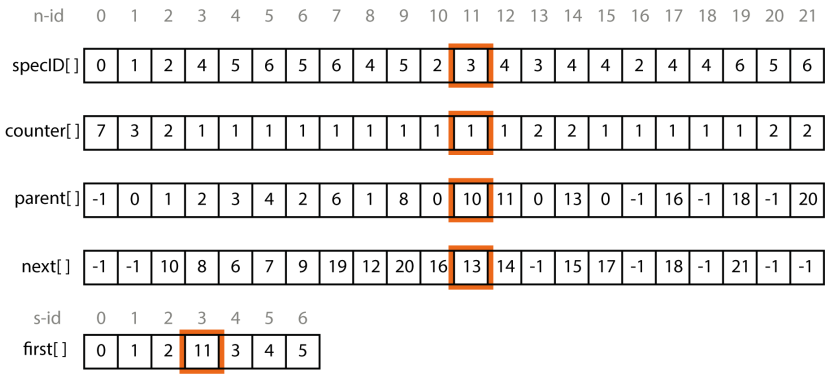


FIGURE 5.15 : Implémentation de la structure de données SpecTrees sous forme tabulaire. L'indexation des différents tableaux est rappelée en gris au-dessus de ces tableaux. Le nœud v bordé d'orange dans la Figure 5.14 ($n_{id}(v) = 11$) est mis en valeur. On peut voir dans SpecID[ ] le spectre associé ($s_{id}(v) = 3$), dans counter[ ] le compteur ($c(v) = 1$), son parent est le nœud v' tel que $n_{id}(v') = 10$ et le nœud suivant (next[ ]) dans la liste 3 de accessLists{ } est le nœud v'' tel que $n_{id}(v'') = 13$. Enfin, on peut voir que v est le premier nœud de la liste 3 dans accessLists{ }.

Allocation de la structure

Travailler avec des tableaux implique de choisir une alternative entre des tableaux statiques dont la taille est fixée à l'avance ou bien des tableaux dynamiques dont la taille peut croître en fonction du nombre d'éléments à y stocker.

Les tableaux dynamiques possèdent certes leur part d'avantages (simplicité d'usage, robustesse) mais dans le cas de gros volumes de données, ils s'accompagnent également de certaines contraintes. En particulier, le processus d'agrandissement de ces tableaux pour accueillir de nouveaux éléments fait intervenir une étape de recopie, laquelle consomme à la fois du temps mais aussi (et surtout) au moins le double de l'espace mémoire originellement occupé par le tableau (parfois nettement plus) pendant la durée de cette recopie. Pour cette raison, SpecTrees utilise des tableaux de taille statique, plus économes en temps et généralement en mémoire, à condition de minimiser l'espace vide dans ce tableau statique, en calculant sa taille au préalable au plus proche de l'occupation réelle.

Une contrepartie à l'efficacité des tableaux de taille statique est la nécessité de calculer à l'avance le nombre d'éléments maximum qu'ils contiennent. Dans le cadre de SpecGrowth, les tableaux peuvent être indexés par deux ensembles de valeurs : les identifiants des spectres et les identifiants des nœuds.

Allouer une taille minimale suffisante pour les tableaux indexés par les identifiants de spectres est simple : les spectres insérés dans SpecTrees sont indexés et comptés avant le Bucket Clustering et le nombre de spectres peut être utilisé comme taille de tableau.

Dans le cadre des tableaux indexés par les identifiants de nœuds, une borne supérieure est utilisée pour assurer que les tableaux possèdent une taille suffisante pour stocker l'ensemble des données. Dans le pire des cas, le processus de construction va aboutir à la création d'un nœud pour chaque identifiant de spectre de chaque bucket. Ce nombre de nœuds maximal est utilisé pour la construction de ces tableaux, ce qui peut conduire à une utilisation mémoire supérieure à la taille nécessaire, mais qui reste négligeable devant le coût de recopie que pourraient entraîner des tableaux dynamiques. En effet, la structure SpecTrees va contenir plusieurs centaines de milliers de nœuds, et il est possible d'établir une borne supérieure relativement précise sur cette quantité de manière à ce que la quantité de mémoire inutilisée des tableaux statiques reste petite devant l'utilisation mémoire entraînée par une recopie.

Ajout dans accessLists{ }

Lors de la construction de Spectrees, chaque nœud v nouvellement créé est ajouté à la suite d'une liste l contenue dans accessLists{ }. Trouver le premier élément de la liste correspondante est rapide ; en effet, le nœud possède un identifiant de spectre $s_{id}(v) = i$ qui permet d'accéder au premier élément de la liste l stocké à la position first[i]. Parcourir la liste chaînée l pour trouver la dernière position (ce qui revient à réaliser de multiples accès non consécutifs dans le tableau next[ ]) est néanmoins inefficace et il n'est pas souhaitable de procéder ainsi à chaque insertion de nœud.

Pour assurer une construction rapide de SpecTrees, l'algorithme SpecGrowth maintient une référence vers le dernier nœud ajouté dans chacune des listes de accessLists{ } dans le tableau last[ ]. Ainsi, n'importe quelle nouvelle insertion peut immédiatement être effectuée à la fin de la liste appropriée sans la reparcourir intégralement. Cette référence est stockée dans le tableau next[ ], qui comme le tableau first[ ], est indexé par les identifiants de spectres. Chaque fois qu'un nouveau nœud v tel que $s_{id}(v) = i$ est ajouté dans la structure, le dernier nœud v' inséré dans la liste l voit la valeur $n - id(v)$ assignée à $next[v']$, c'est-à-dire que l'on assigne v comme nœud suivant de v' dans la liste chaînée. Ensuite, la valeur $last[i]$ est remplacée par $n_{id}(v)$ (on remplace le nœud v' par le nœud v comme dernier élément inséré dans la liste l).

Une fois la procédure de construction de SpecTrees complète, les données contenues dans le tableau last[ ] ne sont plus nécessaires et peuvent être libérées de la mémoire.

Insertion des buckets

Parce que les nœuds de SpecTrees sont connectés depuis un enfant vers son père, il n'est pas possible lors de l'insertion de nouveaux nœuds de simplement partir de la racine de la structure et de la parcourir vers les feuilles pour insérer de nouveaux nœuds ou incrémenter le compteur de nœuds déjà présents.

Pour contourner l'impossibilité de parcourir les arbres de SpecTrees de leur racine vers les feuilles, SpecGrowth exploite le fait que le contenu de chaque bucket est trié par ordre d'identifiants de spectres croissants, et que l'ensemble des buckets est trié par ordre lexicographique croissant. Lors de l'insertion d'un nouvel identifiant dans l'arbre, il y a soit création d'un nouveau nœud, soit incrément du compteur d'un nœud existant dans la branche la plus récemment ajoutée à la structure. De plus, les nœuds qui se situent dans cette branche la plus récemment ajoutée correspondent en fait aux identifiants de spectre du bucket précédemment ajouté à SpecTrees par l'algorithme SpecGrowth. Il est donc possible de conserver une trace de l'insertion du contenu du dernier bucket pour savoir quels sont les nœuds dans la branche la plus récemment ajoutée, et de procéder à l'insertion des identifiants du nouveau bucket en comparant les spectres dans l'ordre. Si les deux buckets comparés comportent des parties préfixes communes, les identifiants de spectres de ces parties préfixes aboutiront à la fusion de nœuds avec incrément de compteurs. Les identifiants de spectre qui ne sont pas en commun dans la partie préfixe conduiront en revanche à la création de deux branches distinctes.

Algorithme SpecGrowth

L'Algorithme 1 décrit formellement le processus de construction de SpecTrees réalisé par SpecGrowth tel que nous l'avons expliqué dans les sections précédentes. Après initialisation des différentes variables, on parcourt le premier bucket dans l'ordre des identifiants (ligne 1) et pour chaque identifiant, un nouveau nœud est créé (lignes 2-4). Ensuite, le reste des buckets est parcouru dans l'ordre (ligne 5). Pour chaque bucket, nous effectuons une comparaison avec le contenu du bucket précédemment inséré dans SpecTrees : tant que les identifiants à insérer appartiennent à une partie préfixe du bucket (ligne 7), on ne créée pas de nouveaux nœuds pour ces identifiants et on incrémente à la place les compteurs des nœuds existants (lignes 8-9).

Une fois que les identifiants ne correspondent plus à une partie préfixe entre le bucket actuellement inséré et le bucket précédemment inséré (ligne 10), alors on insère les identifiants dans de nouveaux nœuds, qui constituent une nouvelle branche dans SpecTrees (lignes 11-15). La dernière ligne (ligne 16) correspond à une réinitialisation des variables pour le parcours des buckets suivants.

Algorithm 1 Algorithme SpecGrowth

Entrée : un Bucket Clustering $BC = \{B_0, B_1 \dots B_x\}$ contenant N_I identifiants de spectres dont N_D sont distincts deux à deux.
Sortie : Structure SpecTrees, i.e. des tableaux d'entiers $specID[\,]$, $counter[\,]$, $parent[\,]$, $next[\,]$ de taille N_I, et $first[\,]$ de taille N_D.
Variables : tableau d'entiers $last[\,]$ de taille m, entiers $pre = -1$, $in = 0$ et pos.

1: **for all** $id \in B_0$ **do** ▷ Un nœud est inséré pour chaque identifiant du premier bucket
2: $specID[in] \leftarrow id\,; counter[in] \leftarrow 1\,; parent[in] \leftarrow pre;$
3: $first[id] \leftarrow in\,; last[id] \leftarrow in;$
4: $pre \leftarrow in\,; in \leftarrow in + 1;$
5: **for** i de 1 à x **do** ▷ Pour chaque bucket, la position et le parent sont réinitialisés
6: $pre \leftarrow -1\,; pos \leftarrow 0;$
7: **while** $B_i[pos] = B_{i-1}[pos]$ **do** ▷ B_i et B_{i-1} partagent une partie préfixe commune
8: $counter[last[B_i[pos]]] \leftarrow counter[last[B_i[pos]]] + 1;$
9: $pre \leftarrow last[B_i[pos]]\,; pos \leftarrow pos + 1;$
10: **while** $pos < |B_i|$ **do** ▷ Maintenant le reste de B_i diffère de B_{i-1}
11: $specID[in] \leftarrow B_i[pos]\,; counter[in] \leftarrow 1\,; parent[in] \leftarrow pre;$
12: $next[last[B_i[pos]]] \leftarrow in;$
13: $last[B_i[pos]] \leftarrow in;$
14: **if** $first[B_i[pos]] = -1$ **then**
15: $first[B_i[pos]] \leftarrow in;$
16: $pre \leftarrow in\,; in \leftarrow in + 1\,; pos \leftarrow pos + 1;$

5.7.3 Implémentation de SpecXtract

Le calcul des similarités réalisé par SpecXtract exécute un très grand nombre d'opérations, mais celles-ci restent simples et efficaces : accès directs dans des tableaux, addition et comparaison d'entiers. La principale difficulté rencontrée dans la conception de SpecXtract concerne l'extraction et la manipulation de l'information. En effet, pour chaque spectre s_i contenu dans la structure SpecTrees, un nombre de masses partagées avec chaque autre spectre s_j tel que $i > j$ va être calculé. Pour éviter de stocker autant d'accumulateurs que de paires de spectres (s_i, s_j), seul un ensemble d'accumulateurs de taille m (avec m le nombre d'identifiants spectres distincts deux à deux présents dans le bucket clustering) sera créé, et réinitialisé par la procédure $read_i$ effectuée pour chaque spectre s_i.

Les accumulateurs utilisés par SpecXtract lors de chaque itération de la procédure $read_i$ seront donc stockés dans le tableau acc[ ]. L'indexation de acc[ ] correspond aux identifiants de spectre, le compteur pour la similarité de la paire (s_i, sj) étant donc stocké à l'index j lors de l'exécution de la procédure $read_i$. Pour ne pas avoir à réinitialiser la totalité du contenu de acc[ ] à chaque procédure, ce qui serait coûteux en temps de calcul, un second tableau iter[ ] construit selon les mêmes modalités de taille et d'indexation va être utilisé. Ce tableau servira à stocker la valeur de i à l'index j chaque fois que la valeur acc[j] est modifiée. Ainsi, si iter[j]=i, alors on sait que la valeur calculée est récente (elle est à jour pour l'itération i de la procédure $read_i$) et doit être utilisée telle quelle. Dans le cas contraire, on sait qu'elle doit d'abord être réinitialisée à 0 car elle date d'une exécution précédente de la procédure de lecture.

Après chaque opération $read_i$, seule une partie des paires de spectres (s_i, s_j) doit être retournée : celle qui contient les paires telles que $S(s_i, s_j) > T$. Pour éviter de parcourir l'intégralité de acc[ ] et iter[ ] à la fin de chaque exécution de la procédure de lecture, les identifiants des spectres théoriques vont être stockés dans une pile nommée $idStack$. Les PSMs reportés par SpecXtract seront reconstitués à la fin de chaque procédure $read_i$ en dépilant cette pile et en accédant dans le tableau acc[ ] la valeur de similarité à l'index $acc[pop(idStack)]$.

Algorithme SpecXtract

L'Algorithme 2 décrit formellement le processus de calcul des similarités entre spectres réalisé par SpecXtract tel que nous l'avons expliqué dans la section précédente. Après initialisation des différentes variables ainsi que de la collection d'accumulateurs et de témoins (ligne 1-3), on parcourt l'ensemble de accessLists{ } en commençant par l'indice de spectre le plus élevé et jusqu'à avoir atteint le spectre expérimental de SpecTrees d'indice le plus petit (ligne 4). Pour chaque nœud v rencontré dans chacune des listes chaînées de accessLists[ ] (ligne 6), on parcourt le chemin de ce nœud jusqu'à la racine (ligne 8) et pour chaque identifiant de spectre i rencontré dans chaque nœud w lors de ce parcours, on effectue un traitement spécifique. Si il existe un accumulateur dans acc qui porte l'identifiant de spectre du nœud w (ligne 9), cela signifie que cet accumulateur a déjà été utilisé pour cette paire de spectres : on ajoute la valeur du compteur du nœud v à l'accumulateur. Sinon (ligne 14), on entame un nouveau décompte pour cette paire de spectres en initialisant l'accumulateur à la valeur du compteur du nœud v. Chaque fois qu'un compteur est incrémenté (ligne 10) ou initialisé (ligne 15), si celui-ci dépasse le seuil (ligne 11 et ligne 17), on empile les identifiants des spectres qui correspondent (ligne 12 et ligne 18) pour les retrouver facilement après parcours afin de reporter les PSMs résultants. Les témoins correspondant aux accumulateurs modifiés sont mis à jour (ligne 13 et ligne 19). A la fin de chaque parcours de chemin (ligne 20) ou de liste chaînée (ligne 21) ; on réinitialise les positions initiales pour l'itération suivante. Enfin (ligne 22), les identifiants stockés dans la pile sont utilisés pour créer les PSMs (ligne 23-24) et ceux-ci sont reportés en résultat (ligne 25).

Maintenant que nous avons exposé dans le détail quels sont les éléments qui constituent la structure de données SpecTrees et fourni une description des différents algorithmes utilisés pour sa construction et son exploitation de manière efficace, nous allons dans le chapitre suivant présenter les résultats obtenus avec l'utilisation de ces différents composants dans le logiciel SpecOMS.

Algorithm 2 Agorithme SpecXtract

Entrée : Structure SpecTrees, i.e. tableaux d'entiers $specID[\,]$, $counter[\,]$, $parent[\,]$, $next[\,]$ de taille N_I, et $first[\,]$ de taille N_D, $N_{t_{max}}$ le plus grand identifiant de spectre théorique.
Sortie : $PMSs$ un ensemble de triplets d'entiers $(s_i, s_j, S(s_i, s_j))$.
Variables : tableaux d'entiers $acc[\,]$, $temoin[\,]$ de taille N_D, pile d'entiers $idStack$ entiers $start$, pos, τ et $temp$.

for i de 0 à $N_D - 1$ **do** ▷ Initialization
  $acc[i] \leftarrow -1\,; temoin[i] \leftarrow -1;$
$idStack \leftarrow \emptyset\,; PSMs \leftarrow \emptyset;$
for i de $N_s - 1$ à $N_{t_{max}} + 1$ **do** ▷ Itération par ordre décroissant des indices de spectres expérimentaux
  $start \leftarrow accessLists[i];$
  while $start \neq -1$ **do** ▷ Jusqu'à ce que la fin de la liste chaînée soit atteinte
    $pos \leftarrow start;$
    while $pos \neq -1$ **do** ▷ Jusqu'à ce que la racine de l'arbre soit atteinte
      if $temoin[specID[pos]] = i$ **then** ▷ Si l'accumulateur a déjà été modifié cette itération, on somme les valeurs
        $acc[specID[pos]] = acc[specID[pos]] + counter[start];$
        if $acc[specID[pos]] \geq \tau$ && $temoin[specID[pos]] \neq i$ **then** ▷ Si un accumulateur dépasse le seuil, l'index est stocké dans la pile et on le répertorie dans la liste de témoins
          $push(idStack, specID[pos]);$
          $temoin[specID[pos]] = i;$
      else ▷ Autrement, on commence une nouvelle somme
        $acc[specID[pos]] = counter[start];$
        $temoin[specID[pos]] = i;$
        if $acc[specID[pos]] \geq \tau$ && $temoin[specID[pos]] \neq i$ **then** ▷ Si un accumulateur dépasse le seuil, l'index est stocké dans la pile et on le répertorie dans la liste de témoins
          $push(idStack, specID[pos]);$
          $temoin[specID[pos]] = i;$
      $pos \leftarrow parent[pos];$
    $start \leftarrow next[start];$
  while $idStack \neq \emptyset$ **do**
    $temp \leftarrow pop(idStack);$ ▷ Dépilage et création des PSMs
    $PSMs \leftarrow PSMs \cap (s_i, s_{temp}, acc[temp]);$
return $PSMs;$

6

Efficacité et limitations de SpecOMS

Après avoir présenté dans le détail les algorithmes et la structure de données qui constituent SpecOMS, nous présentons dans ce chapitre les résultats théoriques et expérimentaux obtenus en utilisant SpecOMS. Nous introduisons dans un premier temps les jeux de données utilisés pour confirmer les discussions théoriques par des résultats expérimentaux. Dans un second temps, nous discutons des complexités spatiales et temporelles théoriques de l'algorithme, puis nous analysons et discutons les résultats expérimentaux qui les accompagnent. Nous les comparons enfin au temps d'exécution et à l'occupation mémoire de MSFragger [86], un autre logiciel d'identification de peptides OMS.

6.1 Jeux de données expérimentaux

6.1.1 Les spectres expérimentaux

Les expérimentations présentées dans la suite de ce manuscrit s'appuient sur un jeu de spectre expérimentaux téléchargé depuis la banque de données PRIDE (identifiant PXD001468). Ce jeu de spectres a été généré à partir d'un extrait cellulaire de cellules embryonnaires de reins humains (Human Embryonic Kidney Cells) connu sous le nom de HEK293. Cet extrait a été digéré par un mélance d'enzymes (la trypsine et la lysC) pour obtenir une forte efficacité proteolytique, c'est-à-dire éviter autant que possible les missed-cleavages. Le mélange de peptides obtenu a ensuite été séparé en 24 fractions sur un critère de polarité.

Les données MS/MS brutes téléchargées depuis PRIDE ont été converties vers le format MGF en utilisant le logiciel RawConverter [66] dans sa version 1.1.0.19. Nous utilisons pour les expérimentations discutées dans ce document les cinq premiers jeux de spectres (de b1906_293T_proteinID_01A_QE3_122212.raw à b1906_293T_proteinID_05A_QE3_122212.raw). Pour tester les performances de SpecOMS, plusieurs configurations (notées de A à E) ont été élaborées, comme présenté dans la Table 6.1. Dans la suite de ce manuscrit et sans précision complémentaire, par abus de langage, nous noterons souvent HEK293 les jeux de spectres analysés dans la configuration A.

6.1.2 Les Banques de protéines

Les spectres ont été identifiés avec SpecOMS en utilisant la banque de protéines Homo Sapiens GRCh37 du projet Ensemble Genome Assembly téléchargée depuis Uniprot. Cette banque contient 86 934 protéines. Dans nos expériences, nous avons considéré la présence d'un missed-cleavage. Les protéines conta-

	Jeux de spectres	
	Fichier de données brutes	Nombre de spectres expérimentaux
jeu de spectres A	b1906_293T_proteinID_01A_QE3_122212.raw	37 703
jeu de spectres B	b1906_293T_proteinID_01A_QE3_122212.raw b1906_293T_proteinID_02A_QE3_122212.raw	80 840
jeu de spectres C	b1906_293T_proteinID_01A_QE3_122212.raw b1906_293T_proteinID_02A_QE3_122212.raw b1906_293T_proteinID_03A_QE3_122212.raw	119326
jeu de spectres D	b1906_293T_proteinID_01A_QE3_122212.raw b1906_293T_proteinID_02A_QE3_122212.raw b1906_293T_proteinID_03A_QE3_122212.raw b1906_293T_proteinID_04A_QE3_122212.raw	163 154
jeu de spectres E	b1906_293T_proteinID_01A_QE3_122212.raw b1906_293T_proteinID_02A_QE3_122212.raw b1906_293T_proteinID_03A_QE3_122212.raw b1906_293T_proteinID_04A_QE3_122212.raw b1906_293T_proteinID_05A_QE3_122212.raw	206 797

TABLE 6.1 : Jeux de spectres de taille croissante élaborés à partir de la concaténation de plusieurs fichiers de spectres.

minantes traditionnellement rencontrées en spectrométrie de masse, au nombre de 116, ont été ajoutées à l'analyse en utilisant le fichier crap.fasta téléchargé depuis le site GPM (Global Proteome Machine, http ://www.thegpm.org/crap/index.html). Une banque decoy qui sert à évaluer statistiquement la fiabilité des identifications a été générée à partir des protéines en inversant leurs séquences d'acides aminés.

6.1.3 Construction des spectre théoriques par SpecOMS

Seuls les peptides dont la longueur varie entre 7 et 25 acides aminés ont servi à générer les spectres théoriques. Dans le cas de peptides identiques, SpecOMS n'en conserve qu'une copie mais prend bien garde à conserver une trace des différentes origines de chaque peptide : protéine et position dans la protéine.

Lorsqu'un ion parent est chargé plusieurs fois par le spectromètre de masse, il peut générer des ions fragments mono ou multichargés. La question s'est donc posée de représenter ces fragments multichargés lors de la modélisation des spectres théoriques par SpecOMS.

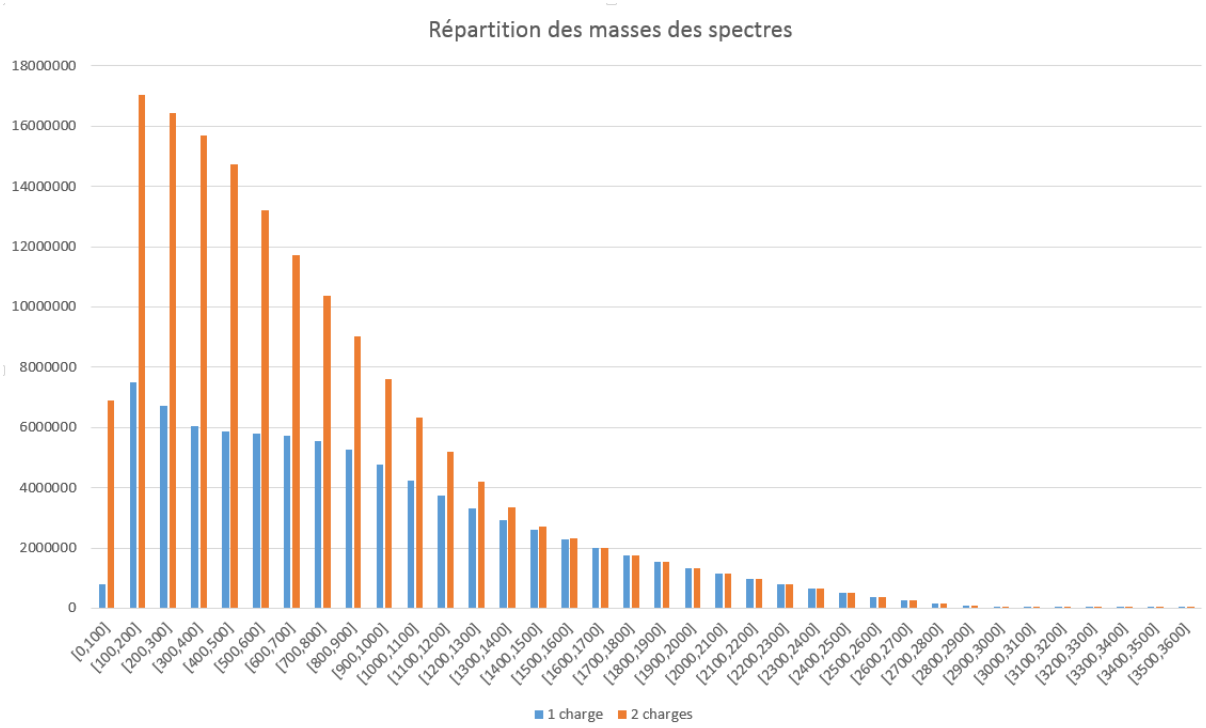


FIGURE 6.1 : Répartition des masses présentes dans les spectres obtenues à partir du dataset A et des spectres théoriques en fonction de la charge des fragments du spectre théorique.

La Figure 6.1 illustre la répartition des masses en intervalles de 100 Da en fonction des charges représentées lors de la construction de modèles théoriques. On peut observer que la densité des masses dans la partie de la courbe de 0 à 1000 Da environ est bien plus importante lors de la génération de spectres théoriques avec des fragments chargés deux fois que lors de la génération de spectres théoriques avec des fragments chargés une seule fois. Ce phénomène est dû à la nature de la mesure qui est en réalité un ratio masse sur charge, qui produit des mesures d'autant plus nombreuses dans les petites gammes de masse que la charge du spectre est importante. Générer des spectres théoriques avec des fragments chargés deux fois conduit donc à une augmentation significative du nombre de masses à prendre en compte par SpecOMS dans les spectres théoriques (un doublement des masses qui affecte les performances). Un biais est également très probablement introduit dans l'identification des peptides, la plus petite gamme de masse comportant un risque nettement plus important de correspondance due au hasard. Pour ces raisons, nous avons choisi de limiter la construction de spectres à des spectres théoriques une fois chargés.

6.1.4 Pré-traitement des spectres expérimentaux

Seuls les spectres expérimentaux dont la charge est 2+ et 3+ sont conservés. SpecOMS sélectionne les 60 masses les plus intenses dans chaque spectre expérimental en appliquant autour de chaque masse une fenêtre d'exclusion dont la largeur est égale à la précision choisie. Ce nombre de masses a été déterminé après une série de tests le faisant varier comme décrit en supplementary data dans notre publication [32]. Dans la fenêtre [45 ;70], les résultats évoluent peu, mas un nombre de masse de 60 donne néanmoins le meilleur taux d'identification.

6.2 Evaluation des performances techniques de SpecOMS

6.2.1 Evaluation de la complexité spatiale

Les performances théoriques de SpecOMS, c'est-à-dire la complexité temporelle et la complexité spatiale du logiciel dépendent toutes deux en très grande partie de la quantité de nœuds présents dans la structure de données SpecTrees (c.f. Section 5.1, page 60). En outre, la complexité temporelle dépend également du nombre de spectres expérimentaux à comparer avec l'ensemble des spectres présents dans SpecTrees. Nous allons dans un premier temps fournir une estimation au pire du nombre de nœuds présents dans SpecTrees sous la forme d'une borne supérieure. L'expression mathématique de cette borne supérieure sera utilisée pour décrire la complexité temporelle de SpecOMS à l'aide de la notation de Landau, mais on conservera l'expression complète pour décrire l'occupation mémoire de SpecOMS. En effet dans ce second cas les constantes ont leur importance pour la description de la consommation mémoire.

Le nombre de nœuds présents dans SpecTrees et leur disposition dans la structure sont influencés à la fois par le nombre total de spectres, le nombre de masses de chaque spectre, mais également par la répartition des identifiants de spectres dans les buckets à l'issue de l'étape de Bucket Clustering. Cette répartition impacte en effet la proportion de nœuds dans SpecTrees qui sont créés ou fusionnés et la disposition des arêtes entre nœuds dans la structure de données, et est difficile à estimer a priori. On peut cependant considérer une estimation maximale de celles-ci en considérant que chaque masse de chaque spectre inséré dans SpecTrees conduit à la création d'un nouveau nœud : c'est dans le cadre de cette hypothèse de travail que sont présentées les complexités temporelles et spatiales théoriques de SpecOMS. Cette hypothèse de travail implique cependant une surestimation importante du nombre de nœuds présents dans SpecTrees en comparaison avec les cas d'utilisation pratique. On notera également que cette hypothèse implique qu'à aucun moment le pré-traitement de la banque de protéines ne génère deux peptides identiques, ce qui est bien le cas.

En partant de l'hypothèse précédente, le nombre de nœuds dans SpecTrees dépend alors uniquement du nombre de masses dans les spectres insérés dans la structure de données par SpecGrowth. En effet, dans ces conditions chaque identifiant de spectre est présent dans les buckets autant de fois que ce spectre comporte de masses, et chacun de ces identifiants conduit à la création d'un nœud. Le nombre de nœuds est donc égal au nombre de masses total réparti dans les buckets lors du Bucket Clustering. Ce nombre total de masses est égal à la somme du nombre de masses issues des spectres théoriques et du nombre de masses issues des spectres expérimentaux.

Le nombre de masses issues des spectres théoriques dépend du nombre de spectres théoriques et du nombre de masse de chaque spectre. Ce paramètre dépend donc des données, mais il est possible de fournir une borne supérieure. En effet, SpecOMS limite la longueur des peptides traités en nombre d'acides aminés grâce au paramètre max et les spectres théoriques sont générés pour un ion parent chargé une seule fois. Pour chaque peptide, on génère donc au maximum $2max$ masses, une pour chaque ion b possible et une pour chaque ion y possible. Si le jeu de données contient S_t spectres théoriques produits à partir des peptides issus de la banque de protéines, le nombre maximum de masses possibles issues de spectres théoriques est donc dans un premier temps $2max \cdot S_t$.

Avant de procéder à la digestion de la banque de protéines *target*, SpecOMS réalise néanmoins la construction d'une banque *decoy*, laquelle contient les mêmes protéines dont les séquences ont été inversées. Cette manipulation a pour conséquence de doubler le nombre de spectres théoriques qui sont construits, conduisant à $4max \cdot S_t$ masses issues des spectres théoriques.

Si la station de travail sur laquelle SpecOMS est exécuté le permet, il est préférable d'intégrer la re-

cherche de missed-cleavages (résultant d'une digestion enzymatique imparfaite) en ajoutant les peptides dans la banque plutôt que de faire la recherche de ces missed-cleavages a posteriori. Le nombre de spectres augmente néanmoins en conséquence, SpecOMS permettant d'inclure jusqu'à 2 missed-cleavages dans une recherche. Dans ce cas on génère pour p peptides tryptiques issus de chaque protéine un nouvel ensemble de $(mc+1)\cdot p-(mc+1)$ spectres théoriques produits, mc étant le nombre de missed-cleavages recherchés. Pour simplifier, on utilisera à la place la borne supérieure $(mc+1)\cdot S_t$. La borne supérieure sur le nombre de masses issues des spectres théoriques, que l'on notera désormais m_t, devient donc alors $4max\cdot(mc+1)\cdot S_t$.

Le nombre de spectres expérimentaux à comparer avec l'ensemble des spectres présents dans la structure est fixe pour chaque jeu de données et correspond au nombre de spectres S_e obtenus après pré-traitement du fichier MGF. Le nombre de masses issues des spectres expérimentaux dépend également du nombre de masses de chacun de ces spectres. Ce paramètre dépend également des données et du paramétrage de SpecOMS, mais il est à nouveau possible de fournir une borne supérieure. SpecOMS permet de limiter le nombre de pics retenus dans un spectre à l'aide du paramètre $massCount$. Pour chacune de ces masses retenues, SpecOMS va générer $2\cdot\beta$ nouvelles masses supplémentaires pour prendre en compte l'incertitude de mesure du spectromètre de masse (β est un paramètre fourni par l'utilisateur). On peut donc borner le nombre de masses issues des spectres expérimentaux, que l'on notera m_e, par $S_e\cdot massCount\cdot 2\cdot\beta$.

La borne supérieure sur le nombre total de masses insérées dans les buckets par le Bucket Clustering est donc m_t+m_e soit $N_{up}=4max\cdot(mc+1)\cdot S_t+S_e\cdot massCount\cdot 2\cdot\beta$, ce qui correspond également au nombre maximal de nœuds qui peuvent être insérés dans SpecTrees.

6.2.2 Evaluation de la complexité temporelle

La complexité temporelle de SpecOMS peut être observée pour quatre algorithmes différents que sont le Bucket Clustering (c.f. Section 5.3, page 64), SpecGrowth (c.f. Section 5.4, page 66), SpecXtract (c.f. Section **??**, page **??**) et enfin SpecFit (c.f. Section 5.6, page 73).

Bucket Clustering

La complexité temporelle de l'algorithme Bucket Clustering dépend directement du nombre total de masses présentes dans chaque spectre expérimental ou théorique après prise en compte des paramètres de SpecOMS et de la répartition des identifiants de ces spectres dans les différents Buckets. On note N_{BC} le nombre de buckets, $S=S_e+S_t$ le nombre total de spectres, et N_{up} le nombre maximal de nœuds calculé précédemment. Les identifiants de spectres sont ajoutés dans chaque bucket par ordre d'identifiant croissant et le contenu de chaque bucket n'a pas besoin d'être trié après remplissage. En revanche, l'ensemble des buckets doit être trié par ordre lexicographique croissant des identifiants de spectres qu'ils contiennent. Par conséquent, la complexité de l'algorithme de Bucket Clustering peut s'exprimer comme $\mathcal{O}(N_{up}+N_{up}\log N_{up})$, soit $\mathcal{O}(N_{up}\log N_{up})$. En pratique néanmoins, le nombre de buckets est nettement inférieur à N_{up}, chaque bucket contenant plusieurs masses (dont le nombre total est au pire N_{up}), et cette complexité est vraisemblablement très largement surestimée.

SpecGrowth

L'algorithme SpecGrowth itère sur chaque identifiant de spectre contenu dans chaque bucket du Bucket Clustering et réalise pour chacun de ces identifiants un nombre constant d'opérations. Le nombre de ces identifiants étant de l'ordre de $\mathcal{O}(S)$ comme illustré dans le paragraphe précédent, la complexité de l'algorithme SpecGrowth est elle aussi de l'ordre de $\mathcal{O}(S)$. Une estimation un peu plus fine de la complexité au pire de SpecGrowth peut être réalisée en utilisant la borne N_{up} définie au préalable. En effet, le nombre d'identifiants de spectres présents dans les Buckets à l'issue du Bucket Clustering est au pire égale à cette valeur, ce qui conduit à une complexité au pire de SpecGrowth de $\mathcal{O}(N_{up})$.

SpecXtract

Pour un spectre expérimental s_i d'identifiant i donné, la complexité temporelle de SpecXtract est linéairement dépendante du nombre A_i de nœuds situés sur l'ensemble des chemins allant des nœuds de SpecTrees contenant l'identifiant i aux racines des arbres contenus dans SpecTrees. Il est très difficile de déterminer A_i pour le cas général mais il est certain que $A_i < \mathcal{O}(N_{up})$. Ainsi, la complexité générale de SpecXtract est de l'ordre de $\mathcal{O}(N_{up} \cdot S_e)$. En pratique, A_i est très nettement inférieur à la borne N_{up} et par conséquent l'algorithme SpecXtract est bien plus rapide que ne le suggère cette complexité dans le pire des cas, comme nous allons le voir par la suite.

SpecFit

La complexité de l'algorithme SpecFit est fortement dépendante du paramètres de threshold T (c.f. Section 5.5.2, page 71) choisi par l'utilisateur. En théorie pour une valeur de T est très basse (1 ou 2), la complexité de SpecFit avoisine $\mathcal{O}(S_t \cdot S_e)$. En effet, de très nombreux spectres théoriques possèdent au moins deux masses partagées avec un spectre expérimental donné et il existe donc beaucoup de PSMs à analyser par SpecFit. En pratique, dès que T augmente un peu (4, 5 ou plus), ce nombre de PSMs à analyser diminue beaucoup pour un spectre expérimental donné.

6.2.3 Configuration des tests réalisés

Nous avons réalisé un ensemble de tests pour appréhender les temps de calcul expérimentaux et la consommation mémoire au regard des évaluations de complexité théorique réalisés dans les paragraphes précédents. Plusieurs configurations de jeux de données ont été élaborées à partir du jeu de données HEK293 comme présenté dans la Table 6.1 et de la banque de protéines GRCh37 (cf. Section 6.1.2).

Pour ces tests, sauf indication contraire explicite, nous avons choisi les paramètres suivants : les peptides générés digérés à partir de la banque de protéines (qui inclut les protéines decoy) en utilisant avec la trypsine comme enzyme de digestion sont conservés si leur longueur est comprise entre 7 et 25 acides aminés. Les missed-cleavages sont générés lors de la digestion et inclus à la recherche. Les spectres expérimentaux lus depuis le fichier MGF sont analysés uniquement si leur charge est comprise entre 1 et 3, et seules les 60 masses les plus intenses de chaque spectre expérimental sont conservées. L'algorithme SpecFit est autorisé à rechercher l'emplacement des modifications par décalages de masses. Pour obtenir de SpecOMS qu'il travaille avec une précision sur les fragments de 0.02 Da, nous avons fixé les paramètres $\alpha = 2$ et $\beta = 2$ (section 5.2.1 page 63). La valeur du threshold T (section 5.5.2 page 59) a été fixée à 7 et la valeur du score $S_m = 0$ (section 5.6 page 71) a été fixée à 8.

Les résultats ont été obtenus en exécutant la version 1.2.0 de SpecOMS sur un ordinateur portable doté de Windows 7 et équipé d'un processeur i7-4910MQ à 2.90GHz et de 32Go de mémoire vive.

6.2.4 Mesures de temps d'exécution

Répartition des temps d'exécution entre les différents algorithmes

Dans un premier temps, nous avons cherché à mesurer le temps d'exécution attribué à chacun des algorithmes constituant le logiciel SpecOMS. Il est très intéressant de constater (c.f. Table 6.2) que le temps de construction de la structure SpecTrees à partir de la lecture des banques de protéines jusqu'à obtention de la structure entièrement remplie demeure très faible devant la durée des autres tâches accomplies par SpecOMS (de l'ordre d'une minute). L'algorithme le plus couteux en temps d'exécution, et de très loin, reste l'extraction des PSMs par SpecXTract. A lui tout seul, SpecXTract utilise 47 minutes sur les 48 nécessaires à l'ensemble de l'exécution.

Précision (Daltons)	Bucket Clustering + SpecGrowth (secondes)	SpecXtract (minutes)	SpecFit (secondes)
0.02	53.41	46.95	5.52

TABLE 6.2 : Temps d'exécution de chacun des algorithmes qui composent SpecOMS.

Impact du nombre de spectres expérimentaux

Nous avons fait varier la quantité de spectres expérimentaux à l'aide des différents jeux de spectres décrits précédemment. Les résultats de cette manipulation sont présentés dans la Table 6.3, dans laquelle nous récapitulons également le nombre de spectres total (expérimentaux et théoriques) manipulés par SpecOMS lors de l'analyse de chacun des jeux de spectres. Ces résultats montrent une augmentation de la durée d'exécution de SpecOMS avec l'accroissement du nombre de spectres expérimentaux. Cette augmentation de la durée d'exécution ne croît néanmoins pas linéairement avec le nombre de spectres expérimentaux.

	spectres théoriques				
	+ jeu de spectres A	+ jeu de spectres B	+ jeu de spectres C	+ jeu de spectres D	+ jeu de spectres E
Nombre total de spectres	2 739 578	2 782 715	2 821 201	2 865 029	2 908 672
Temps d'exécution (minutes)	47.38	108.50	164.81	212.99	337.78

TABLE 6.3 : Temps d'exécution de SpecOMS pour cinq jeux de spectres incluant 2 701 875 spectres théoriques et un nombre croissant de spectres expérimentaux.

Impact de la précision des mesures des ions fragments

Nous avons ensuite mesuré plus précisément le temps d'exécution de chacun des algorithmes qui composent SpecOMS en faisant varier les valeurs des paramètres α et β en utilisant uniquement le jeu de spectres A afin de mesurer l'impact de la précision de mesure sur les fragments choisie par l'utilisateur sur la durée d'exécution du logiciel. Les résultats sont présentés dans la Table 6.4 où figure le temps d'exécution des différents algorithmes. Nous y avons regroupé le Bucket Clustering et SpecGrowth car ils contribuent de manière indissociable à la construction de la structure SpecTrees.

Précision (Daltons)	Bucket Clustering + SpecGrowth (secondes)	SpecXtract (minutes)	SpecFit (secondes)
0.01	52.47	43.05	1.99
0.02	53.41	46.95	5.52
0.03	54.78	55.85	24.21
0.04	56.66	60.94	52.68
0.05	57.94	64.54	103.08
0.06	60.64	68.02	141.02
0.07	61.10	73.92	244.73
0.08	62.57	77.97	366.16
0.09	64.34	81.95	503.36

TABLE 6.4 : Temps d'exécution des différents algorithmes qui composent SpecOMS en fonction de la précision obtenue à partir des paramètres α et β.
Le jeu de données comprend 2 701 875 spectres théoriques et les 37 703 spectres expérimentaux issus du jeu de spectres A.

Concernant l'algorithme de Bucket Clustering et l'algorithme SpecGrowth, nous observons une légère augmentation du temps de calcul (de l'ordre de quelques secondes) lorsque la précision évolue de 0.01 Da

à 0.09. Cette augmentation, d'allure linéaire, est proche de 10% du temps d'exécution de ces deux algorithmes. L'algorithme SpecXtract exhibe lui aussi une augmentation de temps de calcul d'allure globalement linéaire avec les valeurs de précision évoluant de 0.01 Da à 0.09 Da. Le comportement de l'algorithme SpecFit diffère : on observe en effet une augmentation nettement plus prononcée de la durée d'exécution de l'algorithme avec l'évolution de la précision de 0.01 Da à 0.09 Da. Au total, le temps d'exécution de SpecFit aura été multiplié par près de 250. Nous pouvons également constater que le temps de construction de la structure SpecTrees à partir de la lecture des banques de protéines jusqu'à obtention de la structure entièrement remplie demeure faible dans tous les cas devant la durée des autres tâches accomplies par SpecOMS (de l'ordre d'une minute).

Impact du nombre de spectres théoriques

Enfin, nous avons réduit le nombre de spectres théoriques présents dans la structure SpecTrees. Pour ce faire, nous avons utilisé le jeu de spectres A comme précédemment mais nous avons retiré la génération des peptides avec missed-cleavage lors de la digestion enzymatique : les missed-cleavage sont recherchés a posteriori par l'algorithme SpecFit.

Précision (Daltons)	Bucket Clustering + SpecGrowth (secondes)	SpecXtract (minutes)	SpecFit (secondes)
0.01	29.85	11.91	0.96
0.02	29.52	13.07	2.11
0.03	31.57	16.22	8.58
0.04	31.89	17.88	18.58
0.05	34.02	19.30	33.67
0.06	35.60	20.76	56.21
0.07	38.66	22.44	86.98
0.08	39.50	23.93	132.78
0.09	39.96	26.47	192.88

TABLE 6.5 : Temps d'exécution des différents algorithmes qui composent SpecOMS en fonction de la précision obtenue à partir des paramètres α et β.
Le jeu de données comprend 1 055 951 spectres théoriques et 37 703 spectres expérimentaux.

Ces données exhibent les mêmes tendances que dans le cas précédent, où les peptides avec missed-cleavages sont générés au moment de la digestion enzymatique. Le temps d'exécution du Bucket Clustering et de SpecGrowth reste petit devant le temps d'exécution total de SpecOMS et augmente relativement peu avec l'évolution de la précision de 0.01 Da à 0.09 Da. Le temps d'exécution de SpecXtract augmente également de manière à peu près linéaire et atteint pour 0.09 Da un peu plus du double de la durée nécessaire à son exécution à 0.01 Da. Enfin, bien que la durée de SpecFit conserve également la même allure générale, le temps d'exécution entre 0.01 Da et 0.09 Da étant multiplié par près de 200.

Analyse des résultats

L'algorithme de Bucket Clustering, SpecGrowth et SpecFit prennent à peu près deux fois moins de temps à s'exécuter lorsque le nombre de spectres théoriques est environ deux fois moindre. En revanche, le gain de temps pour l'exécution de SpecXtract est plus marqué, nécessitant environ quatre fois moins longtemps pour cette même réduction du nombre de spectres théoriques. Cette remarque est d'autant plus vraie que SpecXtract est de loin l'algorithme dans SpecOMS le plus long à exécuter, dépassant toujours les autres de plusieurs dizaines de minutes dans les cas présentés.

Finalement, avec l'évolution de la précision de 0.01 Da vers 0.09 Da, le temps d'exécution de SpecOMS augmente dans son ensemble. Cette augmentation ne se répartit néanmoins pas uniformément entre tous les algorithmes qui composent SpecOMS : SpecFit et SpecXtract sont les deux algorithmes dont le temps d'exécution augmente le plus sensiblement. Cet accroissement s'explique par le mécanisme de gestion de la précision. En effet, pour chaque augmentation d'un points de la décimale de précision (le paramètre β), deux nouvelles masses supplémentaires sont créées pour chaque masse de chaque spectre expérimental. Une dégradation importante de la précision entraîne donc l'ajout d'un nombre important de masses supplémentaires dans les Buckets, et par ricochets de nœuds dans la structure SpecTrees. On notera que dans le cadre d'une précision fournie aujourd'hui pour un spectromètre de masse de haute précision (0.02 Dalton), le temps d'exécution de SpecOMS reste tout à fait raisonnable (moins d'une heure). De manière plus générale, nous conseillons à l'utilisateur de SpecOMS de choisir une combinaison de paramètres α et β de sorte à ce que β ne comporte qu'un seul chiffre non nul et que la précision utilisée par SpecOMS soit aussi proche que possible de celle utilisée lors de l'acquisition des données. Par exemple une précision de 0.3 sera largement préférée à une précision de 0.25, laquelle introduirait beaucoup de nœuds supplémentaires dans la structure de données. Il s'agit évidemment d'un compromis, ce choix pouvant dégrader légèrement la sensibilité de SpecOMS. Dans le cas où l'utilisateur souhaite privilégier la sensibilité du logiciel, il est toujours possible de fournir à SpecOMS les paramètres exacts au prix d'un temps de calcul supérieur.

Dans des conditions d'exécution usuelles, SpecOMS s'exécute donc comme nous venons de l'illustrer en un temps parfaitement acceptable (moins d'une heure). Le temps de traitement des spectres expérimentaux est proportionnel au nombre de spectres expérimentaux, mais la durée d'exécution croît un peu plus vite que le nombre de spectres. Bien que pénalisant dans le cas de gros jeux de données, cet inconvénient peut être contourné. En effet, le temps de construction de SpecTrees reste dans tous les cas très faible. Par conséquent, il est préférable de limiter les exécutions de SpecOMS à quelques dizaines de milliers de spectres expérimentaux et de réaliser si nécessaire plusieurs analyses successives en séparant les jeux de spectres en sous-ensembles. A l'unité, chaque spectre théorique ajouté à SpecTrees a en général moins d'impact qu'un spectre expérimental (les spectres théoriques comprennent la plupart du temps moins de masses que les spectres expérimentaux et entraînent donc une augmentation de la structure moins conséquente). En revanche, les quantités de spectres théoriques manipulées sont souvent plus importantes, et on a pu constater que la durée d'exécution de SpecOMS augmente de façon importante avec le nombre de spectres théoriques supplémentaires générés en fonction de la configuration de SpecOMS.

6.2.5 Evaluation de la consommation mémoire

L'occupation mémoire de SpecOMS est la résultante de trois éléments : l'occupation mémoire de SpecTrees, l'occupation des structures de données intermédiaires utilisées par les différents algorithmes qui composent SpecOMS (Bucket Clustering, SpecGrowth, SpecXtract, SpecFit) et l'occupation mémoire due à la manipulation et au stockage des protéines, des spectres expérimentaux et théoriques, et des résultats fournis par SpecOMS.

Occupation mémoire de SpecTrees

L'occupation mémoire de la structure de données SpecTrees dépend du nombre de nœuds insérés dans la structure de données, nombre pour lequel nous avons précédemment proposé la borne supérieure N_{up}, la compression des données liée au contenu des données étant difficile à évaluer. A partir de cette borne N_{up}, il est possible d'évaluer l'espace mémoire total occupé par la structure de données SpecTrees.

Chaque nœud de SpecTrees nécessite de stocker quatre informations sous la forme d'entiers : l'identifiant du spectre et le compteur stockés dans le nœud, l'identifiant numérique du nœud parent ainsi que le nœud suivant dans la liste accessLists (section 5.2 page 61). Ces données sont organisées dans quatre

tableaux d'entiers, tous de taille égale à N_{up}. La liste accessLists est quant à elle implémentée sous forme d'un tableau d'entiers de taille $S_e + S_t$. L'occupation mémoire théorique totale de la structure de données SpecTrees est donc au maximum $4N_{up} + S_e + S_t$.

Occupation mémoire des structures temporaires

Les algorithmes utilisés par SpecOMS lors des différentes étapes du processus requièrent l'utilisation de structures de données intermédiaires consommant également de l'espace mémoire. Plusieurs grandes structures doivent être prises en compte.

Lors de l'étape de Bucket Clustering, la mémoire utilisée dépend uniquement du nombre de spectres et du nombre de masses dans chaque spectre. Dans le pire des cas, chaque masse de chaque spectre est distincte et il existe donc autant de buckets que de masses, soit N_{up} buckets, qui contiennent chacun un unique nombre entier. Après construction de SpecTrees, la machine virtuelle java est autorisée à effacer les structures de données du Bucket Clustering si nécessaire.

Lors de l'exécution de SpecGrowth, l'algorithme maintient le tableau d'entiers intitulé last[ ], de taille $S_e + S_t$, nécessaire à la construction de SpecTrees. La machine virtuelle java est autorisée à effacer ce tableau de la mémoire dès la construction de SpecTrees terminée.

Si le paramètre de threshold T de SpecOMS est petit, de nombreux PSMs peuvent être générés par SpecXtract. Bien que le format de stockage des PSMs générés par SpecXtract leur garantisse une occupation mémoire réduite (trois entiers pour le spectre expérimental, le spectre théorique et la similarité calculée), ce grand nombre de PSMs peut rapidement encombrer la mémoire. Pour contourner cette difficulté, SpecFit analyse tous les PSMs obtenus pour un spectre expérimental donné immédiatement après leur obtention par SpecXtract, sélectionne le meilleur et efface les autres. Ainsi, il n'est pas nécessaire de maintenir en mémoire un grand nombre de PSMs pendant toute l'exécution de l'algorithme SpecXtract.

L'espace mémoire temporaire utilisé par les différents algorithmes composant SpecOMS est donc négligeable devant le reste de l'espace mémoire nécessaire. De plus, cet espace est libéré au fur et à mesure au gré de la machine virtuelle de java si le maintien en mémoire des structures intermédiaires n'est plus nécessaire. Ainsi, cet espace ne sera pas un facteur limitant si l'utilisateur dispose de suffisamment de mémoire pour stocker SpecTrees et les données concernant les protéines et les résultats.

Représentation des protéines et des peptides

La version de SpecOMS décrite dans ce manuscrit n'optimise que peu la représentation des protéines et des peptides et il en résulte une consommation mémoire importante.

Actuellement, la banque de protéines est lue et est entièrement stockée en mémoire. A partir de celle-ci, on construit les séquences peptidiques par digestion enzymatique (éventuellement avec les peptides comportant des missed-cleavages si les paramètres appropriés sont sélectionnés). A ces séquences peuvent s'ajouter celles de la base decoy si l'utilisateur choisit de tester à la fois la banque decoy et la banque target. Un fichier permettant de relier les peptides à leur protéine d'origine est généré et stocké sur le disque. Une fois ces opérations effectuées, la banque de protéines est retirée de la mémoire. En revanche, on conserve en mémoire l'ensemble des séquences peptidiques.

Parmi les paramètres sélectionnés par l'utilisateur, l'usage ou non de la base decoy dans la recherche, ainsi que l'intégration à la recherche des missed-cleavages et leur nombre ontun impact multiplicatif. En effet, l'utilisation simultanée de la banque decoy en prenant en compte un ou deux missed-cleavages entraîne une multiplication du nombre de peptides à considérer. Une solution est proposée par SpecOMS à

l'utilisateur qui disposerait de très peu d'espace mémoire. Cette dernière consiste à intégrer la recherche des peptides avec missed-cleavages à l'algorithme SpecFit plutôt que de les générer lors de la digestion enzymatique des protéines. Il faut noter que cette solution peut dégrader la sensibilité de la méthode. En effet l'ensemble de spectres n'est pas mis en compétition avec l'ensemble des peptides possibles incluant les missed-cleavages puisque ceux-ci n'ont pas été générés et n'ont donc pas donné lieu à la création de spectres théoriques. Néanmoins, cette solution permet un gain de temps de calcul et surtout une diminution conséquente de l'espace mémoire requis.

Lors de l'exécution de SpecXtract, de multiples PSMs sont obtenus pour un spectre expérimental donné, mais un seul d'entre eux est conservé après l'exécution de SpecFit. Par conséquent, le résultat final produit par SpecOMS occupe rarement plus de quelques dizaines de mégaoctets dans un fichier une fois les index utilisés dans la structure SpecTrees remplacés par les identifiants du spectre expérimental et la séquence peptidique ainsi que les informations d'identification de SpecFit ajoutées.

6.2.6 Mesures de la consommation mémoire

La consommation mémoire de SpecOMS a été mesurée en exécutant le logiciel dans les mêmes conditions que lors de l'estimation du temps de calcul (section 6.2.2 page 88). Nous avons mesuré dans un premier temps la consommation mémoire totale de SpecOMS en faisant varier la quantité de spectres expérimentaux à l'aide des différents jeux de spectres décrits précédemment (section 6.1.1 page 83). Les résultats sont reportés dans la Table 6.6, dans laquelle nous récapitulons également le nombre de spectres total (expérimentaux et théoriques) manipulés par SpecOMS lors de l'analyse de chacun des jeux de spectres.

	spectres théoriques				
	+ jeu de spectres A	+ jeu de spectres B	+ jeu de spectres C	+ jeu de spectres D	+ jeu de spectres E
Nombre de spectres	2 739 578	2 782 715	2 821 201	2 865 029	2 908 672
Occupation mémoire (GO)	8.36	9.68	10.78	11.58	12.12

TABLE 6.6 : Consommation mémoire totale de SpecOMS pour cinq jeux de spectres incluant 2 701 875 spectres théoriques et un nombre croissant de spectres expérimentaux.

Comme on peut le constater, l'occupation mémoire globale de SpecOMS croît de manière proportionnelle au nombre de spectres expérimentaux. On peut également observer que la quantité de mémoire occupée par les spectres expérimentaux est petite devant le reste des données utilisées par SpecOMS, 30 000 spectres expérimentaux environ nécessitant ici moins de 1Go de mémoire.

Nous avons dans un second temps souhaité évaluer la différence de consommation mémoire sur le jeu de spectres A entre le cas où les peptides comportant des missed-cleavages sont générés lors de la digestion enzymatique et le cas où ces peptides sont recherchés a posteriori par SpecFit. Dans le cadre de cette expérience, nous avons également fait varier les paramètres α et β pour évaluer leur impact sur la consommation mémoire du logiciel. L'ensemble des résultats sont récapitulés dans la Table 6.7.

On peut constater que dans les deux cas (peptides avec un missed-cleavage générés lors de la digestion enzymatique ou bien recherché a posteriori par SpecFit), une évolution de la précision de 0.01 Da vers 0.09 Da entraîne une consommation mémoire de SpecOMS plus importante. Cette augmentation croît néanmoins de manière linéaire et se compte en mégaoctets pour chaque tranche de 0.01 Da dans le cas présent, ce qui reste relativement négligeable. En comparaison, on observe que pour une valeur de précision donnée (calculée à partir des paramètre α et β), prendre en compte les peptides avec un missed-cleavage dès la digestion enzymatique de la banque de protéines double la consommation mémoire de SpecOMS.

	Consommation mémoire (Go)								
Précision (Daltons)	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
missed-cleavage généré à la digestion enzymatique	8.2	8.8	8.9	9.2	9.4	9.6	9.8	9.9	10.0
missed-cleavage cherché a posteriori par SpecFit	4.2	4.3	4.4	4.6	4.8	4.9	4.9	5.0	5.3

TABLE 6.7 : Occupation mémoire totale de SpecOMS en fonction de la précision et du traitement des peptides avec un missed-cleavage.

Tout comme pour le temps d'exécution de SpecOMS, les spectres théoriques entraînent dans le cas général moins de consommation mémoire par spectre que les spectres expérimentaux, mais les quantités de spectres théoriques manipulés sont plus importantes. Les paramètres α et β ont un impact nettement moindre sur la consommation mémoire du logiciel. Comme on peut le voir, analyser un jeu de spectres demande tout de même de disposer d'une dizaine de Go de mémoire, ce qui aujourd'hui sans être rare n'est pas non plus toujours accessible sur tous les postes de travail. SpecOMS dispose néanmoins de la possibilité d'alléger la consommation mémoire en recherchant les peptides avec missed-cleavage a posteriori, ce qui réduit environ de moitié l'espace mémoire nécessaire à son exécution et le rend exécutable sur l'immense majorité des stations de travail.

7

Identification de spectres avec SpecOMS

Nous analysons dans ce chapitre les résultats fournis par SpecOMS lors de l'analyse du jeu de données HEK293 présenté précédemment. La discussion que nous proposons est dans un premier temps orientée autour de l'influence des principaux paramètres de SpecOMS. A partir de ces résultats, nous discutons les différents contextes d'utilisation du logiciel. Enfin, nous comparons les résultats obtenus avec ceux obtenus par MSFragger [86], un autre logiciel OMS d'identification de spectres. MSFragger et SpecOMS ont été développés pendant la même période et ils ont tous deux beaucoup amélioré les performances des approches OMS ciblées sur la comparaison avec des banques de protéines.

7.1 Mise à disposition du logiciel

En date de la publication du premier article concernant la structure de données SpecTrees sous-jacente à SpecOMS dans les actes de la conférence WABI [33], SpecOMS était accessible uniquement sous forme d'un programme exécutable en mode console et n'était pas diffusé. A l'occasion de la publication d'un second article, SpecOMS a été diffusé sur la plateforme GitHub (https ://github.com/matthieu-david/SpecOMS) dans l'onglet *Releases*. Les sources du logiciel n'ont pas été mises à disposition, mais un exécutable est téléchargeable, accompagné de fichiers permettant l'exécution d'un exemple très simple. Pour faciliter sa diffusion dans les laboratoires de spectrométrie de masse sans nécessiter l'intervention d'un informaticien, SpecOMS a été implémenté en Java et nécessite uniquement la présence d'une machine virtuelle java sur le poste exécutant le logiciel, un pré-requis facile à satisfaire. Le logiciel est lancé à l'aide d'un simple script qui initialise la machine virtuelle avec des paramètres optimisés pour une exécution efficace de SpecOMS, lesquels peuvent être modifiés par l'utilisateur en fonction de ses contraintes logicielles et matérielles.

Une interface graphique a été développée (travail réalisé par Venceslas Douillard, stagiaire de Master 2 Bioinformatique) pour améliorer la saisie des paramètres et accroître l'attractivité du logiciel auprès des biologistes. L'exécution en ligne de commande représente souvent en effet un obstacle à l'utilisation de logiciels scientifiques par la communauté. Cette interface graphique s'accompagne d'une commodité de sélection des fichiers de données, mais également d'une fonctionnalité de visualisation et de représentation des résultats que nous présenterons brièvement ultérieurement.

7.2 Impact des paramètres sur les résultats de SpecOMS

Nous nous proposons dans cette section d'évaluer l'influence des principaux paramètres de SpecOMS sur les identifications des spectres du jeu de données HEK293, puis nous apportons des éléments de réflexion pour aider à leur bon usage. Sauf indication contraire, les paramètres sont donc configurés comme présenté dans la Section 6.2.3. La Figure 7.1 présente l'interface de lancement de SpecOMS configuré en utilisant ces mêmes paramètres. Dans la suite de ce manuscrit, nous nous référons à cette configuration comme la configuration standard. Nous y avons mis en valeur en rouge les paramètres que nous allons faire varier dans la suite de cette section pour analyser leur impact sur la sensibilité de SpecOMS.

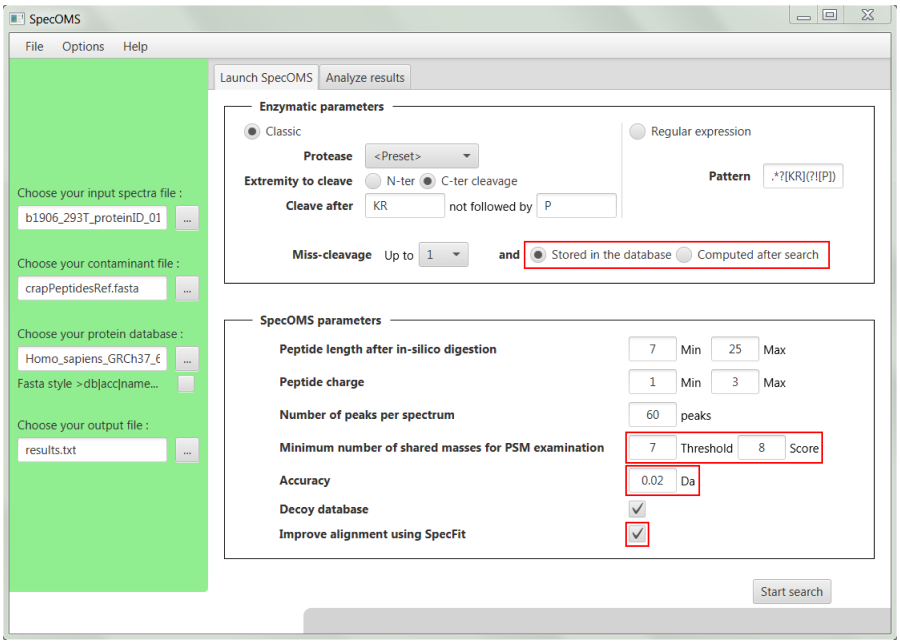


FIGURE 7.1 : Interface graphique du logiciel SpecOMS. Nous avons mis en valeur en rouge les paramètres que nous allons faire varier dans ce manuscrit pour analyser leur impact sur la sensibilité de SpecOMS.

Nous avons distribué les PSMs issus de chaque analyse en trois groupes en fonction de l'écart de masse entre le spectre expérimental s_i et le spectre théorique s_j : le groupe G1 est constitué des PSMs pour lesquels $\Delta(s_i, s_j) \in [-0.02; +0.02]$, le groupe G2 est constitué des PSMs pour lesquels $\Delta(s_i, s_j) > 0.02$, et enfin, le groupe G3 est constitué des PSMs pour lesquels $\Delta(s_i, s_j) < -0.02$. Dans ce chapitre, les PSMs sont validés en respectant une FDR locale à chacun des groupes < 1%.

7.2.1 Influence du ré-alignement de spectres avec SpecFit

SpecOMS offre la possibilité de ré-aligner les couples de spectres des PSMs présentant un écart de masse non nul grâce à l'algorithme SpecFit, ceci avant le choix du meilleur peptide candidat retenu. Ce ré-alignement permet d'améliorer le score de similarité de certains PSMs et de ce fait d'augmenter le nombre de peptides modifiés validés. En contrepartie, les essais multiples d'alignement entre le spectre expérimental et le peptide pourraient introduire du bruit (de nouveaux faux positifs), entrainer une augmentation du score pour respecter une FDR < 1% et en conséquence induire une perte de la sensibilité. Nous avons par conséquent cherché à évaluer l'influence de ce ré-alignement effectué par l'algorithme SpecFit sur la sensibilité de SpecOMS. Pour ce faire, nous avons tout d'abord réalisé l'analyse du jeu de données HEK293 avec la configuration standard de SpecOMS. Ensuite, nous avons analysé une seconde fois ce même jeu de données mais en désactivant le ré-alignement des masses effectué par SpecFit : ainsi, SpecFit ne recherche plus spécifiquement les peptides modifiés en essayant d'améliorer le score d'un PSM à l'aide du delta de

masse entre le spectre expérimental et le spectre théorique de ce PSM. Le reste des fonctionnalités de SpecFit demeure inchangé (par exemple la recherche de missed-cleavages). La Table 7.1 récapitule les résultats obtenus lors de cette analyse : le nombre d'identifications obtenues pour chaque groupe y figure, ainsi que le score auquel le seuil de $FDR < 1\%$ est atteint accompagné de la FDR exacte associée.

		avec ré-alignement des spectres	sans ré-alignement des spectres
G1	S_{FDR} FDR en % nombre d'identifications	7 0.87 12 694	8 0.85 12 541
G2	S_{FDR} FDR en % nombre d'identifications	11 0.73 3 821	11 0.68 2 498
G3	S_{FDR} FDR en % nombre d'identifications	19 0.00 38	11 0.67 1 202
G3'	S_{FDR} FDR en % nombre d'identifications	13 0.96 625	11 0.56 894
total (G1+G2+G3')	nombre d'identifications	17 140	15 933
	temps d'exécution (minutes) consommation mémoire (Go)	48.90 8.8	46.99 8.7

TABLE 7.1 : Influence du ré-alignement des spectres avec SpecFit. S_{FDR} représente la valeur de score seuil qui permet de respecter un taux de FDR < 1%.

Le nombre de spectres identifiés augmente légèrement avec l'utilisation de SpecFit pour les PSMs du groupe G1, ce qui est peu intuitif. En effet, lorsque SpecFit permet le ré-alignement, certains PSMs de score égal à 7 ($S_{\Delta m=0}$ retenu dans notre test) et de $\Delta(s_i, s_j)$=0 pourront être remplacées par d'autres PSMs candidats avec un score amélioré grâce à SpecFit. Cet effet doit donc logiquement diminuer le nombre de PSMs du groupe G1. Cependant, dans le même temps, le nombre d'erreurs d'identification restant parmi les PSMs de score=7 et de $\Delta(s_i, s_j)$=0 diminue également, permettant de baisser le score S_{FDR} respectant le seuil de FDR < 1%.

Comme attendu, le nombre d'identifications dans le groupe G2 augmente de manière significative (35% d'identifications en plus dans le cas présent). Ces identifications sont atteintes pour le même seuil de S_{FDR} et avec une valeur de FDR estimée très proche. Cette observation met en valeur une réelle efficacité de SpecFit pour améliorer les identifications de peptides modifiés, identifications qui seraient autrement perdues par SpecOMS.

Enfin, on observe un effondrement du nombre d'identifications obtenues dans le groupe G3. En effet, seules 38 identifications sont obtenues dans ce cas spécifique, avec un score S_{FDR} particulièrement élevé (19). Après un examen attentif des PSMs de ce groupe, nous constatons la présence d'un très grand nombre de PSMs identifiés dans la banque decoy et associés à des peptides de grande taille impliquant des valeurs élevées de deltas de masse négatifs. Une restriction des PSMs associés à des $\Delta(s_i, s_j)$ > -400 Da permet de retrouver une partie des identifications perdues. Ces tests mettent en évidence un biais de SpecFit dans le cadre de l'identification de PSMs pour lesquels le delta de masse entre les deux spectres est négatif. Nous proposons ci-après une hypothèse qui pourrait expliquer la dégradation de la sensibilité de SpecOMS dans ce contexte. Lorsque le delta de masse entre les deux spectres d'un PSM est négatif, cela signifie que le spectre théorique a une masse plus importante que le spectre expérimental. Le delta de masse étant négatif, les masses des ions b qui correspondent aux acides aminés à droite de la modification et des ions y

qui correspondent aux acides aminés à gauche de la modification dans le spectre théorique vont diminuer, créant un possible chevauchement avec des masses non modifiées du spectre théorique. Sur une plage de masses d'autant plus importante que le delta de masse est négatif, SpecFit va ainsi augmenter le nombre de pics présents, pouvant ainsi générer du bruit et induire par hasard quelques pics supplémentaires lors de l'alignement.

Au final, l'utilisation de SpecFit permet d'identifier 17 140 spectres répartis dans les trois groupes G1, G2 et G3', avec principalement une augmentation des identifications de peptides modifiés. On observe toutefois une difficulté quand $\Delta(s_i, s_j)$ est fortement négatif. Dans la suite de ce manuscrit, nous nous proposons donc d'abandonner la définition du groupe G3 et d'utiliser à la place le groupe G3' définit comme les identifications telles que -400 < $\Delta(s_i, s_j)$ < -0,02 (cf. Table 7.1). Ce seuil de -400 Da a été établi empiriquement pour ce jeu de données, et permet de récupérer quelques centaines d'identifications supplémentaires, mais il serait souhaitable d'implémenter dans les prochaines versions de SpecOMS un mécanisme dans SpecFit afin d'éviter que les PSMs avec des deltas négatifs ne dégradent autant les résultats. Par ailleurs, nous pouvons noter la faible utilisation des ressources du ré-alignement de spectres effectué par SpecFit.

7.2.2 Prise en compte des missed-cleavages

La digestion enzymatique des protéines pouvant être incomplète, elle est susceptible de générer des missed-cleavage. Leur prise en compte peut s'effectuer de deux manières différentes : (i) les missed-cleavages potentiels sont considérés lors de la digestion enzymatique et sont ensuite utilisés pour construire les spectres théoriques associés, (ii) les missed-cleavages sont évalués a posteriori par SpecFit à partir du delta de masse de chaque PSM, une fois les similarités entre spectres calculées par SpecXtract. Notons que dans le second cas, leur identification peut être favorisée par l'application du seuil $S_{\Delta m=0}$. L'inclusion des missed-cleavages dans la banque de spectres théoriques entraîne une augmentation conséquente du nombre de spectres, et l'on s'attend donc à ce qu'elle s'accompagne d'une dégradation du temps de calcul et d'une augmentation de la consommation mémoire augmentée. En contrepartie, nous espérons observer une augmentation de la sensibilité.

Nous avons donc cherché à évaluer l'influence de ce paramètre en analysant le jeu de données HEK293 dans les deux configurations. La Table 7.2 récapitule les résultats obtenus en terme de nombre d'identifications, de temps d'exécution et de consommation mémoire pour ces deux analyses.

		missed-cleavages générés à la digestion	missed-cleavages non générés à la digestion
groupe 1	S_{FDR}	7	7
	FDR en %	0.87	0.86
	nombre d'identifications	12 694	9 716
groupe 2	S_{FDR}	11	12
	FDR en %	0.73	0.61
	nombre d'identifications	3 821	3 436
groupe 3'	S_{FDR}	13	14
	FDR en %	0.96	0.99
	nombre d'identifications	625	302
total	nombre d'identifications	17 140	13 454
	temps d'exécution (minutes)	48.90	13.60
	consommation mémoire (Go)	8.8	4.3

TABLE 7.2 : Influence du traitement des missed-cleavages par SpecOMS.

Lorsque les missed-cleavages ne sont pas générés à la digestion, SpecOMS obtient bien moins d'identifications (9 716 identifications) dans le groupe G1 que lorsque les spectres théoriques correspondants aux peptides avec des missed cleavages sont générés (12 694 identifications). Cette différence de nombre d'identifications paraît logique. En effet, si un spectre expérimental correspond à un missed cleavage, le spectre théorique ayant été généré, l'identification produit un PSM associé à un $\Delta(s_i, s_j)$ nul, ce qui n'est pas le cas sinon. Cependant, on pourrait s'attendre à retrouver cette perte d'identifications dans le groupe G2. Or, on observe en cumulant les deux groupes G1 et G2 une réelle perte du nombre d'identifications lorsque les spectres théoriques qui correspondent aux missed-cleavages n'ont pas été générés.

La perte d'identifications lorsque les missed-cleavages ne sont pas générés à la digestion peut s'expliquer notamment par la présence de modifications qui affectent les peptides avec missed-cleavage. C'est un cas particulier d'un peptide comportant deux modifications, comme nous l'avons déjà illustré en Section 4.2, dans lequel le missed-cleavage joue le rôle d'une modification. Pour illustrer cet effet, nous proposons de considérer l'exemple du spectre 36 898. Lorsque SpecOMS est exécuté avec la génération de missed-cleavages lors de la digestion enzymatique de la banque de protéines, ce spectre est identifié par le peptide MIDMGFEPQIRK avec un delta de masse de 15.9946 Daltons. Ce peptide comporte un missed-cleavage en 11^{me} position et le delta de masse correspond très vraisemblablement à une oxydation de la méthionine en première position. Lorsque SpecOMS est exécuté sans générer les peptides avec missed-cleavages lors de la digestion enzymatique de la banque de protéines, ce spectre n'est pas identifié.

La Figure 7.2 illustre la modification des masses du spectre dans les deux cas suivants : lorsque SpecOMS génère les missed-cleavages lors de la digestion enzymatique et lorsqu'il ne le fait pas. Nous présentons sur plusieurs images les différents cas de figures qui peuvent se produire et mettons en évidence l'influence sur le nombre de masses partagées entre le spectre théorique (au-dessus de l'axe des abscisses) et le spectre expérimental (en dessous de l'axe des abscisses).

La première image (a) est une représentation du spectre théorique MIDMGFEPQIRK généré avec prise en compte d'un missed-cleavage en position 11 ainsi qu'une oxydation de la méthionine en première position (15.9946 Dalton). Comme nous pouvons le constater, ce spectre théorique partage 15 masses en commun avec le spectre expérimental, ce qui est ici suffisant pour la validation statistique de cette identification. La seconde image (b) représente la position initiale des masses du spectre théorique avec missed-cleavage mais sans l'oxydation de la méthionine, c'est-à-dire le spectre théorique construit par SpecOMS. Comme nous pouvons le constater, 9 masses sont partagées entre le spectre expérimental et le spectre théorique, ce qui suffit à SpecXtract pour retenir ce PSM avec la valeur $T = 7$. Ainsi, SpecOMS aboutit à la même identification que celle présentée en (a).

La troisième image (c) est une représentation du spectre théorique MIDMGFEPQIR généré sans la prise en compte du missed-cleavage. Comme nous pouvons le constater, ce spectre théorique partage uniquement 6 masses en commun avec le spectre expérimental, soit trop peu pour que ce PSM soit conservé par SpecXtract. En revanche, si $T < 7$, alors ce PSM serait conservé par SpecXtract. Néanmoins, comme nous l'illustrons sur la dernière image (d), le delta de masse observé entre les deux spectres (la masse de l'oxydation et de l'acide aminé K résultat du missed-cleavage) ne suffit pas à SpecFit pour réaligner les masses. En effet, ce delta de masse est utilisé intégralement sur un unique acide aminé par SpecFit pour réaligner les masses alors que ce delta de masse devrait être scindé et utilisé sur deux acides aminés différents. Sans information supplémentaire sur les origines de ce delta de masse total, il n'est néanmoins pas possible de réaliser cette opération. Ceci explique pourquoi SpecOMS ne parvient pas à recouvrer autant d'identifications lorsque les missed-cleavages ne sont pas pris en compte lors de la digestion enzymatique des protéines.

Le phénomène que nous venons de décrire n'est pas limité à l'oxydation de la méthionine en première position : il peut se produire pour n'importe quelle modification qui affecte un peptide avec un missed-

cleavage. SpecXtract reportera les deux peptides qui correspondent au peptide avec missed-cleavage si leur score est supérieur au threshold, puis SpecFit essaiera d'améliorer le score sans succès, et ces identifications correctes seront très probablement remplacées par des identifications incorrectes qui génèrent un score légèrement meilleur mais souvent insuffisant à leur validation statistique.

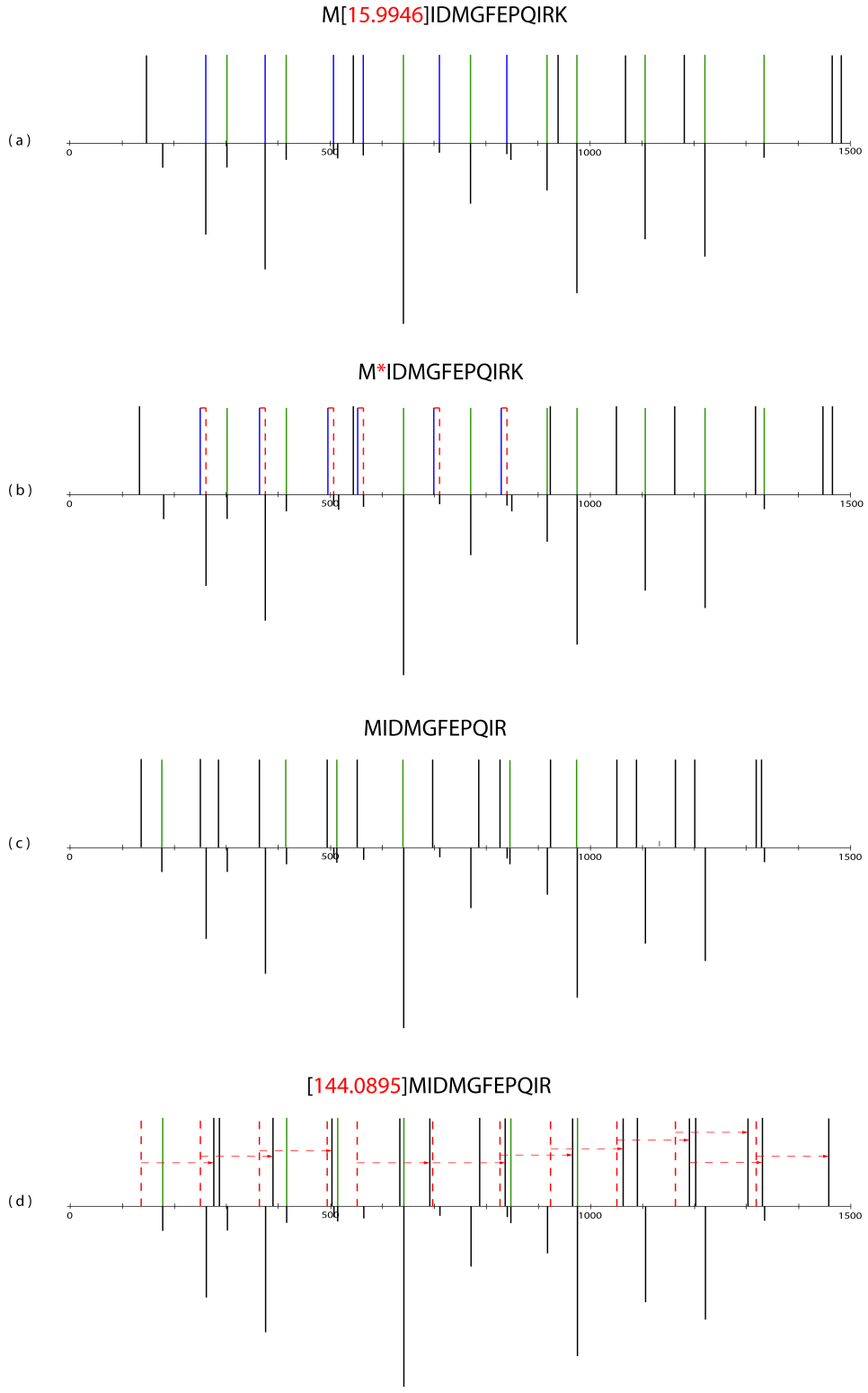


FIGURE 7.2 : Influence de la génération du peptide théorique avec missed-cleavage sur l'identification d'un spectre. Nous avons représenté en bleu les ions b et en vert les ions y lorsque leurs masses sont partagées entre le spectre théorique et le spectre expérimental. Les ions dont les masses ne sont pas partagées sont représentés en noir. A partir d'un PSM, SpecFit peut aligner les masses en utilisant le delta de masse observé entre les deux spectres comme nous l'avons représenté en pointillés rouges.

De plus, lorsqu'un spectre S_e représente un missed-cleavage composé de deux peptides p_1 et p_2 qui comprennent respectivement nb_1 et nb_2 acides aminés, si le missed-cleavage a été identifié "à la volée", alors le nombre de pics du PSM proposé par SpecXtract sera au maximum $max(nb_1/2, nb_2/2)$. Ainsi, la

condition $T = 7$ utilisée par SpecXtract pour extraire les PSMs candidats sélectionne bien plus de PSMs lorsque les spectres théoriques des missed-cleavages sont dans la banque plutôt que lorsqu'ils sont identifiés « à la volée ». Un missed-cleavage composé de deux petits peptides pourra donc difficilement être identifié « à la volée ». Pour les mêmes raisons, le nombre d'identifications obtenu pour le groupe G3' est également plus faible lorsque les missed-cleavages ne sont pas générés lors de la digestion enzymatique.

Pour obtenir la meilleure sensibilité possible lors de l'utilisation de SpecOMS, il est donc largement préférable d'inclure la génération de missed-cleavages lors de la digestion de la banque de protéines. SpecOMS laisse néanmoins à l'utilisateur la possibilité de ne pas générer ces missed-cleavages pour permettre des analyses plus rapides qui diminuent l'empreinte mémoire. Ce gain de temps et d'espace mémoire se fait néanmoins au prix d'un nombre d'identifications perdues conséquent.

7.2.3 Choix des paramètres de précision α et β

Les paramètres α et β présentés dans le chapitre précédent permettent de configurer la précision de la mesure de masse des ions fragments générés par le spectromètre de masse. Nous nous attendons à ce que les résultats de SpecOMS (et des approches OMS plus généralement) se dégradent lorsque la précision sur la mesure des ions fragments est moins bonne. En effet, plus cette précision est faible, plus le risque est élevé que la masse d'un ion fragment corresponde par hasard à une masse qui caractérise en réalité un ion autre que celui-ci. Pour évaluer l'influence de ces paramètres α et β sur la sensibilité de SpecOMS, nous avons comparé les résultats de l'analyse en configuration standard avec les résultats des analyses pour lesquelles nous avons fait varier le paramètre β sur une échelle de 1 à 9. Cette variation nous permet de simuler une évolution de la précision de la mesure sur les ions fragments de 0.01 Da à 0.09 Da. Les résultats obtenus sont présentés dans la Table 7.3, où nous reportons également le score S_{FDR} pour lequel la FDR < 1% est atteinte, la valeur exacte de la FDR associée, le nombre d'identifications pour chacun des trois groupes G1, G2 et G3', ainsi que le nombre d'identifications au total. La ligne mise en valeur en italique correspond aux résultats de l'analyse en configuration standard.

	groupe G1			groupe G2			groupe G3'			total
Précision (Dalton)	S_{FDR}	FDR en %	nombre d'identifications	S_{FDR}	FDR en %	nombre d'identifications	S_{FDR}	FDR en %	nombre d'identifications	nombre d'identifications
0.01	7	0.30	12 536	11	0.47	3 601	13	0.50	597	16 734
0.02	*7*	*0.87*	*12 694*	*11*	*0.73*	*3 821*	*13*	*0.96*	*625*	*17 140*
0.03	9	0.71	11 203	12	0.56	2 877	14	0.95	423	14 503
0.04	10	0.41	9 519	13	0.33	2 091	15	0.00	288	11 898
0.05	10	0.60	9 612	13	0.35	2 066	15	0.00	280	11 918
0.06	10	0.84	9 699	13	0.46	1 977	15	0.36	274	11 950
0.07	11	0.31	8 008	13	0.75	1 872	16	0.00	180	10 060
0.08	11	0.37	8 052	14	0.40	1 246	15	0.79	254	9 552
0.09	11	0.43	8 097	14	0.42	1 199	16	0.59	170	9 466

TABLE 7.3 : Influence de la précision sur la mesure des fragments sur la sensibilité de SpecOMS.

Les résultats présentés mettent en évidence dans un premier temps une augmentation du nombre d'identifications tant que la valeur de paramètre β n'a pas atteint la précision de l'appareil (ici 0.02 Da). Nous observons ensuite une diminution globale du nombre d'identifications dans les trois groupes G1, G2 et G3' quand la précision de la mesure sur les ions fragments évolue de 0.02 Da à 0.09 Da. Cette diminution est cohérente avec l'hypothèse avancée : lorsque la précision sur les fragments diminue, le risque d'interprétation erronée des masses augmente, ce qui se traduit par une augmentation du nombre de faux positifs, et donc une diminution de la sensibilité de SpecOMS. En conclusion, les paramètres α et β de SpecOMS doivent être configurés de sorte à représenter la précision effective du spectromètre de masse lors de l'acquisition des données pour maximiser le nombre d'identifications.

Nous observons également que malgré des seuils identiques de S_{FDR} dans le groupe G2, le nombre d'identifications diminue de manière significative avec la perte de précision. Ainsi, lorsque la précision évolue de 0.04 Da à 0.07 Da, le nombre d'identifications passe de 2091 à 1872. Pourtant, le score de FDR est resté le même (S_{FDR}=13). La modification de la précision de mesure sur les fragments permet d'augmenter le nombre de PSMs candidats pour la plupart des spectres, et peut introduire des PSMs de meilleur score classés dans un autre groupe (G1 ou G3'). Malheureusement, les conditions de validation des PSMs dans le groupe G3' sont actuellement très pénalisantes (S_{FDR} plus élevé, seuil de Δ_m > -400 Da) et induisent des pertes d'identifications.

Un point également à noter dans la Table 7.3 est la conséquence de l'évolution du score S_{FDR}. Le score de SpecOMS est calculé sous la forme d'une valeur discrète et non d'une plage de valeurs continues. De ce fait, d'une valeur à l'autre, on peut observer une variation importante du nombre d'identifications dans la banque target et dans la banque decoy, lesquelles sont regroupées autour de valeurs discrétisées. Il en résulte une évolution par paliers du nombre d'identifications. Nous pouvons raisonnablement supposer que des identifications actuellement ignorées dont le score est immédiatement inférieur à la valeur de S_{FDR} auraient pu être conservées avec un calcul sur une plage de valeur continue. En effet, pour certaines valeurs de S_{FDR}, la FDR associée est nettement inférieure à 1%. Nous discutons en conclusion de ce manuscrit d'une manière de rendre continu le score calculé par SpecOMS afin de faciliter l'interprétation de la FDR et éventuellement d'améliorer ainsi le nombre d'identifications obtenues.

7.2.4 Favoriser l'identification des peptides non modifiés

En réalisant le ré-alignement des spectres des PSMs, SpecFit favorise l'identification des spectres modifiés. Cependant, cette option peut dans le même temps dégrader le nombre total d'identifications, avec une diminution du nombre de spectres non modifiés identifiés, comme nous l'avons déjà illustré dans la Section 7.2.1. Pour éviter une diminution de la sensibilité de SpecOMS causée par ce phénomène, nous avons introduit un mécanisme permettant de privilégier les identifications pour lesquelles $\Delta(s_i, s_j)$ est nul (c.f. Section 5.6, page 73), favorisant ainsi la cohérence entre la masse de l'ion précurseur et la masse du spectre expérimental. Pour évaluer l'influence de notre solution sur la sensibilité de SpecOMS, nous avons comparé l'analyse en configuration standard avec des analyses pour lesquelles nous avons fait varier la valeur du paramètre $S_{\Delta m=0}$, lequel contrôle à quel seuil de score SpecOMS commence à privilégier les identifications pour lesquelles $\Delta(s_i, s_j)$=0. Nous avons reporté dans la Table 7.4 le nombre d'identifications obtenues pour les trois groupes G1, G2 et G3' les valeurs de S_{FDR} ainsi que la FDR associée. La ligne correspondant à la configuration standard est mise en valeur en italique.

	groupe G1			groupe G2			groupe G3'			total
$S_{\Delta m=0}$	S_{FDR}	FDR en %	nombre d'identifications	S_{FDR}	FDR en %	nombre d'identifications	S_{FDR}	FDR en %	nombre d'identifications	nombre d'identifications
7	8	0.85	12 541	11	0.68	3 688	13	0.69	580	16 809
8	*7*	*0.87*	*12 694*	*11*	*0.73*	*3 821*	*13*	*0.96*	*625*	*17 140*
9	7	0.29	11 395	11	0.87	3 887	14	0.00	431	15 713
10	7	0.13	10 382	12	0.48	2 930	14	0.00	437	13 749
11	7	0.09	9 708	12	0.64	2 967	14	0.23	441	13 116
12	7	0.08	9 260	12	0.70	3 013	14	0.23	446	12 719
13	7	0.08	8 923	12	0.76	3 040	14	0.21	474	12 437
14	7	0.08	8 730	12	0.75	3 054	14	0.54	552	12 336
15	7	0.08	8 577	12	0.75	3 059	14	0.47	634	12 270

TABLE 7.4 : Influence du paramètre $S_{\Delta m=0}$ sur la sensibilité de SpecOMS.

Nous observons tout d'abord avec l'augmentation de $S_{\Delta m=0}$ une nette diminution du nombre d'identifications dans le groupe G1, allant de 12 541 identifications pour une valeur de $S_{\Delta m=0}$ de 7 à 8 577 identifications pour une valeur de $S_{\Delta m=0}$ de 15. Parce que SpecOMS favorise de moins en moins les interprétations pour lesquelles $\Delta(s_i, s_j)$=0 avec une valeur croissante du paramètre $S_{\Delta m=0}$, ces interprétations

du groupe G1 sont remplacées par des interprétations dans le groupe G2 ou le groupe G3' avec un score plus élevé. Néanmoins, la même augmentation du paramètre $S_{\Delta m=0}$ de 7 à 15 ne se traduit pas par une augmentation notable du nombre d'identifications dans les groupes G2 et G3'. En effet ce nombre d'identifications oscille entre 3 821 et 2 930 pour le groupe G2, tandis qu'il oscille entre 431 et 634 pour le groupe G3'. Ces nombres d'identification sont loin de compenser la perte d'identifications dans le groupe G1 (jusqu'à 3 964 identifications perdues). Nous observons en réalité ici le même phénomène que précédemment : une grande partie des interprétations perdues du groupe G1 étaient des identifications correctes et ces identifications perdues sont majoritairement remplacées par un nombre conséquent d'interprétations dues au hasard (donc dans la banque decoy) dans les groupes G2 et G3'. Par conséquent, ces interprétations se trouvent en dessous du seuil de validation statistique des identifications par FDR. Ainsi, il est préférable de conserver le paramètre de S_{FDR} petit : lorsqu'un PSM présente un delta de masse nul, il est rare qu'il existe une meilleure identification avec un delta de masse non nul qui ne soit pas un faux positif. On notera néanmoins que cette valeur S_{FDR} est bornée par le bas par la valeur du threshold T utilisé par SpecXtract.

7.2.5 Impact et choix du threshold T

Lors du calcul des valeurs de similarité, SpecXtract utilise le paramètre de threshold T pour minimiser le nombre de paires de spectres à manipuler en ignorant les paires dont la similarité est inférieure à ce seuil (c.f. Section 5.5.2). Nous avons également souhaité évaluer l'influence de cette restriction sur la sensibilité de SpecOMS. Pour ce faire, nous avons comparé les résultats obtenus lors d'une analyse en configuration standard avec les résultats d'analyses lors desquelles nous avons fait varier la valeur du paramètre de threshold T. Les résultats obtenus sont présentés dans la Table 7.5, où sont reportés le nombre d'identifications obtenues pour les trois groupes, le score S_{FDR} pour lequel la FDR < 1% est respectée, ainsi que la valeur exacte de la FDR associée. La ligne mise en valeur en italique correspond aux résultats de l'analyse en configuration standard.

	groupe G1			groupe G2			groupe G3'			total
Threshold	S_{FDR}	FDR en %	nombre d'identifications	S_{FDR}	FDR en %	nombre d'identifications	S_{FDR}	FDR en %	nombre d'identifications	nombre d'identifications
4	4	0.84	12 641	11	0.76	4 184	14	0.23	431	17 256
5	5	0.85	12 628	11	0.80	4 113	14	0.23	429	17 170
6	6	0.85	12 639	11	0.85	4 007	14	0.23	428	17 074
7	*7*	*0.87*	*12 694*	*11*	*0.73*	*3 821*	*13*	*0.96*	*625*	*17 140*
8	8	0.85	12 541	11	0.45	3 570	13	0.83	605	16 716
9	9	0.24	10 861	11	0.63	3 338	13	0.36	563	14 762
10	10	0.65	9 146	12	0.26	2 300	12	0.43	690	12 136
11	11	0.03	7 505	12	0.59	2 034	12	0.69	852	10 121
12	12	0.00	5 939	12	0.85	1 759	13	0.00	332	8 030

TABLE 7.5 : Influence du paramètre T sur la sensibilité de SpecOMS.

Nous pouvons observer que le nombre d'identifications atteintes par SpecOMS dans le groupe G1 stagne au fur et à mesure que la valeur du threshold T augmente jusqu'à $T = 7$. Lorsque le paramètre T est bas, SpecXtract a plus de chance de retenir, pour un spectre expérimental donné, un ou plusieurs PSMs supplémentaires qui ont un delta de masse non nul. Ces PSMs sont ensuite analysés par SpecFit pour essayer d'améliorer le score et peuvent aboutir à un score plus élevé que le PSM avec un delta de masse nul, mais plus faible que le score $S_{\Delta m=0}$. L'un de ces PSMs peut donc être retenu, ce qui explique le faible gain d'identifications quand T < 7. Une fois que la valeur du paramètre T atteint, puis dépasse celle de $S_{\Delta m=0}$ (c'est-à-dire de T > 7), on observe une très forte diminution du nombre d'identifications atteint par SpecOMS. En effet, dans ce cas l'algorithme SpecXtract reporte dans les résultats de moins en moins de paires de spectres, puisqu'une grande partie des paires lues dans SpecTrees possèdent moins de masses partagées que la valeur de T. Les mêmes tendances s'observent pour les identifications du groupe 2, et ce pour les mêmes raisons. Le groupe G3' présente une stagnation du nombre d'identifications obtenues pour

des valeurs de T de 4 à 6. Nous observons une augmentation du nombre d'identifications pour une valeur de T de 7, qui s'accompagne d'un saut de pallier de S_{FDR}. Comme nous l'avons expliqué précédemment, de tels sauts, dûs à la nature discrète du score de SpecOMS, compliquent l'interprétation des évolutions du nombre d'identifications obtenues. Pour des valeurs de T supérieures à 7, le nombre d'identifications obtenues dans le groupe G3' semble augmenter légèrement (en prenant en compte les variations dues aux sauts des paliers de S_{FDR}). Cette augmentation peut s'expliquer par une diminution du nombre de faux positifs obtenus dans ce groupe. Avec l'augmentation de T, les paires considérées partagent plus de masses en commun et le risque de les identifier par hasard diminue. Contrairement aux groupes G1 et G2, le groupe G3' est, comme nous l'avons présenté au début, bien plus sujet aux identifications au hasard : il bénéficie donc bien plus d'une diminution de leur quantité.

Il est souhaitable de choisir une valeur de threshold basse pour maximiser la sensibilité de SpecOMS. Néanmoins, en dessous d'un certain seuil (ici 7), le nombre d'identifications supplémentaires obtenues est relativement faible, et l'algorithme se trouve d'autant plus ralenti que le threshold est faible en raison du très grand nombre de paires de spectres à considérer qui partagent une faible similarité.

7.3 Les cas d'usage de SpecOMS

Compte tenu des tests que nous avons réalisés dans le paragraphe précédent, nous constatons que nous pouvons adapter l'usage de SpecOMS à différents contextes d'utilisation en favorisant sa rapidité, sa sensibilité ou la recherche de modifications fréquentes ou rares. Dans un premier temps, nous nous intéresserons à l'usage de SpecOMS pour identifier rapidement les principales modifications portées par les spectres d'un échantillon, ce qui peut être très utile pour contrôler les artefacts d'un protocole par exemple.

7.3.1 Identification rapide des modifications fréquentes d'un échantillon

Avec une identification des missed cleavages « à la volée » et un paramètre T > 8, SpecOMS peut s'exécuter avec un usage limité de ressources mémoire et de CPU (cf Table 7.2). Bien sûr, la sensibilité de la méthode sera faible, mais restera néanmoins suffisante pour identifier les principales modifications. SpecOMS pourra ainsi être ajouté facilement à des pipelines d'analyse pour contrôler la qualité des échantillons.

Pour faciliter une interprétation rapide, depuis la version 1.2.0, SpecOMS fournit, au travers de son interface graphique, une représentation visuelle des différences de masses entre le spectre expérimental et le spectre théorique de chaque identification. La Figure 7.3 montre cette représentation graphique pour le jeu de données HEK293. Une base de données des modifications telle Unimod peut être utilisée pour trouver une explication probable aux différents delta de masse observés.

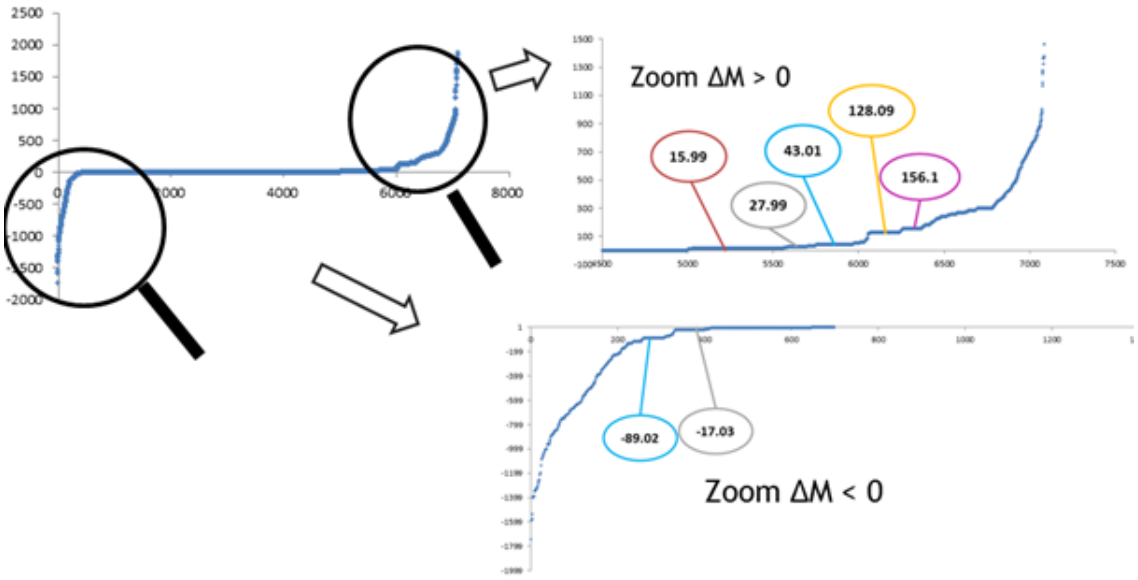


FIGURE 7.3 : Mise en évidence, par représentation graphique, des modifications fréquentes identifiées dans le jeu de données HEK293. Nous y voyons à gauche l'allure générale de la courbe des deltas de masse, sur laquelle les PSMs sont triés par deltas de masse croissants et représentés sur l'axe des abscisses, l'axe des ordonnées reportant ces deltas de masse en Daltons. Nous avons représenté à droite des agrandissements des zones avec les deltas de masse non nuls. Nous y avons annoté les deltas de masse que l'on retrouve fréquemment dans le jeu de données HEK293 : oxydation (15.99 Da), formylation (27.99 Da), carbamylation (43.01 Da), addition d'une lysine (128.09 Da), addition d'une arginine (156.10 Da), suppression de la méthionine C-terminale et acétylation (-89.02 Da), perte d'ammoniac (-17.03 Da)).

Cette représentation permet également de mettre en valeur des plages de delta de masses similaires pour des groupes de spectres, qui sont bien plus facilement identifiables par l'utilisateur à l'œil nu que par analyse manuelle des données. Ces groupes de spectres qui présentent des deltas de masses similaires permettent d'identifier rapidement les modifications les plus fréquentes dans le jeu de données analysé ou bien les artefacts de l'expérience. Les utilisateurs du logiciel peuvent inclure eux-mêmes une liste de modifications qu'ils souhaitent mettre en évidence et créer différents filtres sur les résultats reportés à l'issue de l'analyse.

On identifie également sur cette représentation les plages de deltas de masses correspondant aux erreurs isotopiques, décalées de 1 Dalton par rapport aux plages qui correspondent aux modifications courantes.

Si l'échantillonnage rapide de gros volumes de données expérimentales produites par spectrométrie de masse en tandem a longtemps été considéré comme difficilement exploitable à la fois en raison de complexité temporelle et de sensibilité de l'approche [3, 103], l'exécution possible de l'analyse précédente en quelques minutes illustre ici que ce n'est plus le cas aujourd'hui. Cette rapidité et simplicité d'échantillonnage permet de mettre en valeur des modifications qui sont des conséquences de la préparation d'échantillons en l'absence de protocoles d'enrichissement. SpecOMS est donc un excellent outil pour contrôler les artefacts expérimentaux. Ces artefacts peuvent générer la perte d'un nombre d'identifications important ou même conduire à des identifications erronées et être la cause d'erreurs lors de l'inférence de protéines. Beaucoup de ces artefacts sont connus, mais la difficulté à prédire leur présence d'un échantillon à l'autre justifie l'intérêt de cette approche.

7.3.2 Extension de la couverture des identifications

Nous avons effectué l'analyse de 37 703 spectres expérimentaux issus du jeu de données HEK293 en les comparant avec 2 701 875 spectres théoriques issus de la digestion enzymatique de 87 049 protéines dans différentes configurations. Pour chaque spectre expérimental, SpecOMS a sélectionné le peptide identifiant le plus probablement ce spectre et a reporté une possible modification ainsi que sa localisation et le delta de masse associé. Nous avons ensuite effectué une validation statistique des résultats avec une FDR < 1%.

Sans utiliser SpecFit pour rechercher des identifications supplémentaires en améliorant le score des PSMs, SpecOMS permet l'identification de 14 696 spectres expérimentaux. Lorsque SpecFit est utilisé, ce sont 1 194 spectres supplémentaires qui sont identifiés et validés statistiquement, soit 7% des 17 113 identifications atteintes par SpecOMS. Ce nombre d'identifications varie également selon si l'on utilise uniquement SpecFit pour rechercher les missed-cleavages, ou si ces derniers sont pris en compte lors de la digestion enzymatique des protéines. En effet, lorsque seul SpecFit est utilisé pour la recherche des missed-cleavages, 15 929 spectres sont identifiés. On observe un gain de 2 417 spectres lorsque ces missed-cleavages sont pris en compte lors de la digestion enzymatique, soit 14% du nombre total d'identifications. Ce gain s'explique par une meilleure exploitation du delta de masse des PSMs par SpecOMS.

SpecOMS nécessite l'utilisation d'une précision de mesure des fragments élevée, sans quoi la proportion d'identifications dues au hasard est très importante. Les résultats présentés montrent cependant que même avec une fonction de score simple, en mettant à profit la précision accrue des nouvelles générations de spectromètres de masse, SpecOMS est capable d'atteindre un nombre d'identifications important. En particulier, pour obtenir la meilleure sensibilité possible, les paramètres de précision de SpecOMS doivent être configurés de manière à s'approcher le plus possible de la précision de la mesure des ions fragments effectuée par le spectromètre de masse lors de l'acquisition.

Enfin, pour obtenir la meilleure sensibilité possible, SpecOMS favorise les PSMs dont les deltas de masse sont nuls à l'aide du paramètre de score $S_{\Delta m=0}$. Une décomposition en différents groupes pour le calcul de la FDR permet de maximiser le nombre d'identifications validées pour chaque groupe. En effet, la FDR globale de l'échantillon n'est pas homogène, et comme nous l'avons constaté, ce n'est pas la même valeur de score qui suffit pour la validation statistique de PSMs avec un delta nul, négatif ou positif. Nous observons néanmoins d'importantes difficultés pour l'identification de PSMs qui exhibent des deltas de masse négatifs.

En définitive, SpecOMS permet l'identification de 17 113 spectres expérimentaux, soit 45% du nombre total de spectres dans le jeu de données HEK293. Les méthodes OMS sont aujourd'hui souvent critiquées pour leur manque de sensibilité, mais nous montrons ici que cette critique n'est pas fondée. En effet, sur ce même jeu de données HEK293, XTandem ! identifie 18 275 spectres expérimentaux, soit une différence presque négligeable entre les deux logiciels. En implémentant une fonction de score plus élaborée et en résolvant le problème des identifications avec les deltas de masse négatifs, nous nous attendons donc à observer une sensibilité meilleure que celle actuellement atteinte par les méthodes basées sur les filtres de masse.

7.3.3 Recherche de modifications rares

Dans le cas des approches restreintes, l'identification de modifications rares pose beaucoup de difficultés. En effet, la sélection des peptides de masse proche de celle du spectre expérimental par le biais d'un filtre de masse exclut les modifications de la recherche, à moins que celles-ci ne soient explicitement recherchées. Malheureusement, on observe une forte dégradation des performances lorsque trop de modifications spécifiques sont recherchées simultanément, due à une augmentation conséquente de l'espace de recherche. De plus, il n'est pas possible de rechercher ainsi les modifications rares voire inconnues sans avoir connaissance

a priori de leur présence dans le mélange analysé. Les approches OMS, et plus particulièrement SpecOMS, sont capables de mettre en évidence de telles modifications rares grâce à l'absence de filtres préliminaires sur les peptides, et donc la mise en compétition de tous les peptides pour l'analyse de chaque spectre expérimental. Nous illustrons ici quelques cas intéressants de modifications rares que SpecOMS a été en mesure d'identifier dans le jeu de données HEK293.

La Figure 7.4 présente une large variété de deltas de masse présents parmi les identifications effectués par SpecOMS lors de l'analyse du jeu de données HEK293. La taille des bulles est proportionnelle à la fréquence de détection du delta de masse associé, dont la valeur est indiquée dans la bulle correspondante pour les plus fréquentes. Les bulles de grande taille représentent pour la plupart des identifications exhibant des modifications courantes connues mais plusieurs deltas de masse ne sont représentés que par quelques spectres.

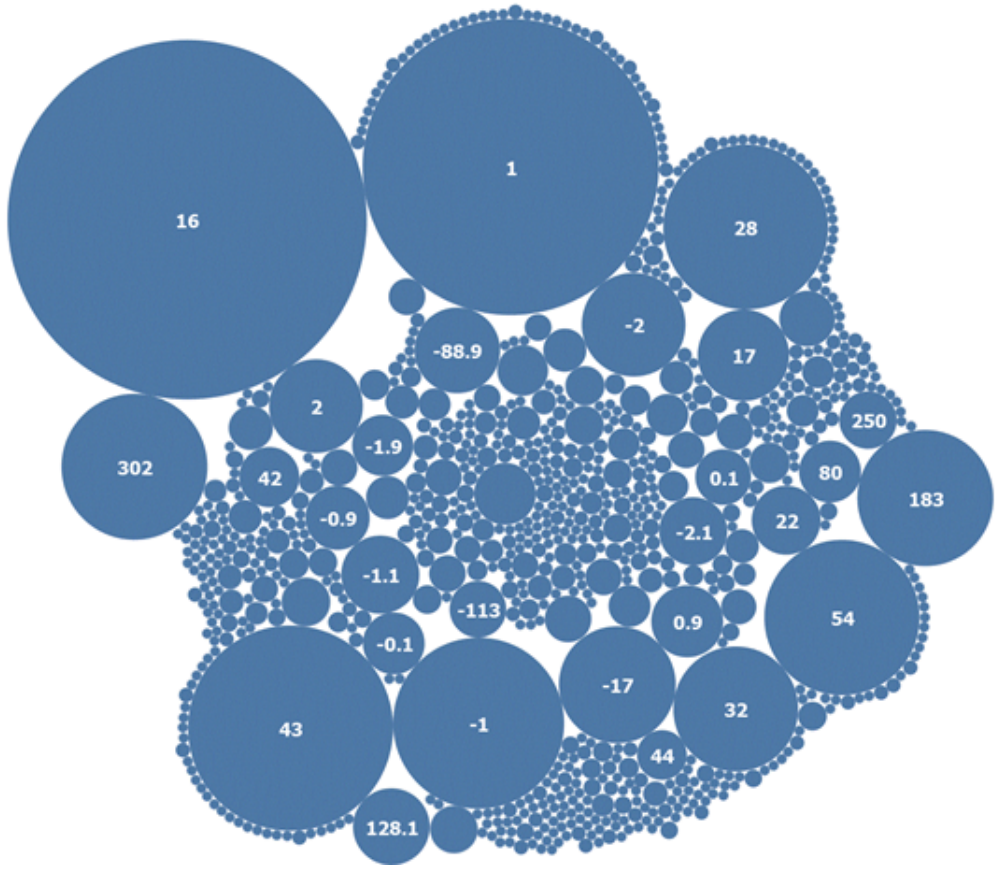


FIGURE 7.4 : Représentation en bulles des deltas de masses observés par SpecOMS dans le dataset A de HEK293 avec une tolérance de 0.05 Da.

La Figure 7.5 illustre la présence d'un missed-cleavage non anticipé en première position. Ce spectre (n° 12 186) est en effet identifié par le peptide KGISEDTTEETIK avec un delta de masse qui correspond à l'acide aminé K. Cet acide aminé K a été localisé par SpecFit en première position dans le peptide et permet d'augmenter significativement le nombre de masses partagées (des astérisques désignent les masses recouvrées lors de la recherche effectuée par l'algorithme SpecFit). Ce type de missed-cleavage ne peut être détecté lors de la génération des missed-cleavages pendant la digestion enzymatique des protéines car le peptide précédant GISEDTTEETIK dans la séquence de la protéine n'est pas un composé d'un unique acide aminé K mais est une séquence de plusieurs acides aminés qui se termine par K.

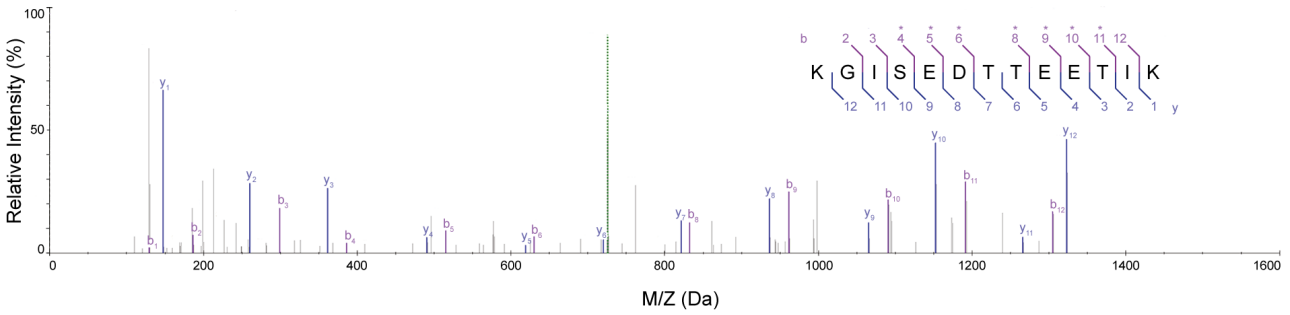


FIGURE 7.5 : Spectre (id : 12186) identifié à l'aide de SpecOMS exhibant un site de missed-cleavage non anticipé.

La figure 7.6 illustre la présence d'une modification de masse très importante sur un peptide : une glycosylation. Ce spectre (n° 34 487) est en effet identifié par le peptide FGVSSSSSGPSQTITSTGNFK avec un delta de masse de 203.0805 Daltons, lequel correspond à une N-acetylglucosamine. Cette identification est renforcée par la présence de plusieurs ions marqueurs (c.f. Section 3.2.4, page 40) caractéristiques de cette modification (168.0660 Da., 186.0766 Da. et 204.0872 Da.). L'importance du delta de masse de ce PSM (> 200 Da) le rend inaccessible aux algorithmes de recherche restreinte, mais les méthodes OMS et en particulier SpecOMS sont capables d'identifier de tels peptides.

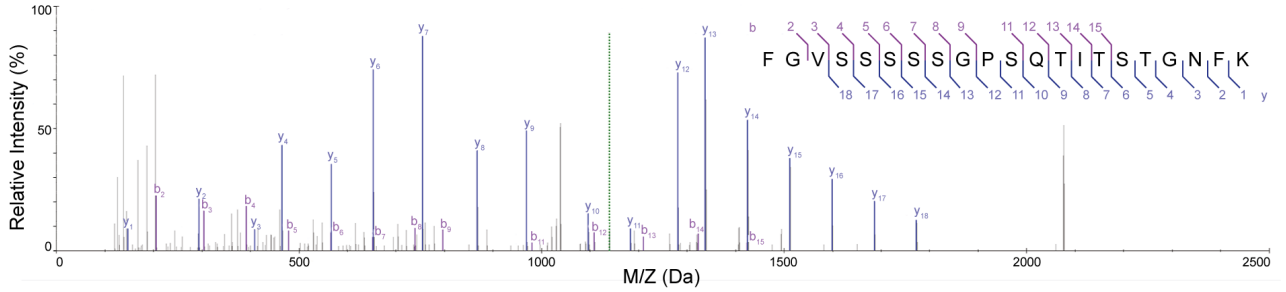


FIGURE 7.6 : Spectre (id : 34487) identifié à l'aide de SpecOMS comme un peptide glycosylé.

La Figure 7.7 illustre la mauvaise attribution de charge au spectre expérimental par le logiciel de prétraitement des spectres de masse. Ce specre (n° 38 793) est en effet identifié par le peptide HDAATCFVDAGNAFK avec un delta de masse de 811.3735 Daltons et un grand nombre de masses partagées (18). La charge attribuée à ce spectre par le logiciel RawConverter est 3, alors qu'une interprétation du spectre proposée par SpecFit avec 2 charges explique très exactement le delta de masse. SpecOMS reporte cette identification qui est ensuite statistiquement validée malgré un important delta de masse au départ, lequel rend cette identification inaccessible aux méthodes de recherche restreinte malgré un très bon score atteint pour ce PSM.

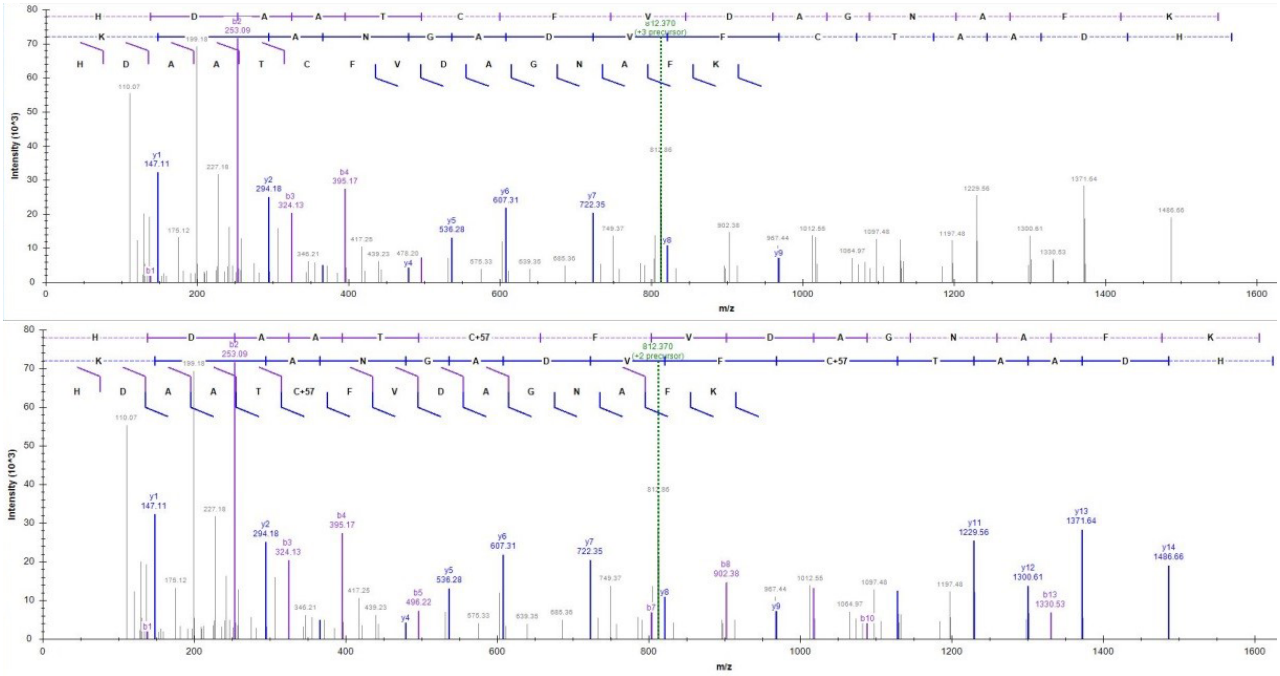


FIGURE 7.7 : Spectre (id : 38793) identifié à l'aide de SpecOMS pour lequel l'attribution de charge était erronée.

En outre, SpecOMS met à profit une stratification de l'espace de recherche, qui permet d'imposer des seuils de validation différents selon si les identifications portent ou non sur des peptides modifiés. Ainsi, cette approche permet de mettre en compétition toutes les PTMs lors du processus d'interprétation des spectres expérimentaux sans pour autant introduire de biais vers certaines PTMs en introduisant un nombre de peptides candidats trop élevé. Une étude récente [64] a par exemple démontré que la FDR associée avec un sous-ensemble de peptides méthylés pouvait être dégradée jusqu'à 80% alors que la FDR globale de l'analyse était environ de 1%. L'utilisation de SpecOMS avec des seuils de FDR différents contribue à une FDR faible même pour des sous-ensemble de peptides modifiés et par conséquent SpecOMS est un algorithme approprié pour rechercher les modifications rares.

7.4 Comparaison avec MSFragger

Afin de positionner SpecOMS par rapport aux méthodes OMS existantes et de situer ses performances, nous avons choisi de comparer les valeurs obtenues précédemment avec les performances de MSFragger [86]. MSFragger est un outil développé par le groupe de Nesvizhskii, publié dans la revue Nature Methods en mai 2017. Les deux logiciels ont été développés pendant la même période et ont beaucoup amélioré la vitesse d'exécution des approches OMS par comparaison à une banque de protéines (c.f. [86], Supplementary Table 1). MSFragger est qualifié par ses concepteurs de méthode OMS ultra rapide.

7.4.1 Principes de l'algorithme MSFragger

Préparation des spectres Dans un premier temps, MSFragger lit les données des spectres expérimentaux à partir d'un fichier MGF, mzML ou mzXML. Les intensités des pics sont normalisées puis stockées sous la forme d'entiers pour une manipulation efficace. Pour chaque spectre, les k pics les plus intenses sont sélectionnés, avec k un paramètre fixé par l'utilisateur. Ces pics sélectionnés sont ensuite filtrés pour en exclure les pics inférieurs à une certaine intensité ou hors d'une certaine gamme de masse. A nouveau, ces paramètres sont indiqués par l'utilisateur.

Préparation des peptides MSFragger concatène les protéines en une unique séquence ensuite découpée en blocs pour simuler en parallèle une digestion enzymatique. Les peptides sont représentés de manière efficace sous la forme d'une position et d'une longueur dans la séquence d'une protéine. Les séquences de peptides obtenues par digestion enzymatique sont triées, et les peptides dupliqués marqués comme tels par comparaison aux peptides voisins. MSFragger permet de réaliser trois types de digestion au choix : enzymatique (les deux extrémités du peptide correspondent à une digestion par l'enzyme choisie), semi-enzymatique (seule une extrémité correspond) ou non enzymatique (aucune des deux extrémités n'est restreinte). Lors de cette digestion, MSFragger ne conserve que les peptides qui rentrent dans les limites des paramètres de nombre de missed-cleavages, longueur de peptide ou de masse du peptide spécifiés par l'utilisateur. A partir de ces peptides, MSFragger va construire un index des peptides, nécessaire pour réaliser ensuite la construction d'un index des fragments, dans lequel MSFragger effectuera les recherches de masses partagées entre spectre expérimental et spectre théorique.

Construction de l'index des peptides A partir des peptides générés par digestion enzymatique, MSFragger construit des peptides modifiés en utilisant les modifications éventuellement renseignées par l'utilisateur. Chaque modification est représentée par un masque, et chaque combinaison de modification par un entier résultant de ce masque. Pour chaque peptide, MSFragger stocke la longueur de la séquence de modifications, sa position, et la masse modifiée. Les peptides sont ensuite triés en parallèle en utilisant leur masse modifiée et forment ainsi l'index de peptides.

Construction de l'index des fragments Pour la génération des ions fragments, MSFragger construit tous les fragments qui correspondent aux ions b et aux ions y pour les valeurs de charge données par l'utilisateur. Chaque fragment est associé à l'enregistrement de peptide à partir duquel il a été produit. Les fragments sont ensuite répartis dans des ensembles, chaque ensemble correspondant à un intervalle de masse. Dans chaque intervalle, les ions fragments sont triés par ordre de masse croissante de leur ion parent. Pour optimiser le temps de recherche de MSFragger, la largeur des intervalles doit être proportionnelle à la précision sur la mesure des fragments indiquée par l'utilisateur et au nombre de candidats attendus en moyenne pour les spectres expérimentaux. MSFragger calcule lui-même cette largeur d'intervalle.

Recherche dans l'index des fragments Dans un premier temps, MSFragger identifie le nombre de peptides candidats pour chaque spectre expérimental en utilisant la fenêtre de masse de l'ion précurseur des spectres expérimentaux spécifiée par l'utilisateur. Ensuite, MSFragger alloue pour chaque candidat une table de score pour stocker le nombre et l'intensité des ions b et des ions y partagés entre le spectre expérimental et ce candidat. Finalement, MSFragger calcule pour chaque candidat le score obtenu entre celui-ci et le spectre expérimental de la manière qui suit. Pour chaque fragment de masse m dans un spectre expérimental, MSFragger récupère dans l'index des fragments tous les fragments qui se trouvent dans le ou les ensemble couverts par l'intervalle $[m - p, mp]$ avec p la précision fournie par l'utilisateur. Un parcours de ce ou ces ensembles permet de comparer les fragments contenus pour vérifier qu'ils correspondent effectivement à l'ion fragment issu du spectre expérimental, c'est-à-dire qu'ils tombent bien dans la fenêtre de précision autour de la masse de ce fragment et que leur état de charge est le même. Si c'est le cas, les scores de l'ion parent sont incrémentés dans la table réservée préalablement à cet effet. Ce procédé, répété pour tous les spectres expérimentaux, permet d'obtenir le nombre de masses partagées et les intensités des ions fragments associés et de les utiliser pour générer un score de similarité.

Calcul du score d'un PSM Pour calculer le score de chaque PSM, MSFragger utilise une fonction de score très similaire à celle de X !Tandem.

$$score = \log\left(N_b! N_y! \sum_{i=1}^{N_b} I_{b,i} \sum_{j=1}^{N_y} I_{y,j}\right)$$

N_b représente le nombre d'ions b qui sont partagés, N_y le nombre d'ions y partagés, I_b et I_y leurs intensités respectives associées. En plus de prendre en compte l'intensité, cette fonction de score permet notamment de donner plus de poids aux ions en fonction du nombre d'ions de ce type qui sont partagés entre le spectre expérimental et le spectre théorique. Une régression linéaire permet ensuite de calculer l'espérance associée à ce PSM. Finalement, MSFragger conserve un nombre de PSMs fixé par l'utilisateur, par ordre d'espérance décroissante.

7.4.2 Éléments de comparaison

Nous avons analysé le jeu de données HEK293 avec le logiciel MSFragger (version package 11062017) pour comparer les performances des deux logiciels dans une configuration où l'on souhaite maximiser le nombre d'identifications de peptides. Les paramètres de MSFragger décrits ci-après ont été choisis de manière à obtenir des conditions d'analyse comparables avec celles utilisées par SpecOMS. Il n'est néanmoins pas toujours possible de trouver des correspondances parfaites entre les paramètres des deux logiciels. Nous avons donc, lorsque nous le jugions pertinent, conservé les configurations par défaut des paramètres recommandés par les concepteurs de MSFragger pour ne pas pénaliser ce dernier lors de la comparaison des résultats. En résumé, la digestion enzymatique des protéines a ainsi été configurée pour l'usage de la trypsine avec un unique missed-cleavage autorisé. Les peptides ont été filtrés pour conserver uniquement les séquences qui comprenaient entre 7 et 25 acides aminés et dont la masse est inférieure à 7 000 Daltons. Cette fourchette de masse fait partie des paramètres par défaut du logiciel, mais couvre l'ensemble des masses de peptides non modifiés qui peuvent être produits à partir de 25 acides aminés. Les spectres théoriques ont été construits à partir de ces peptides en ne considérant que les ions b et y des ions précurseurs une fois chargés. Seuls les spectres expérimentaux qui contiennent plus de 10 pics (paramètre par défaut recommandé) ont été conservés, et seuls les 100 pics les plus intenses de chaque spectre ont été conservés (paramètre par défaut recommandé). Nous n'avons exclu aucune masse (ni gamme de masse), ce qui là aussi représente les paramètres par défaut de MSFragger. Aucune modification additionnelle n'a été recherchée explicitement autre que la modification systématique de 57.021 Da sur les cystéines, aussi utilisée pour SpecOMS.

Lors de l'analyse, nous avons configuré MSFragger pour n'utiliser qu'un seul cœur du processeur afin de lui donner accès à une puissance de calcul similaire à celle utilisée par SpecOMS. En effet, la version actuelle de SpecOMS ne supporte pas d'exécution parallèle. Nous avons autorisé MSFragger à effectuer une recherche non restreinte avec une fourchette de tolérance de masse sur l'ion parent entre -3 000 et 3 000 Da, ce qui élimine peu de couples spectre-peptide. La précision sur la mesure des ions fragments a été fixée à 0.02 Da. Nous avons configuré MSFragger pour rechercher plusieurs modifications variables par peptide et jusqu'à deux modifications affectant le même peptide (paramètres recommandés par défaut). Pour chaque spectre expérimental, seule la PSM qui fournit le meilleur score a été sélectionnée et les autres ignorées. Puisque MSFragger ne dispose pas d'option pour construire la banque decoy, nous avons généré les protéines decoy comme le fait SpecOMS.

Dans les conditions que nous venons de décrire, l'analyse du jeu de données HEK293 par MSFragger a duré 60 minutes et a occupé 5.4 Go de mémoire vive. La même tâche exécutée par SpecOMS, a duré 48 minutes et utilisé 8.8 Go de mémoire vive. Nous constatons donc un temps d'exécution légèrement plus faible pour SpecOMS, tandis que sa consommation mémoire est plus importante. La majorité de cette consommation mémoire est due en particulier à la représentation et la manipulation des séquences de peptides.

Les résultats de l'identification par les deux logiciels sont présentés Table 7.6. Les résultats de SpecOMS sont présentés avec des FDRs locales <1%, comme dans toutes les tables précédentes. En appliquant

		SpecOMS	MSFragger	MSFragger (FDR locale)
groupe 1	FDR en % nombre d'identifications	0.87 12 694	0.01 10 617	< 0.1 12 890
groupe 2	FDR en % nombre d'identifications	0.73 3 821	1.30 6 286	1.0 5 862
groupe 3'	FDR en % nombre d'identifications	0.96 625	2.90 1 971	1.0 1 659
Total	FDR en % nombre d'identifications	0.84 17 140	0.85 18 874	<1.0 20 411
	temps d'exécution (minutes) consommation mémoire (Go)	48.90 8.8	60 5.4	60 5.4

TABLE 7.6 : Nombre d'identifications obtenues pour MSFragger et SpecOMS pour une exécution avec une FDR globale < 0.85%.

ces FDRs locales, la FDR globale correspondante est <0.85% ; nous avons donc présenté les résultats de MSFragger à la fois avec une FDR globale de <0.85% et avec une FDR locale <1%. Sans surprise, l'usage d'une FDR globale introduit une répartition des erreurs d'identification peu homogène pour MSFragger. En effet les FDRs associées aux identifications avec un delta de masse positif (1.30%) et négatifs (2.90%) ne sont pas en dessous du seuil des 1%, contrairement aux résultats fournis par SpecOMS.

Nous pouvons observer que SpecOMS présente une capacité à identifier les spectres non modifiés très similaire à celle de MSFragger, malgré une évaluation de la similarité entre spectres plus simple. En effet, lors de l'utilisation d'une FDR locale, le nombre d'identifications atteint par MSFragger (12 890) est proche de celui atteint par SpecOMS (12 964).

Par contre, nous pouvons constater que MSFragger identifie nettement plus de peptides avec un delta de masse négatif, soit 1 971 identifications pour MSFragger contre 625 pour SpecOMS si l'on ignore les deltas de masse inférieurs à 400 Da. Le nombre d'identifications obtenues par MSFragger dans le cas de l'utilisation d'une FDR locale diminue néanmoins cet écart (1 659). Ces résultats mettent une nouvelle fois en valeur le problème des PSMs associées aux deltas de masse négatifs qui explique probablement le défaut de comportement observé de SpecOMS.

De manière plus surprenante, un écart important peut être souligné pour les identifications avec un delta de masse positif, pour lesquelles MSFragger identifie 6 286 peptides avec la FDR globale ou 5 862 avec une FDR locale. Bien sûr, SpecOMS a un calcul de score plus simple qui induit un effet de seuil pour le calcul de la FDR qui peut le pénaliser, mais il dispose en contre-partie de la capacité à réaligner les spectres avec SpecFit avant le choix du meilleur candidat. Pour compléter l'analyse, nous avons donc comparé le comportement des deux logiciels sur les quatre modifications les plus fréquentes dans le jeu de spectres (cf. Table 7.7). SpecOMS est capable d'identifier un nombre plus élevé de peptides porteurs d'oxydation ou de formylation, tandis que MSFragger retrouve plus d'erreurs isotopiques et de peptides porteurs de carbamylation. Au final, nous observons un nombre d'identifications parfaitement équivalent sur l'ensemble des quatre modifications. Ceci implique que la différence entre le nombre de peptides modifiés trouvés concerne uniquement les peptides porteurs de modifications plus rares. Il serait tout à fait utile de creuser cette différence pour mieux comprendre l'intérêt (ou pas) du ré-alignement des spectres avant le choix du meilleur peptide candidat.

Pour vraiment comprendre les différences de comportement entre les deux logiciels, il serait aussi nécessaire d'étendre les tests à tout un ensemble de jeux de spectres, pour mieux maitriser, entre autre, la variabilité de comportement.

	SpecOMS		MSFragger	
	nombre d'identifications	FDR en %	nombre d'identifications	FDR en %
Erreur isotopique (1Da)	791	< 0.1	880	< 0.1
Oxydation	1 130	< 0.1	950	0.5
Carbamylation	455	< 0.3	572	< 0.2
Formylation	250	0.4	224	< 0.1
Total	2 626		2 626	

TABLE 7.7 : Comparaison des taux d'identification entre MSFragger et SpecOMS sur les 4 modifications les plus fréquentes. La FDR est calculée localement.

Nous avons dans ce chapitre discuté de l'influence des différents paramètres sur la sensibilité de SpecOMS. Nous avons ensuite présenté plusieurs cadres d'usage du logiciel (identification de modifications rares, profilage de jeux de données, détection des artefacts expérimentaux...) puis comparé celui-ci avec un autre logiciel d'identification de peptides de type OMS. Nous avons montré que SpecOMS est capable d'effectuer des analyses de données de spectrométrie de masse de volumétrie habituelle en un temps raisonnable sur une station de travail classique. Nous avons également montré qu'avec une bonne configuration des paramètres, SpecOMS est capable, malgré sa fonction de score très simple, d'obtenir des résultats très proches des méthodes d'identification avec filtre de masse ou bien du logiciel OMS actuellement le plus utilisé.

8

Conclusion et perspectives

8.1 Conclusion

Nous avons présenté dans ce manuscrit notre contribution au domaine de l'identification de spectres obtenus par spectrométrie de masse sous la forme d'une nouvelle méthode ne nécessitant pas l'utilisation de filtre de masse. L'objectif de ce travail était d'apporter des réponses aux critiques couramment avancées à l'égard de ces méthodes développées jusqu'à présent : coût élevé en puissance de calcul, et sensibilité fortement réduite.

Nous avons montré au travers de différentes analyses que SpecOMS était capable d'effectuer rapidement une analyse d'une collection de spectres de masse sur une station de travail standard avec une consommation mémoire restant acceptable (4 à 9 Go) pour des volumétries conformes à celles générées par les appareils actuels. Nous avons également montré que SpecOMS, contrairement aux idées souvent avancées jusqu'à présent, n'introduisait pas une sensibilité fortement dégradée en raison de l'absence de filtre de masse.

Plusieurs facteurs sont à l'origine de l'efficacité de SpecOMS. La répartition des masses dans les différents bins effectuée dès le début du traitement permet de limiter les comparaisons aux couples de masses effectivement présents dans les spectres. Cette technique, partagée avec l'algorithme MSFragger, accélère notablement les temps de calcul en évitant de nombreuses comparaisons inutiles. Nous avons également réglé de manière satisfaisante la prise en charge de l'incertitude en dupliquant les masses dans les différents bins. Ensuite, la structure de données SpecTrees, au coeur de SpecOMS, permet une compression de l'information necessaire pour le calcul du nombre de pics en commun entre les spectres, réduisant ainsi l'usage de la mémoire. Puis, SpecXTract optimise le parcours de la structure SpecTrees pour fournir rapidement les PSMs (s_i, s_j) et leurs deltas de masse associés. Enfin, si l'utilisateur le souhaite, SpecFit tente d'améliorer le score des PSMs candidats avant de choisir le meilleur d'entre eux, fonctionnement original par rapport aux logiciels concurrents (à l'heure de la rédaction de ce document), permettant de "rattraper" certaines identifications. Nous sommes convaincus que SpecOMS revêt un intérêt notable pour la communauté qui travaille sur l'identification des peptides par spectrométrie de masse en mode tandem. Cet interêt peut s'observer sur la Figure 8.1, qui représente le nombre de téléchargements du logiciel SpecOMS dans ses différentes versions depuis sa première publication.

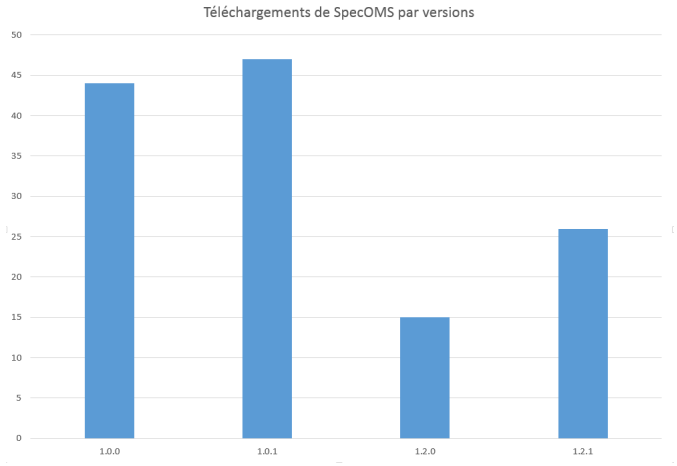


FIGURE 8.1 : Nombre de téléchargements de SpecOMS par version du logiciel depuis des adresses IP distinctes.

Nous avons toutefois identifié plusieurs voies d'améliorations de SpecOMS, que nous n'avons pas pu implémenter faute de temps et que nous discutons ci-après.

Nouvelle fonction de score

Décliné comme le nombre de pics en commun entre deux spectres, le score implémenté dans SpecOMS a le très grand avantage d'être facilement appréhendé du fait de sa simplicité, ce qui pourrait rester un avantage dans les perspectives que nous dégageons plus loin. Ce score s'appuie sur l'hypothèse que l'amélioration des mesures de masse des ions fragments des nouveaux appareils de spectrométrie de masse rend le nombre de masses partagées entre les spectres suffisant pour discriminer les peptides et identifier un grand nombre de spectres, même sans filtre de masse. Cependant, si comme nous l'avons montré dans ce document, la sensibilité de SpecOMS avec ce score simple est suffisante pour identifier, sur le jeu de données HEK293, à peu près autant de spectres que le fait un logiciel d'identification avec filtre de masse (X !Tandem), SpecOMS identifie légèrement moins de spectres que ne le fait MSFragger, qui dispose d'un système de score plus élaboré. En particulier, le système de score utilisé par SpecOMS ne prend en compte ni l'intensité, ni la présence de séries d'ions fragments dans un spectre. De plus, la nature non continue du système de score de SpecOMS conduit à des effets de seuils lors du calcul de la FDR. Pour obtenir une meilleure sensibilité, une amélioration possible de SpecOMS serait donc de mettre au point une fonction de score qui permette de calculer un seuil de FDR sur une plage de valeurs continues, plutôt que discrétisées. Ce calcul de score ne devrait néanmoins pas impacter les performances de SpecOMs de manière significative. Pour ce faire, nous proposons de ne modifier ni la structure de données, ni l'algorithme SpecXtract. Nous proposons donc plutôt de calculer un score plus complet grâce à l'algorithme SpecFit. Ainsi, pour chaque PSM retenu par SpecXtract sur la base du nombre de masses partagées, SpecFit peut recalculer un score plus complexe pour ce PSM ainsi que pour d'éventuels PSMs qui résultent de l'alignement des masses par SpecFit. Ce calcul de score peut permettre de prendre en compte l'intensité des ions du spectre expérimental, les séries d'ions successives, etc. Ainsi, il serait possible d'obtenir des valeurs de scores continues, lesquelles permettraient de fixer un seuil de FDR plus finement (au plus près de 1%). Nous pensons que ce score nous permettrait d'atteindre une meilleure sensibilité en récupérant des identifications actuellement non validées statistiquement.

Parallélisation

Nous avons choisi, lors du développement de SpecOMS, de ne pas prioriser la parallélisation du logiciel. Néanmoins, les logiciels exécutés sur des ordinateurs modernes peuvent tirer parti de la présence de multiples processeurs sur la même puce, ou bien connectés en réseau, et ce afin d'améliorer significativement

les performances en terme de temps de calcul. Paralléliser les opérations effectuées par SpecOMS devrait permettre d'obtenir un gain de temps conséquent. La majorité des tâches effectuées par SpecOMS peuvent être partagées entre plusieurs processus, chacun de ces derniers traitant un sous-ensemble de spectres expérimentaux. Pour éviter une augmentation de l'encombrement mémoire, une unique structure de données SpecTrees pourrait être partagée entre les processus. Ensuite, l'ensemble des spectres expérimentaux à analyser serait divisé équitablement entre tous les processus. Ainsi, chacun de ces processus pourrait exécuter ses propres instances de SpecXtract et de SpecFit, en utilisant les données lues depuis la structure SpecTrees partagée mais sans y apporter de modifications. Chaque processus pourrait donc proposer un ensemble de PSMs pour les sous-ensemble de spectres expérimentaux qu'il a analysé. Finalement, le résultat complet de l'exécution de SpecOMS par de multiples processus parallèles serait constitué de l'union des résultats fournis par chacun des processus. En fonction des besoins en mémoire et de l'architecture parallèle sollicitée, plusieurs gestions différentes de la structure SpecTrees peuvent être envisagées. Cette dernière pourrait être intégralement partagée entre les processus comme mentionné précédemment, ou bien plusieurs instances de SpecTrees, contenant des ensembles de spectres expérimentaux différents, pourraient être utilisées par des processus exécutés sur des machines physiques différentes. Pour obtenir un temps d'exécution plus faible sans augmenter la consommation mémoire, nous proposons une autre possibilité. Il serait possible de partager entre les différents processus la partie de SpecTrees constituée uniquement des spectres théoriques. Chaque processus construirait ensuite un complément de la structure SpecTrees constituée uniquement des spectres expérimentaux, ainsi qu'un ensemble de références vers la partie de SpecTrees partagée entre les processus. Pour ce faire, il serait nécessaire de pouvoir construire la structure SpecTrees en deux temps : (i) construire la partie de SpecTrees qui contient l'ensemble des spectres théoriques, (ii) ajouter n'importe quel ensemble de spectres expérimentaux à cette structure en assurant la bonne exécution de SpecXtract pour cet ensemble de spectres expérimentaux. En procédant ainsi, chaque processus serait alors en mesure d'exécuter sa version locale des algorithmes SpecXtract et SpecFit en manipulant uniquement les spectres théoriques et le sous-ensemble de spectres expérimentaux distribués à ce processus (et non la structure SpecTrees dans son ensemble).

Découpage en exécutions multiples

Lors de l'identification de jeux de spectres volumineux, nous proposons aux utilisateurs de les séparer en plusieurs sous-ensembles, puis de réaliser plusieurs exécutions de SpecOMS afin de minimiser le temps d'exécution et de réduire l'espace mémoire utilisé. Cette méthode est particulièrement intéressante parce que le temps de construction de SpecTrees par l'algorithme SpecGrowth est petit devant le temps d'exécution total de SpecOMS, mais aussi parce que le temps d'exécution de SpecOMS ne croît pas de manière linéaire avec la taille des données. SpecOMS pourrait réaliser de manière transparente à l'utilisateur la division du jeu de spectres en plusieurs sous-ensembles et restituer l'union des résultats après la construction de plusieurs structures de données SpecTrees et leur exploitation. En pratique, une telle adaptation de SpecOMS comporterait des points communs avec la parallélisation du logiciel où plusieurs sous-ensembles de spectres, seraient analysés séquentiellement plutôt qu'en parallèle.

Amélioration de l'empreinte mémoire

Comme nous l'avons illustré dans ce manuscrit, SpecOMS a une consommation mémoire supérieure à celle de MSFragger. En réalité, la consommation mémoire de SpecTrees et des algorithmes associés reste similaire à la consommation mémoire de MSFragger. L'excédent de mémoire utilisé par SpecOMS est dû à une représentation non optimisée des protéines et des peptides, deux éléments qui n'étaient pas au centre de la problématique soulevée dans le cadre de cette thèse. Il serait possible de réduire significativement l'espace mémoire consommé par ces données en les représentant différemment : conserver en mémoire uniquement la séquence de chaque protéine et construire directement les spectres théoriques à partir de celle-ci. Les informations concernant le peptide à partir duquel chaque spectre théorique est construit seraient condensées sous la forme d'un quadruplet de valeurs : l'index de la protéine à partir de laquelle le peptide est obtenu, sa

position dans la protéine, sa longueur en nombre d'acides aminés et un témoin indiquant si le spectre provient de la banque target ou bien de la banque decoy. En procédant de la sorte, il serait possible de réduire très significativement l'empreinte mémoire de SpecOMS.

Les différentes voies d'améliorations que nous avons présenté ci-dessus devraient permettre à SpecOMS de très bien se positionner parmi l'ensemble des logiciels OMS disponibles, à la fois en terme d'utilisation des ressources et en terme de sensibilité.

8.2 Perspectives

Même si les logiciels OMS améliorent les identifications (et SpecOMS y contribue), le pourcentage de spectres identifiés à l'issue d'une analyse de spectrométrie de masse reste malgré tout encore très bas. L'une des explications possibles est la présence de plusieurs modifications sur le même peptide, situation qui est encore mal prise en charge par les algorithmes. L'enjeu pour les sciences de la vie est très important : l'inventaire complet des protéines sous leurs différentes formes (avec leurs modifications) est une étape importante pour mieux comprendre le fonctionnement cellulaire. Les algorithmes que nous avons développés dans le cadre de cette thèse pourront être utilisés pour essayer de lever les verrous méthodologiques pour identifier les protéines modifiées. En effet, dans les travaux que nous avons présentés, nous avons limité l'utilisation de SpecOMS a un cadre très classique de comparaison des spectres expérimentaux avec des spectres théoriques obtenus à partir des protéines d'une banque. Cependant, la structure de données SpecTrees et les algorithmes de Bucket Clustering, SpecGrowth et SpecXtract permettent de déterminer rapidement le cardinal de l'intersection entre deux ensembles dans une collection d'ensembles qui contiennent des données de même type. Ces dernières peuvent représenter des masses, comme dans notre cas, ou bien des intensités, des temps de rétention, ou même tout autre information non nécessairement relative à la spectrométrie de masse. Tout particulièrement, il nous intéresse d'utiliser la structure SpecTrees et les algorithmes associés soit pour calculer les similarités uniquement entre des spectres expérimentaux, soit uniquement entre des spectres théoriques.

Similarités entre spectres expérimentaux

Calculer rapidement la similarité entre n'importe quelle paire de spectres expérimentaux nécessite de modifier principalement les entrées et sorties de SpecOMS. Outre ces modifications, il peut être nécessaire de concevoir un nouveau score spécifiquement pour cette utilisation ou au moins d'évaluer différemment la pertinence d'un nombre de masses partagées donné. En effet, le nombre de masses partagées entre deux spectres expérimentaux est très différent de celui partagé entre un spectre expérimental et un spectre théorique, en particulier puisque la précision de l'appareil sur les ions fragments doit être prise en compte pour les deux spectres expérimentaux. Comparer les spectres expérimentaux d'un échantillon entre eux fournit des informations relatives au mélange de peptides analysé. Dans ce mélange, certains peptides sont présents en de multiples exemplaires, certains portant des modifications chimiques et d'autres non. Il doit donc être possible, en exploitant les similarités entre les spectres issus de ces peptides, de conforter des identifications faites par SpecOMS ou même d'atteindre de nouvelles identifications.

Similarités entre spectres théoriques

Il est également possible de modifier les entrées et sorties de SpecOMS pour calculer la similarité pour n'importe quelle paire de spectres théoriques. Dans ce cas, considérer le score actuel proposé pour SpecOMS sous la forme du simple nombre de masses partagées est probablement suffisant pour fournir une compréhension générale de la ressemblance entre deux peptides. Nous pensons qu'il doit ainsi être possible de déterminer pour une banque de protéines donnée quels sont les peptides qui ont de fortes chances d'être

identifiés par erreur. Une telle analyse est plus longue à réaliser que l'identification des spectres expérimentaux, et nécessite probablement la parallélisation des algorithmes de SpecOMS. Néanmoins, en raison du rythme d'évolution des banques de protéines, les résultats de cette analyse pourraient vraisemblablement être conservés pour de nombreuses utilisations successives.

Construction du graphe des similarités

Grâce à l'ensemble des relations qui existent entre les spectres, qu'ils soient expérimentaux ou théoriques, nous pouvons construire le graphe des similarités entre spectres. Dans un tel graphe, un spectre est représenté par un nœud, et 2 nœuds sont reliés par une relation s'ils ont suffisamment de pics en commun. L'identification de spectres à l'aide de graphes de spectres (spectral networks) a été introduite il y a déjà une dizaine d'années [10]. Cependant, jusqu'à présent, l'usage de cette approche très prometteuse a été limitée par la capacité à générer et à manipuler les graphes de spectres de très grande dimension [59]. En effet, le graphe des similarités entre spectres que nous imaginons est un objet volumineux, qui peut comprendre plusieurs millions de sommets (spectres expérimentaux, spectres théoriques), et nous nous attendons à ce que ces sommets soient reliés par plusieurs dizaines de millions d'arêtes.

Cependant, ce graphe représenterait une vue d'ensemble inédite des relations entre tous les spectres d'une expérimentation et nous permettrait certainement de mieux comprendre les défauts d'identification. Dans le cadre de cette thèse, nous n'avons pas pu développer ces possibilités, mais quelques travaux préliminaires ont pu être réalisés. Nous avons de plus implémenté les modifications nécessaires à SpecOMS pour obtenir les relations entre spectres expérimentaux.

La représentation sous forme de graphe peut aussi être utilisée de manière exploratoire, c'est-à-dire pour essayer de mettre en évidence des cas particuliers ou des situations d'intérêt. Pour représenter et manipuler le graphe de manière efficace, la solution qui nous est apparue la meilleure a été d'utiliser une base de données graphe. Nous avons plus spécifiquement choisi la base de données Neo4J (`https://neo4j.com/`) pour stocker et manipuler les données produites par SpecOMS.

La Figure 8.2 présente une proposition simple de modèle pour un tel graphe des similarités créé à l'aide de Neo4J. Nous pouvons y voir quatre grands ensembles de sommets : les spectres expérimentaux en bleu, les spectres théoriques en jaune, les protéines issues de la banque decoy en rouge et les protéines issues de la banque target en vert. Les spectres sont reliés par des arêtes qui étiquetées avec la similarité calculée entre ces deux spectres et leur delta de masse. Un spectre théorique est relié à une protéine si la séquence dont il est issu a été digérée à partir de cette protéine. Nous sommes convaincus qu'une telle représentation, en permettant l'exploitation des informations sous forme condensée, ouvre des possibilités nouvelles pour améliorer la performance des logiciels d'identification de spectres et offre l'accès à une meilleure compréhension des résultats. En particulier, cette représentation peut être exploitée soit pour effectuer des requêtes spécifiques dans le but de valider des hypothèses, soit de manière exploratoire pour comprendre pourquoi certains spectres restent non identifiés.

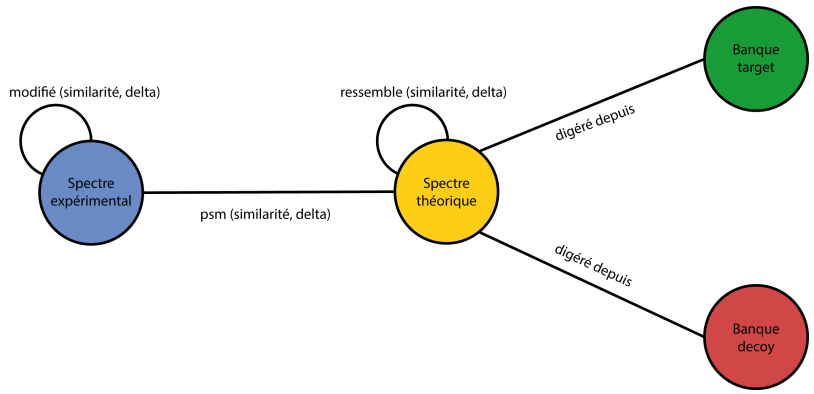


FIGURE 8.2 : Proposition de schéma de base de données graphe utilisée pour stocker les résultats de SpecOMS sous Neo4J.

Pour illustrer les potentialités de l'approche, nous proposons deux cas d'utilisation : Figure 8.3, nous montrons comment l'usage du graphe pourrait aider l'interprétation des spectres porteurs de plusieurs modifications en utilisant la présence de différents variants du même peptide dans le jeu de spectres. Rappelons que cette interprétation est encore difficile à atteindre aujourd'hui par l'ensemble des logiciels.

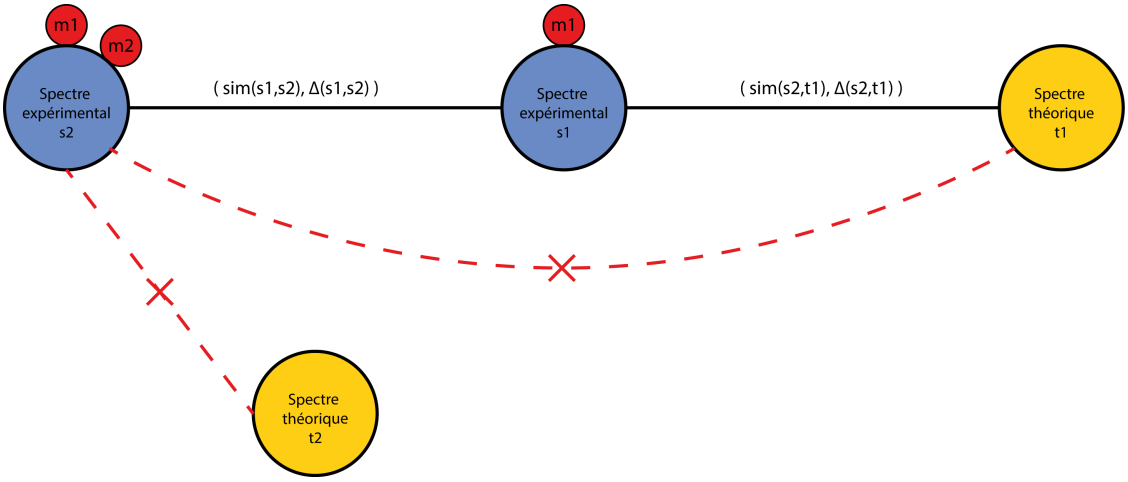


FIGURE 8.3 : Analyse d'un spectre qui porte deux modifications à l'aide de la représentation sous forme de graphe.

La Figure 8.4, quant à elle, illustre la possibilité de « rattraper » des identifications non validées par l'information contenue dans les relations avec le voisinage, ceci pour permettre un taux d'identification plus élevé.

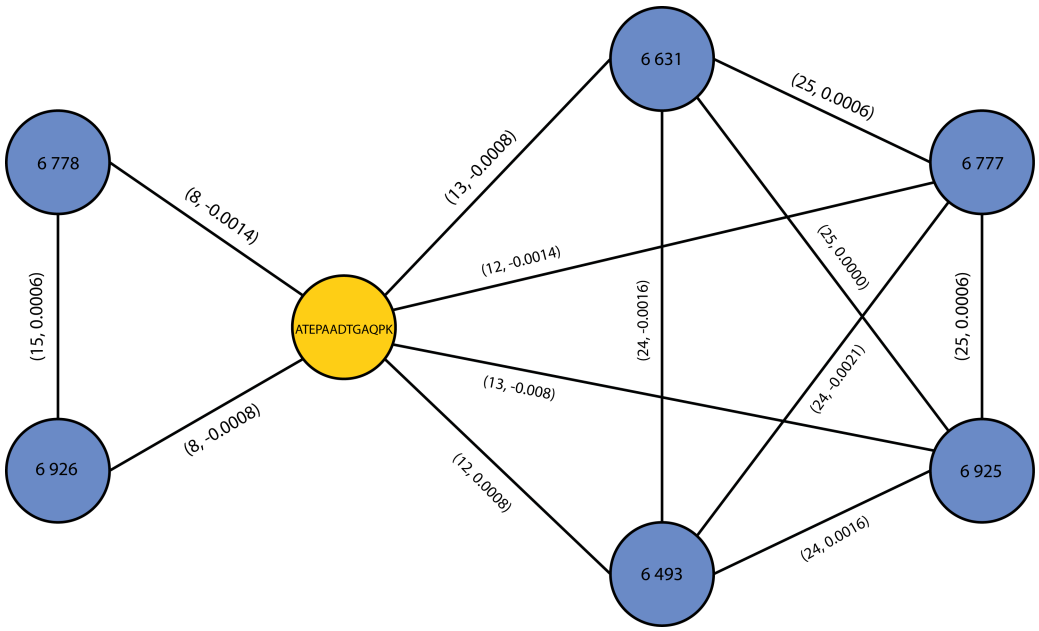


FIGURE 8.4 : Sous-graphe des similarités associé à un peptide extrait de l'interprétation du jeu HEK293. Les identifiants des spectres et la séquence du peptide sont reportés dans les nœuds. Deux spectres expérimentaux sont reliés s'ils partagent au moins 15 pics en commun, tandis qu'un spectre expérimental et un peptide sont reliés s'ils partagent au moins 7 pics en commun. La précision attribuée (0.02 Da) ne permet pas de relier tous les spectres expérimentaux entre eux. Pourtant, ces 6 spectres expérimentaux sont reliés au même peptide avec un delta de masse proche de 0. En élargissant cette précision (au-delà de la valeur estimée pour l'appareil), nous avons néanmoins constaté que les 2 spectres 6778 et 6926 identifient sans ambiguïté le même peptide que les autres spectres et que les 6 spectres expérimentaux et le peptide forment un graphe complet.

Au travers ces deux exemples, nous espérons avoir montré que le graphe des similarités entre spectres est un outil puissant, dont l'utilisation peut amener une meilleure compréhension du processus d'analyse des spectres de masses expérimentaux et aider au développement de meilleurs algorithmes d'identification.

Il reste néanmoins un important travail de fouille de données à effectuer sur ce graphe, notamment parce qu'il n'est pas évident de formuler des hypothèses à tester à l'aide de celui-ci, et parce que son volume et la densité des arêtes rendent le travail exploratoire fastidieux.

Liste des tableaux

Table des figures

Bibliographie

[1] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, pages 487–499, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc. 55

[2] Erik Ahrné, Alexandre Masselot, Pierre-Alain Binz, Markus Müller, and Frederique Lisacek. A simple workflow to increase ms2 identification rate by subsequent spectral library search. *Proteomics*, 9(6) :1731–1736, 2009. 40

[3] Erik Ahrné, Markus Müller, and Frederique Lisacek. Unrestricted identification of modified proteins using ms/ms. *Proteomics*, 10(4) :671–686, 2010. 51, 108

[4] Erik Ahrné, Frederic Nikitin, Frederique Lisacek, and Markus Muller. Quickmod : a tool for open modification spectrum library searches. *Journal of proteome research*, 10(7) :2913–2921, 2011. 53

[5] Gelio Alves and Yi-Kuo Yu. Improving peptide identification sensitivity in shotgun proteomics by stratification of search space. *Journal of proteome research*, 12(6) :2571–2581, 2013. 42

[6] Pedro Alves, Randy J Arnold, Milos V Novotny, Predrag Radivojac, James P Reilly, and Haixu Tang. Advancement in protein inference from shotgun proteomics using peptide detectability. In *Biocomputing 2007*, pages 409–420. World Scientific, 2007. 43

[7] Manor Askenazi and Michal Linial. Implicit biology in peptide spectral libraries. *Analytical chemistry*, 84(18) :7919–7925, 2012. 38

[8] Vineet Bafna and Nathan Edwards. Scope : a probabilistic model for scoring tandem mass spectra against a peptide database. *Bioinformatics*, 17(suppl_1) :S13–S21, 2001. 37, 38

[9] Vineet Bafna and Nathan Edwards. On de novo interpretation of tandem mass spectra for peptide identification. In *Proceedings of the seventh annual international conference on Research in computational molecular biology*, pages 9–18. ACM, 2003. 37

[10] Nuno Bandeira, Dekel Tsur, Ari Frank, and Pavel A Pevzner. Protein identification by spectral networks analysis. *Proceedings of the National Academy of Sciences*, 104(15) :6140–6145, 2007. 121

[11] C Bartel. Fast algorithm for peptide sequencing by mass spectrometry. *Biomed. Environ. Mass. Spectrom*, 19 :363–368, 1990. 36

[12] Martin Beck, Alexander Schmidt, Johan Malmstroem, Manfred Claassen, Alessandro Ori, Anna Szymborska, Franz Herzog, Oliver Rinner, Jan Ellenberg, and Ruedi Aebersold. The quantitative proteome of a human cell line. *Molecular systems biology*, 7(1) :549, 2011. 15

[13] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate : a practical and powerful approach to multiple testing. *Journal of the royal statistical society. Series B (Methodological)*, pages 289–300, 1995. 42

[14] Marshall Bern, Yuhan Cai, and David Goldberg. Lookup peaks : a hybrid of de novo sequencing and database search for protein identification by tandem mass spectrometry. *Analytical chemistry*, 79(4) :1393–1400, 2007. 52

[15] K Biemann, Conrad Cone, Brian R Webster, and Guy P Arsenault. Determination of the amino acid sequence in oligopeptides by computer interpretation of their high-resolution mass spectra1. *Journal of the American Chemical Society*, 88(23) :5598–5606, 1966. 41

[16] Boris Bogdanow, Henrik Zauber, and Matthias Selbach. Systematic errors in peptide and protein identification and quantification by modified peptides. *Molecular & Cellular Proteomics*, 15(8) :2791–2801, 2016. 51

[17] Michael T Boyne, Benjamin A Garcia, Mingxi Li, Leonid Zamdborg, Craig D Wenger, Shannee Babai, and Neil L Kelleher. Tandem mass spectrometry with ultrahigh mass accuracy clarifies peptide identification by database retrieval. *Journal of proteome research*, 8(1) :374–379, 2008. 53

[18] Rl Breathnach, CKOH Benoist, K O'hare, F Gannon, and P Chambon. Ovalbumin gene : evidence for a leader sequence in mrna and dna sequences at the exon-intron boundaries. *Proceedings of the National Academy of Sciences*, 75(10) :4853–4857, 1978. 9

[19] Meghan C Burke, Yuri A Mirokhin, Dmitrii V Tchekhovskoi, Sanford P Markey, Jenny Heidbrink Thompson, Christopher Larkin, and Stephen E Stein. The hybrid search : A mass spectral library search method for discovery of modifications in proteomics. *Journal of proteome research*, 16(5) :1924–1935, 2017. 45, 53

[20] Ting Chen, Ming-Yang Kao, Matthew Tepel, John Rush, and George M Church. A dynamic programming approach to de novo peptide sequencing via tandem mass spectrometry. *Journal of Computational Biology*, 8(3) :325–337, 2001. 37

[21] Hao Chi, Haifeng Chen, Kun He, Long Wu, Bing Yang, Rui-Xiang Sun, Jianyun Liu, Wen-Feng Zeng, Chun-Qing Song, Si-Min He, et al. pnovo+ : de novo peptide sequencing using complementary hcd and etd tandem mass spectra. *Journal of proteome research*, 12(2) :615–625, 2012. 37

[22] Hao Chi, Rui-Xiang Sun, Bing Yang, Chun-Qing Song, Le-Heng Wang, Chao Liu, Yan Fu, Zuo-Fei Yuan, Hai-Peng Wang, Si-Min He, et al. pnovo : de novo peptide sequencing and identification using hcd spectra. *Journal of proteome research*, 9(5) :2713–2724, 2010. 37

[23] Joel M Chick, Deepak Kolippakkam, David P Nusinow, Bo Zhai, Ramin Rad, Edward L Huttlin, and Steven P Gygi. A mass-tolerant database search identifies a large proportion of unassigned spectra in shotgun proteomics as modified peptides. *Nature biotechnology*, 33(7) :743, 2015. 45, 53

[24] Deanna M Church, Leo Goodstadt, LaDeana W Hillier, Michael C Zody, Steve Goldstein, Xinwe She, Carol J Bult, Richa Agarwala, Joshua L Cherry, Michael DiCuccio, et al. Lineage-specific biology revealed by a finished genome assembly of the mouse. *PLoS biology*, 7(5) :e1000112, 2009. 22

[25] Jacques Colinge, Alexandre Masselot, Marc Giron, Thierry Dessingy, and Jérôme Magnin. Olav : Towards high-throughput tandem mass spectrometry data identification. *PROTEOMICS : International Edition*, 3(8) :1454–1463, 2003. 40, 52

[26] Robertson Craig and Ronald C Beavis. A method for reducing the time required to match protein sequences with tandem mass spectra. *Rapid communications in mass spectrometry*, 17(20) :2310–2316, 2003. 52

[27] Robertson Craig and Ronald C Beavis. Tandem : matching proteins with tandem mass spectra. *Bioinformatics*, 20(9) :1466–1467, 2004. 37, 41

[28] Robertson Craig, JC Cortens, David Fenyo, and Ronald C Beavis. Using annotated peptide mass spectrum libraries for protein identification. *Journal of proteome research*, 5(8) :1843–1849, 2006. 38

[29] David M. Creasy and John S. Cottrell. Unimod : Protein modifications for mass spectrometry. *PROTEOMICS*, 4(6) :1534–1536. 21, 51

[30] David M Creasy and John S Cottrell. Error tolerant searching of uninterpreted tandem mass spectrometry data. *Proteomics*, 2(10) :1426–1434, 2002. 52

[31] Vlado Dančík, Theresa A Addona, Karl R Clauser, James E Vath, and Pavel A Pevzner. De novo peptide sequencing via tandem mass spectrometry. *Journal of computational biology*, 6(3-4) :327–342, 1999. 36, 37

[32] Matthieu David, Guillaume Fertin, Hélène Rogniaux, and Dominique Tessier. Specoms : A full open modification search method performing all-to-all spectra comparisons within minutes. *Journal of Proteome Research*, 16(8) :3030–3038, 2017. PMID : 28660767. 87

[33] Matthieu DAVID, Guillaume Fertin, and Dominique D. Tessier. SpecTrees : an efficient without a priori data structure for MS/MS spectra identification. In Martin Frith and Christian N. S. Pedersen, editors, *16th Workshop on Algorithms in Bioinformatics (WABI 2016)*, Lecture Notes in Bioinformatics, Aarhus, Denmark, August 2016. Springer-Verlag. 97

[34] A. J. Dempster. A new method of positive ray analysis. *Phys. Rev.*, 11 :316–325, Apr 1918. 23

[35] Frank Desiere, Eric W Deutsch, Nichole L King, Alexey I Nesvizhskii, Parag Mallick, Jimmy Eng, Sharon Chen, James Eddes, Sandra N Loevenich, and Ruedi Aebersold. The peptideatlas project. *Nucleic acids research*, 34(suppl_1) :D655–D658, 2006. 38

[36] Benjamin J Diament and William Stafford Noble. Faster sequest searching for peptide identification from tandem mass spectra. *Journal of proteome research*, 10(9) :3871–3879, 2011. 38, 54

[37] Hester A Doyle and Mark J Mamula. Posttranslational modifications of self-antigens. *Annals of the New York Academy of Sciences*, 1050(1) :1–9, 2005. 21

[38] du Bois H. Ueber magnetische schirmwirkung. *Annalen der Physik*, 301(5) :1–37. 23

[39] Ingvar Eidhammer, Kristian Flikka, Lennart Martens, and Svein-Ole Mikalsen. *Computational methods for mass spectrometry proteomics*. John Wiley & Sons, 2008. 24

[40] Joshua E Elias, Francis D Gibbons, Oliver D King, Frederick P Roth, and Steven P Gygi. Intensity-based protein identification by machine learning from a library of tandem mass spectra. *Nature biotechnology*, 22(2) :214, 2004. 38, 40

[41] Joshua E Elias and Steven P Gygi. Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry. *Nature methods*, 4(3) :207, 2007. 42

[42] Jimmy K Eng, Bernd Fischer, Jonas Grossmann, and Michael J MacCoss. A fast sequest cross correlation algorithm. *Journal of proteome research*, 7(10) :4598–4602, 2008. 38

[43] Jimmy K Eng, Ashley L McCormack, and John R Yates. An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. *Journal of the American Society for Mass Spectrometry*, 5(11) :976–989, 1994. 38, 41

[44] Jimmy K Eng, Ashley L McCormack, and John R Yates. An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. *Journal of the American Society for Mass Spectrometry*, 5(11) :976–989, 1994. 45

[45] Logan J Everett, Charlene Bierl, and Stephen R Master. Unbiased statistical analysis for multi-stage proteomic search strategies. *Journal of proteome research*, 9(2) :700–707, 2010. 52

[46] Jian Feng, Daniel Q Naiman, and Bret Cooper. Probability model for assessing proteins assembled from peptide sequences inferred from tandem mass spectrometry data. *Analytical chemistry*, 79(10) :3901–3911, 2007. 43

[47] John B Fenn, Matthias Mann, Chin Kai Meng, Shek Fu Wong, and Craig M Whitehouse. Electrospray ionization for mass spectrometry of large biomolecules. *Science*, 246(4926) :64–71, 1989. 29

[48] Jorge Fernández-de Cossio, Javier Gonzalez, and Vladimir Besada. A computer program to aid the sequencing of peptides in collision-activated decomposition experiments. *Bioinformatics*, 11(4) :427–434, 1995. 36, 37

[49] Ari Frank and Pavel Pevzner. Pepnovo : de novo peptide sequencing via probabilistic network modeling. *Analytical chemistry*, 77(4) :964–973, 2005. 36, 37

[50] Ari Frank, Stephen Tanner, Vineet Bafna, and Pavel Pevzner. Peptide sequence tags for fast database search in mass-spectrometry. *Journal of proteome research*, 4(4) :1287–1295, 2005. 42

[51] Barbara E Frewen, Gennifer E Merrihew, Christine C Wu, William Stafford Noble, and Michael J MacCoss. Analysis of peptide ms/ms spectra from large-scale proteomics experiments using spectrum libraries. *Analytical chemistry*, 78(16) :5678–5684, 2006. 38

[52] Yan Fu. Bayesian false discovery rates for post-translational modification proteomics. *Statistics and Its Interface*, 5(1) :47–59, 2012. 51

[53] Yan Fu, Li-Yun Xiu, Wei Jia, Ding Ye, Rui-Xiang Sun, Xiao-Hong Qian, and Si-Min He. Deltamt : a statistical algorithm for fast detection of protein modifications from lc-ms/ms data. *Molecular & Cellular Proteomics*, pages mcp–M110, 2011. 53

[54] Lewis Y Geer, Sanford P Markey, Jeffrey A Kowalak, Lukas Wagner, Ming Xu, Dawn M Maynard, Xiaoyu Yang, Wenyao Shi, and Stephen H Bryant. Open mass spectrometry search algorithm. *Journal of proteome research*, 3(5) :958–964, 2004. 42

[55] Sarah Gerster, Ermir Qeli, Christian H Ahrens, and Peter Bühlmann. Protein and gene model inference based on statistical modeling in k-partite graphs. *Proceedings of the National Academy of Sciences*, 107(27) :12101–12106, 2010. 44

[56] Mark A Goldberg, Susan P Dunning, and H Franklin Bunn. Regulation of the erythropoietin gene : evidence that the oxygen sensor is a heme protein. *Science*, 242(4884) :1412–1415, 1988. 9

[57] Johannes Griss, Yasset Perez-Riverol, Steve Lewis, David L Tabb, José A Dianes, Noemi del Toro, Marc Rurik, Mathias Walzer, Oliver Kohlbacher, Henning Hermjakob, et al. Recognizing millions of consistently unidentified spectra across hundreds of shotgun proteomics datasets. *Nature methods*, 13(8) :651, 2016. 45

[58] Monica A Grobei, Ermir Qeli, Erich Brunner, Hubert Rehrauer, Runxuan Zhang, Bernd Roschitzki, Konrad Basler, Christian H Ahrens, and Ueli Grossniklaus. Deterministic protein inference for shotgun proteomics data provides new insights into arabidopsis pollen development and function. *Genome research*, 19(10) :1786–1800, 2009. 44

[59] Adrian Guthals, Jeramie D Watrous, Pieter C Dorrestein, and Nuno Bandeira. The spectral networks paradigm in high throughput mass spectrometry. *Molecular bioSystems*, 8(10) :2535–2544, 2012. 121

[60] CW Hamm, WE Wilson, and DJ Harvan. Peptide sequencing program. *Bioinformatics*, 2(2) :115–118, 1986. 40

[61] Jiawei Han, Jian Pei, and Yiwen Yin. Mining frequent patterns without candidate generation. In *ACM sigmod record*, volume 29, pages 1–12. ACM, 2000. 59

[62] Xi Han, Lin He, Lei Xin, Baozhen Shan, and Bin Ma. Peaksptm : mass spectrometry-based identification of peptides with unspecified modifications. *Journal of proteome research*, 10(7) :2930–2936, 2011. 52

[63] H. Harold. Origin of the word 'protein.'. *Nature. 168 (4267)*, pages 244–244, 1951. 15

[64] Gene Hart-Smith, Daniel Yagoub, Aidan P Tay, Russell Pickford, and Marc R Wilkins. Large scale mass spectrometry-based identifications of enzyme-mediated protein methylation are subject to high false discovery rates. *Molecular & Cellular Proteomics*, 15(3) :989–1006, 2016. 112

[65] Moshe Havilio, Yariv Haddad, and Zeev Smilansky. Intensity-based statistical scorer for tandem mass spectrometry. *Analytical Chemistry*, 75(3) :435–444, 2003. 37

[66] Lin He, Jolene K Diedrich, Yen-Yin Chu, and John Yates. Extracting accurate precursor information for tandem mass spectra by rawconverter. 87, 10 2015. 85

[67] William J Henzel, Todd M Billeci, John T Stults, Susan C Wong, Christopher Grimley, and Colin Watanabe. Identifying proteins from two-dimensional gels by molecular mass searching of peptide fragments in protein sequence databases. *Proceedings of the National Academy of Sciences*, 90(11) :5011–5015, 1993. 42

[68] Alejandro Heredia-Langner, William R Cannon, Kenneth D Jarman, and Kristin H Jarman. Sequence optimization as an alternative to de novo analysis of tandem mass spectrometry data. *Bioinformatics*, 20(14) :2296–2304, 2004. 41

[69] Brandon Ho, Anastasia Baryshnikova, and Grant W Brown. Unification of protein abundance datasets yields a quantitative saccharomyces cerevisiae proteome. *Cell Systems*, 2018. 15

[70] Oliver Horlacher, Frederique Lisacek, and Markus Muller. Mining large scale tandem mass spectrometry data for protein modifications using spectral libraries. *Journal of proteome research*, 15(3) :721–731, 2015. 53

[71] J Jeffry Howbert and William S Noble. Computing exact p-values for a cross-correlation shotgun proteomics score function. *Molecular & Cellular Proteomics*, pages mcp–O113, 2014. 42

[72] Ting Huang, Jingjing Wang, Weichuan Yu, and Zengyou He. Protein inference : a review. *Briefings in bioinformatics*, 13(5) :586–614, 2012. 43

[73] Xin Huang, Lin Huang, Hong Peng, Ashu Guru, Weihua Xue, Sang Yong Hong, Miao Liu, Seema Sharma, Kai Fu, Adam P Caprez, et al. Isptm : an iterative search algorithm for systematic identification of post-translational modifications from complex proteome mixtures. *Journal of proteome research*, 12(9) :3831–3842, 2013. 52

[74] K Ishikawa and Y Niwa. Computer-aided peptide sequencing by fast atom bombardment mass spectrometry. *Biological Mass Spectrometry*, 13(7) :373–380, 1986. 41

[75] Peter James, Manfredo Quadroni, Ernesto Carafoli, and G Gonnet. Protein identification by mass profile fingerprinting. *Biochemical and biophysical research communications*, 195(1) :58–64, 1993. 42

[76] Ole Nørregaard Jensen. Modification-specific proteomics : characterization of post-translational modifications by mass spectrometry. *Current opinion in chemical biology*, 8(1) :33–41, 2004. 21

[77] Chao Ji, Randy J Arnold, Kevin J Sokoloski, Richard W Hardy, Haixu Tang, and Predrag Radivojac. Extending the coverage of spectral libraries : A neighbor-based approach to predicting intensities of peptide fragmentation spectra. *Proteomics*, 13(5) :756–765, 2013. 38, 40

[78] Richard S Johnson and Klaus Biemann. Computer program (seqpep) to aid in the interpretation of high-energy collision tandem mass spectra of peptides. *Biological Mass Spectrometry*, 18(11) :945–957, 1989. 41

[79] Michael. Karas, Doris. Bachmann, and Franz. Hillenkamp. Influence of the wavelength in high-irradiance ultraviolet laser desorption mass spectrometry of organic molecules. *Analytical Chemistry*, 57(14) :2935–2939, 1985. 23, 29

[80] Uri Keich and William Stafford Noble. On the importance of well-calibrated scores for identifying shotgun proteomics spectra. *Journal of proteome research*, 14(2) :1147–1160, 2014. 42

[81] Andrew Keller, Jimmy Eng, Ning Zhang, Xiao-jun Li, and Ruedi Aebersold. A uniform proteomics ms/ms analysis platform utilizing open xml file formats. *Molecular systems biology*, 1(1), 2005. 38

[82] Attila Kertesz-Farkas, Uri Keich, and William Stafford Noble. Tandem mass spectrum identification via cascaded search. *Journal of proteome research*, 14(8) :3027–3038, 2015. 53

[83] Jung-whan Kim and Chi V Dang. Multifaceted roles of glycolytic enzymes. *Trends in biochemical sciences*, 30(3) :142–150, 2005. 9

[84] Sangtae Kim, Nitin Gupta, and Pavel A Pevzner. Spectral probabilities and generating functions of tandem mass spectra : a strike against decoy databases. *Journal of proteome research*, 7(8) :3354–3363, 2008. 42

[85] Aaron A Klammer, Christopher Y Park, and William Stafford Noble. Statistical calibration of the sequest xcorr function. *Journal of proteome research*, 8(4) :2106–2113, 2009. 42

[86] Andy T Kong, Felipe V Leprevost, Dmitry M Avtonomov, Dattatreya Mellacheruvu, and Alexey I Nesvizhskii. Msfragger : ultrafast and comprehensive peptide identification in mass spectrometry–based proteomics. *Nature methods*, 14(5) :513, 2017. 53, 85, 97, 112

[87] Henry Lam, Eric Deutsch, James Eddes, Jimmy Eng, Nichole King, Sara Yang, Jeri Roth, Lisa Kilpatrick, Pedatsur Neta, Steve Stein, et al. Spectrast : An open-source ms/ms spectramatching library search tool for targeted proteomics. In *Poster at 54th ASMS Conference on Mass Spectrometry*, 2006. 40

[88] Henry Lam, Eric W Deutsch, James S Eddes, Jimmy K Eng, Nichole King, Stephen E Stein, and Ruedi Aebersold. Development and validation of a spectral library searching method for peptide identification from ms/ms. *Proteomics*, 7(5) :655–667, 2007. 38

[89] Jing Li, Lisa J Zimmerman, Byung-Hoon Park, David L Tabb, Daniel C Liebler, and Bing Zhang. Network-assisted protein identification and data interpretation in shotgun proteomics. *Molecular systems biology*, 5(1) :303, 2009. 44

[90] Qunhua Li, Jimmy K Eng, Matthew Stephens, et al. A likelihood-based scoring method for peptide identification using mass spectrometry. *The Annals of Applied Statistics*, 6(4) :1775–1794, 2012. 37

[91] Yong Fuga Li, Randy J Arnold, Yixue Li, Predrag Radivojac, Quanhu Sheng, and Haixu Tang. A bayesian approach to protein inference problem in shotgun proteomics. *Journal of Computational Biology*, 16(8) :1183–1193, 2009. 43

[92] Bingwen Lu and Ting Chen. A suboptimal algorithm for de novo peptide sequencing via tandem mass spectrometry. *Journal of Computational Biology*, 10(1) :1–12, 2003. 37

[93] Bingwen Lu and Ting Chen. A suffix tree approach to the interpretation of tandem mass spectra : applications to peptides of non-specific digestion and post-translational modifications. *Bioinformatics*, 19(suppl_2) :ii113–ii121, 2003. 39

[94] Bin Ma, Kaizhong Zhang, Christopher Hendrie, Chengzhi Liang, Ming Li, Amanda Doherty-Kirby, and Gilles Lajoie. Peaks : powerful software for peptide de novo sequencing by tandem mass spectrometry. *Rapid communications in mass spectrometry*, 17(20) :2337–2342, 2003. 41

[95] Bin Ma, Kaizhong Zhang, and Chengzhi Liang. An effective algorithm for the peptide de novo sequencing from ms/ms spectrum. In *Annual Symposium on Combinatorial Pattern Matching*, pages 266–277. Springer, 2003. 37

[96] Chun Wai Manson Ma and Henry Lam. Hunting for unexpected post-translational modifications by spectral library searching with tier-wise scoring. *Journal of proteome research*, 13(5) :2262–2271, 2014. 53

[97] Matthias Mann, Peter Højrup, and Peter Roepstorff. Use of mass spectrometric molecular weight information to identify proteins in sequence databases. *Biological mass spectrometry*, 22(6) :338–345, 1993. 42

[98] Matthias Mann and Ole N Jensen. Proteomic analysis of post-translational modifications. *Nature biotechnology*, 21(3) :255, 2003. 21

[99] Matthias Mann and Matthias Wilm. Error-tolerant identification of peptides in sequence databases by peptide sequence tags. *Analytical chemistry*, 66(24) :4390–4399, 1994. 37

[100] Roger E Moore, Mary K Young, and Terry D Lee. Qscore : an algorithm for evaluating sequest database search results. *Journal of the American Society for Mass Spectrometry*, 13(4) :378–386, 2002. 43

[101] G.J. Mulder. Sur la composition de quelques substances animales. *Bulletin des Sciences Physiques et Naturelles en Néerlande : 104*, 1838. 9, 15

[102] Seungjin Na, Nuno Bandeira, and Eunok Paek. Fast multi-blind modification search through tandem mass spectrometry. *Molecular & Cellular Proteomics*, 11(4) :M111–010199, 2012. 53

[103] Seungjin Na and Eunok Paek. Software eyes for protein post-translational modifications. *Mass Spectrometry Reviews*, 34(2) :133–147. 108

[104] Alexey I Nesvizhskii and Ruedi Aebersold. Analysis, statistical validation and dissemination of large-scale proteomics datasets generated by tandem ms. *Drug Discovery Today*, 9(4) :173 – 181, 2004. 37

[105] Alexey I Nesvizhskii, Andrew Keller, Eugene Kolker, and Ruedi Aebersold. A statistical model for identifying proteins by tandem mass spectrometry. *Analytical chemistry*, 75(17) :4646–4658, 2003. 43

[106] Darryl JC Pappin, Peter Hojrup, and Alan J Bleasby. Rapid identification of proteins by peptide-mass fingerprinting. *Current biology*, 3(6) :327–332, 1993. 38, 42

[107] David N Perkins, Darryl JC Pappin, David M Creasy, and John S Cottrell. Probability-based protein identification by searching sequence databases using mass spectrometry data. *ELECTROPHORESIS : An International Journal*, 20(18) :3551–3567, 1999. 38, 41

[108] Thomas S Price, Margaret B Lucitt, Weichen Wu, David J Austin, Angel Pizarro, Anastasia K Yocum, Ian A Blair, Garret A FitzGerald, and Tilo Grosser. Ebp, a program for protein identification using multiple tandem mass spectrometry datasets. *Molecular & Cellular Proteomics*, 6(3) :527–536, 2007. 43

[109] Thomas S Price, Margaret B Lucitt, Weichen Wu, David J Austin, Angel Pizarro, Anastasia K Yocum, Ian A Blair, Garret A FitzGerald, and Tilo Grosser. Ebp, a program for protein identification using multiple tandem mass spectrometry datasets. *Molecular & Cellular Proteomics*, 6(3) :527–536, 2007. 44

[110] Smriti R Ramakrishnan, Christine Vogel, Taejoon Kwon, Luiz O Penalva, Edward M Marcotte, and Daniel P Miranker. Mining gene functional networks to improve mass-spectrometry-based protein identification. *Bioinformatics*, 25(22) :2955–2961, 2009. 44

[111] Smriti R Ramakrishnan, Christine Vogel, John T Prince, Rong Wang, Zhihua Li, Luiz O Penalva, Margaret Myers, Edward M Marcotte, and Daniel P Miranker. Integrating shotgun proteomics and mrna expression data to improve protein identification. *Bioinformatics*, 25(11) :1397–1403, 2009. 44

[112] Ivan Rayment, Hazel M Holden, Michael Whittaker, Christopher B Yohn, Michael Lorenz, Kenneth C Holmes, and Ronald A Milligan. Structure of the actin-myosin complex and its implications for muscle contraction. *Science*, 261(5117) :58–65, 1993. 9

[113] Jesse Rodriguez, Nitin Gupta, Richard D. Smith, and Pavel A. Pevzner. Does trypsin cut before proline ? *Journal of Proteome Research*, 7(1) :300–305, 2008. PMID : 18067249. 28

[114] Peter Roepstorff and Jan Fohlman. Proposal for a common nomenclature for sequence ions in mass spectra of peptides. *Biomedical mass spectrometry*, 11 11 :601, 1984. 32

[115] Rovshan G Sadygov, Hongbin Liu, and John R Yates. Statistical models for protein validation using tandem mass spectral data and protein amino acid sequence databases. *Analytical chemistry*, 76(6) :1664–1671, 2004. 43

[116] Tsuneaki Sakurai, T Matsuo, Hideo Matsuda, and Ituso Katakuse. Paas 3 : A computer program to determine probable sequence of peptides from mass spectrometric data. *Biological Mass Spectrometry*, 11(8) :396–399, 1984. 40

[117] Pablo C Sandoval, Dane H Slentz, Trairak Pisitkun, Fahad Saeed, Jason D Hoffert, and Mark A Knepper. Proteome-wide measurement of protein half-lives and translation rates in vasopressin-sensitive collecting duct cells. *Journal of the American Society of Nephrology*, 24(11) :1793–1805, 2013. 16

[118] Brian C Searle. Scaffold : a bioinformatic tool for validating ms/ms-based proteomic studies. *Proteomics*, 10(6) :1265–1269, 2010. 44

[119] Wenguang Shao and Henry Lam. Tandem mass spectral libraries of peptides and their roles in proteomics research. *Mass spectrometry reviews*, 36(5) :634–648, 2017. 38, 40

[120] Changyu Shen, Zhiping Wang, Ganesh Shankar, Xiang Zhang, and Lang Li. A hierarchical statistical model to assess the confidence of peptides and proteins inferred from tandem mass spectrometry. *Bioinformatics*, 24(2) :202–208, 2007. 44

[121] Ignat V Shilov, Sean L Seymour, Alpesh A Patel, Alex Loboda, Wilfred H Tang, Sean P Keating, Christie L Hunter, Lydia M Nuwaysir, and Daniel A Schaeffer. The paragon algorithm, a next generation search engine that uses sequence temperature values and feature probabilities to identify peptides from tandem mass spectra. *Molecular & Cellular Proteomics*, 6(9) :1638–1655, 2007. 52

[122] Marshall M Siegel and Norman Bauman. An efficient algorithm for sequencing peptides using fast atom bombardment mass spectral data. *Biological Mass Spectrometry*, 15(6) :333–343, 1988. 41

[123] Owen S Skinner and Neil L Kelleher. Illuminating the dark matter of shotgun proteomics. *Nature biotechnology*, 33(7) :717, 2015. 45

[124] Douglas J Slotta, Melinda A McFarland, and Sanford P Markey. Masssieve : panning ms/ms peptide data for proteins. *Proteomics*, 10(16) :3035–3039, 2010. 43

[125] Victor Spirin, Alexander Shpunt, Jan Seebacher, Marc Gentzel, Andrej Shevchenko, Steven Gygi, and Shamil Sunyaev. Assigning spectrum-specific p-values to protein identifications by mass spectrometry. *Bioinformatics*, 27(8) :1128–1134, 2011. 42

[126] Marina Spivak, Jason Weston, Daniela Tomazela, Michael J MacCoss, and William Stafford Noble. Direct maximization of protein identifications from tandem mass spectra. *Molecular & Cellular Proteomics*, 11(2) :M111–012161, 2012. 44

[127] Gordon Squires. Francis aston and the mass spectrograph. *J. Chem. Soc., Dalton Trans.*, pages 3893–3900, 1998. 23

[128] Hanno Steen and Matthias Mann. The abc's (and xyz's) of peptide sequencing. *Nature reviews Molecular cell biology*, 5(9) :699, 2004. 23

[129] David L Tabb, David B Friedman, and Amy-Joan L Ham. Verification of automated peptide identifications from proteomic tandem mass spectra. *Nature protocols*, 1(5) :2213, 2006. 36

[130] David L Tabb, W Hayes McDonald, and John R Yates. Dtaselect and contrast : tools for assembling and comparing protein identifications from shotgun proteomics. *Journal of proteome research*, 1(1) :21–26, 2002. 43

[131] David L Tabb, Anita Saraf, and John R Yates. Gutentag : high-throughput sequence tagging via an empirically derived fragmentation model. *Analytical chemistry*, 75(23) :6415–6421, 2003. 42, 51

[132] Koichi Tanaka, Hiroaki Waki, Yutaka Ido, Satoshi Akita, Yoshikazu Yoshida, Tamio Yoshida, and T Matsuo. Protein and polymer analyses up to m/z 100 000 by laser ionization time-of-flight mass spectrometry. *Rapid communications in mass spectrometry*, 2(8) :151–153, 1988. 29

[133] Stephen Tanner, Hongjun Shu, Ari Frank, Ling-Chi Wang, Ebrahim Zandi, Marc Mumby, Pavel A Pevzner, and Vineet Bafna. Inspect : identification of posttranslationally modified peptides from tandem mass spectra. *Analytical chemistry*, 77(14) :4626–4639, 2005. 51

[134] J Alex Taylor and Richard S Johnson. Sequence database searches via de novo peptide sequencing by tandem mass spectrometry. *Rapid communications in mass spectrometry*, 11(9) :1067–1075, 1997. 36

[135] Ravi Tharakan, Nathan Edwards, and David RM Graham. Data maximization by multipass analysis of protein mass spectra. *Proteomics*, 10(6) :1160–1171, 2010. 52

[136] The UniProt Consortium. Uniprot : the universal protein knowledgebase. *Nucleic Acids Research*, 45(D1) :D158–D169, 2017. 21, 28

[137] Dekel Tsur, Stephen Tanner, Ebrahim Zandi, Vineet Bafna, and Pavel A Pevzner. Identification of post-translational modifications via blind search of mass-spectra. In *Computational Systems Bioinformatics Conference, 2005. Proceedings. 2005 IEEE*, pages 157–166. IEEE, 2005. 53

[138] E. M. Widdowson, E. A. McCance, and C. M. Spray. The chemical composition of the human body. *Clinical Science*, 10 :113–125, 1951. 9

[139] Eric S Witze, William M Old, Katheryn A Resing, and Natalie G Ahn. Mapping protein post-translational modifications with mass spectrometry. *Nature methods*, 4(10) :798, 2007. 51

[140] Linfeng Wu and David K Han. Overcoming the dynamic range problem in mass spectrometry-based shotgun proteomics. *Expert review of proteomics*, 3(6) :611–619, 2006. 15

[141] Masamichi Yamashita and John B. Fenn. Electrospray ion source. another variation on the free-jet theme. *The Journal of Physical Chemistry*, 88(20) :4451–4459, 1984. 23

[142] Hao Yang, Hao Chi, Wen-Jing Zhou, Wen-Feng Zeng, Kun He, Chao Liu, Rui-Xiang Sun, and Si-Min He. Open-pnovo : de novo peptide sequencing with thousands of protein modifications. *Journal of proteome research*, 16(2) :645–654, 2017. 37

[143] Xiaoyu Yang, Vijay Dondeti, Rebecca Dezube, Dawn M Maynard, Lewis Y Geer, Jonathan Epstein, Xiongfong Chen, Sanford P Markey, and Jeffrey A Kowalak. Dbparser : web-based software for shotgun proteomic data analyses. *Journal of proteome research*, 3(5) :1002–1008, 2004. 43

[144] John R Yates, Scott F Morgan, Christine L Gatlin, Patrick R Griffin, and Jimmy K Eng. Method to compare collision-induced dissociation spectra of peptides : potential for library searching and subtractive analysis. *Analytical chemistry*, 70(17) :3557–3565, 1998. 38

[145] John R Yates, Stephen Speicher, Patrick R Griffin, and Tim Hunkapiller. Peptide mass maps : a highly informative approach to protein identification. *Analytical biochemistry*, 214(2) :397–408, 1993. 42

[146] Ding Ye, Yan Fu, Rui-Xiang Sun, Hai-Peng Wang, Zuo-Fei Yuan, Hao Chi, and Si-Min He. Open ms/ms spectral library search to identify unanticipated post-translational modifications and increase spectral identification rate. *Bioinformatics*, 26(12) :i399–i406, 2010. 53

[147] Fengchao Yu, Ning Li, and Weichuan Yu. Pipi : Ptm-invariant peptide identification using coding method. *Journal of proteome research*, 15(12) :4423–4435, 2016. 53

[148] Bing Zhang, Matthew C Chambers, and David L Tabb. Proteomic parsimony through bipartite graph analysis improves accuracy and transparency. *Journal of proteome research*, 6(9) :3549–3557, 2007. 43

[149] Zhongqi Zhang. Prediction of collision-induced-dissociation spectra of peptides with post-translational or process-induced modifications. *Analytical chemistry*, 83(22) :8642–8651, 2011. 38

[150] Zhongqi Zhang and Bhavana Shah. Prediction of collision-induced dissociation spectra of common n-glycopeptides for glycoform identification. *Analytical chemistry*, 82(24) :10194–10202, 2010. 38

UNIVERSITE BRETAGNE LOIRE / MATHSTIC

UNIVERSITÉ DE NANTES

Titre : Découverte de modifications chimiques portées par les protéines : identification rapide de jeux de spectres de masse sans filtre de masse.

Mot clés : Informatique, Algorithmique, Bio-informatique, Protéomique, Spectrométrie de masse

Resumé : Les protéines remplissent de nombreuses fonctions indispensables au développement et à la survie des organismes vivants. Leur étude est primordiale pour comprendre les mécanismes biologiques au sein des cellules. La technique la plus utilisée pour caractériser les protéines est la spectrométrie de masse en tandem qui produit des dizaines de milliers de spectres expérimentaux par analyse. Pour interpréter ces spectres –c'est-à-dire retrouver et caractériser les protéines présentes dans le mélange analysé– la méthode la plus courante consiste à les comparer à une banque pouvant contenir des centaines de milliers de spectres "modèles". Pour accélérer l'analyse, la majorité des approches limitent le nombre de comparaisons, limitation suspectée à l'origine de difficultés d'identification. En effet, à l'issue d'une expérience de spectrométrie de masse, 50 à 75% des spectres demeurent aujourd'hui non identifiés.

Pour résoudre ces difficultés, nous avons conçu une méthode pour interpréter les spectres de masse sans filtre de la banque. Cette méthode repose sur une structure de données SpecTrees qui permet une représentation compacte du nombre de masses partagées entre chaque paire de spectres, et sur des algorithmes qui manipulent efficacement l'information contenue dans cette structure de donnée. L'ensemble constitue le logiciel SpecOMS diffusé à la communauté. Nous discutons l'usage des différents paramètres de notre méthode et la comparons à un autre algorithme performant.

Title : Discovery of chemical modifications carried by proteins : rapid identification of mass spectrometry datasets without the use of a mass filter.

Keywords : Computer Science, Algorithmics, Bio-informatics, Protéomics, Mass Spectrometry

Abstract : Proteins fulfill numerous functions mandatory to the growth and the survival of living organisms. The study of those proteins is paramount in order to understand biological mechanisms inside the cells. The most widely used technique to characterize proteins is tandem mass spectrometry, which produces tens of thousands of experimental mass spectra per analysis. To interpret those spectra -which leads to finding and characterizing the proteins from the analyzed mixture- the most common methodology consists in comparing them to a database that can contain hundreds of thousands of "model" spectra. To speed up the analysis, the majority of the approaches reduce the number of comparisons, a limitation supposed to be the cause of identification difficulties. Indeed, the output of a mass spectrometry analysis nowadays still exhibit 50 to 75% of unidentified spectra.

In order to solve this problem, we conceived a method to interpret mass spectra without filtering the database. This method relies on a specific datastructure, SpecTrees, which allows a compact representation of the number of shared masses between each pair of spectra, alongside algorithms manipulating efficiently the information contained in this datastructure. Altogether, the datastructure and algorithms compose SpecOMS, a software available to the community. We discuss the use of different parameters for our method and compare it to another efficient algorithm.