# Peptide identification by tandem mass spectrometry : a tag-oriented open-modification search method

Hernandez, Patricia

UNIVERSITE DE GENEVE

Département d'informatique
Institut Suisse de Bioinformatique

FACULTE DES SCIENCES

Professeur R.D. Appel
Docteur R. Gras

---

# Peptide Identification by Tandem Mass Spectrometry: a Tag-Oriented Open-Modification Search Method

THESE

présentée à la Faculté des sciences de l'Université de Genève

pour obtenir le grade de Docteur ès sciences, mention bioinformatique

par

**Patricia Hernandez**

de

Torny-le-Grand (FR)

Thèse N° 3698

Genève
2005

**FACULTÉ DES SCIENCES**

**UNIVERSITÉ DE GENÈVE**

***Doctorat ès sciences***
***mention bioinformatique***

Thèse de *Madame Patricia HERNANDEZ*

intitulée :

# "Peptide Identification by Tandem Mass Spectrometry : a Tag-Oriented Open-Modification Search Method"

La Faculté des sciences, sur le préavis de Messieurs R. D. APPEL, professeur ordinaire et directeur de thèse (Département d'informatique), R. GRAS, docteur et co-directeur de thèse (Institut Suisse de Bioinformatique – Genève, Suisse et INRIA/IRISA – Rennes, France), C. PELLEGRINI, professeur ordinaire (Département d'informatique), et R. AEBERSOLD, professeur (Eidgenössische Technische Hochschule Zürich – Institut für Molekulare Systembiologie – Zürich, Suisse), autorise l'impression de la présente thèse, sans exprimer d'opinion sur les propositions qui y sont énoncées.

Genève, le 15 décembre 2005

**Thèse - 3698 -**

**Le Doyen, Pierre SPIERER**

N.B.- La thèse doit porter la déclaration précédente et remplir les conditions énumérées dans les "Informations relatives aux thèses de doctorat à l'Université de Genève".

**Nombre d'exemplaires à livrer par colis séparé à la Faculté : - 7 -**

# Abstract

Dans la majorité des projets de recherche en Protéomique, il faut, à un moment où un autre, déterminer l'identité des protéines présentes dans l'échantillon biologique étudié. Lorsque l'identification porte sur un grand nombre de protéines, il est essentiel de pouvoir disposer de techniques automatisées et fournissant des résultats non ambigus. De nos jours, la méthode la plus répandue est la corrélation de spectres de protéines ou peptides (obtenus par spectrométrie de masse) avec des séquences protéiques théoriques répertoriées dans des banques de données. Bien que de nombreux développements aient été réalisés dans ce domaine, l'identification de protéines par spectrométrie de masse rencontre un certain nombre de difficultés liées, entre autres, à la présence de modifications non attendues sur les peptides analysés.
Notre travail avait donc deux objectifs: fournir un algorithme d'identification qui soit robuste à la présence de modifications non prévues sur les peptides et, pour autant que les spectres contiennent suffisamment d'information, préciser la position sur la séquence peptidique et la nature des modifications présentes. Notre approche, Popitam peut être définie comme une méthode d'identification à "modification ouverte". La taille de l'espace de recherche étant agrandie, nous nous sommes également concentrés sur la conception de fonctions de score optimisées par programmation génétique.

Protein identification by tandem mass spectrometry (MS/MS) is key to most proteomics projects and has been widely explored in bioinformatics research. Obtaining good and trustful identification results has important implications for biological and clinical work. Although well matured, automated identification of proteins from MS/MS data still faces a number of obstacles due to the complexity of the proteome or to procedural issues of mass spectrometry data acquisition. Expected or unexpected modifications of the peptide sequences, polymorphisms, errors in databases, missed or non-specific cleavages, unusual fragmentation patterns and co-eluting peptides are many pitfalls for identification algorithms. A lot of research work has been carried out in recent years; this has given rise to new strategies designed to handle a number of these issues. In this work, we describe a new approach for identifying and characterizing modified peptides using MS/MS data. Our method, Popitam, does not require to give a list of suspected modifications and is therefore classified as an "open-modification search" algorithm. The size of the search space is much greater with such an approach, so we also focused our work on the conception of optimized scoring functions. We defined a set of subscores describing different aspects of similarity between an MS/MS spectrum and a theoretical sequence. Then we used Genetic Programming to explore the different ways of combining these subscores, so that the built scoring function can efficiently identify and discriminate the correct peptide amongst a list of candidate peptide sequences.

# ACKNOWLEDGMENTS

Cette thèse n'aurait pas été possible sans le support, l'aide et l'amitié de certaines personnes. Ron Appel m'a accueillie dans son groupe. Je tiens à le remercier tout particulièrement pour la confiance qu'il m'a accordée. A de nombreuses occasions, il m'a encouragée à présenter et publier mon travail. Je suis redevable à Robin Gras pour les nombreuses idées dont il a nourri la thèse et pour le support moral qu'il m'a apporté. Je remercie également les Professeurs Christian Pellegrini et Ruedi Aebersold, membres du jury, pour leurs remarques constructives et leur enthousiasme.
Je dois cette thèse à mes collègues. Céline Hernandez a travaillé sur l'optimisation de Popitam et sur l'interface web; Marc Tuloup a fourni une aide considérable ainsi que des lignes de code, entre autre pour le module de digestion; David Hernandez a fourni la classe de l'arbre des séquences, et Yoann Mescam celle de l'arbre des suffix; Julien Frey a implémenté le module de Programmation Génétique; Alexander Scherl a produit plusieurs sets de données MS/MS; Pierre-Alain a répondu avec patience à de nombreuses questions. D'autres encore, comme Amos Bairoch, Garry Corthals, Khaled Mostaguir, Markus Müller et Nadine Zangger, ont contribué à mon travail par leur aide, leur curiosité, leur intérêt et leur enthousiasme. Notamment, Amos a trouvé le livre "Monsieur Hippopotame", de Tanikawa Shuntarô, duquel ont été adaptées les illustrations d'hippopotame.
L'écriture de ce document a été une tâche difficile, et plusieurs collègues et membres de la famille ont été mis à contribution pour sa relecture. Leur aide a été précieuse.
Enfin, je remercie tout particulièrement mes parents, pour leur affection et pour m'avoir donné l'opportunité de réaliser de longues études. Merci également à Luc, pour son amour, et pour avoir supporté mes humeurs instables les quelques semaines précédant la soutenance.

This thesis wouldn't have been possible without the support, help and friendship of many people. Ron Appel accepted me in his group. He undoubtedly trusted me more than I did myself. In many occasions, he urged me to publish and present my work. I am indebted to Robin Gras, who largely contributed to the thesis with many fruitful ideas and provided an appreciable amount of moral support. I thank the Professors Christian Pellegrini and Ruedi Aebersold, members of the jury, for their constructive remarks and their enthusiasm.
I owe this thesis to my colleagues. Céline Hernandez worked on the optimization of Popitam and on the web interface; Marc Tuloup provided great help and code, notably for the digestion module; David Hernandez provided the sequence tree class and Yoann Mescam the suffix tree class; Julien Frey implemented the Genetic Programming module; Alexander Scherl produced several sets of MS/MS data; Pierre-Alain gave answers to many questions. Others, like Amos Bairoch, Garry Corthals, Khaled Mostaguir, Markus Müller and Nadine Zangger, contributed to my work with their support, curiosity, interest and enthusiasm. Notably, Amos found the book "Monsieur Hippopotame" by Tanikawa Shuntarô, from where the hippopotamus illustrations were taken and adapted. Writing this document in English has been a difficult work and several colleagues and family members were asked to reread one or another chapter. Their help has been very precious.
Special thanks go to my parents, for their affection and for having given me the opportunity to make long studies, and to Luc, for his love and for having supported my instable mood the few weeks before the thesis defense.

# FRENCH SUMMARY

## INTRODUCTION

Les protéines sont des constituants essentiels des organismes et leurs fonctions sont aussi nombreuses que variées. Elles régulent l'expression des gènes ainsi que l'activité d'autres protéines, transportent des molécules à l'intérieur ou à l'extérieur des cellules, délivrent des messages et permettent aux muscles de se contracter. Elles ont un rôle de structure, de mémoire immunologique et peuvent même être utilisées comme réserves d'énergies. Les protéines sont codées dans les gènes. Leur synthèse comprend la transcription des gènes en ARN messager, puis la traduction des molécules d'ARN en séquence d'acides aminés. Lors de ce processus, les acides aminés sont liés les uns aux autres et forment une chaîne peptidique. Après plusieurs étapes de maturation, les protéines sont dirigées dans leur compartiment cible où elles remplissent leur fonction, avant d'être dégradées et recyclées.
Lors de leur maturation, les protéines subissent des modifications post-traductionnelles (PTMs), incluant des coupures de la chaîne peptidique en des endroits spécifiques, la formation de liaisons (ponts disulfures) intra ou inter-chaînes, et l'ajout de molécules (par exemple un groupe acétate ou phosphate, un sucre, un lipide) sur des acides aminés. Ces modifications permettent à la protéine de devenir complètement fonctionnelle. De nombreux processus biologiques en sont dépendants, comme le repliement de la protéine, la régulation de son activité, sa capacité d'interagir avec d'autres protéines, sa localisation dans la cellule et sa durée de vie.

Dans la majorité des projets de recherche sur les protéines, il faut, à un moment où un autre, déterminer l'identité des protéines présentes dans l'échantillon biologique étudié. Lorsque l'identification porte sur un grand nombre de protéines, il est essentiel de pouvoir disposer de techniques automatisées et fournissant des résultats non ambigus. De nos jours, la méthode la plus répandue est la corrélation de spectres de protéines ou peptides (obtenus par spectrométrie de masse) avec des séquences protéiques théoriques répertoriées dans des banques de données. Bien que de nombreux développements aient été réalisés dans ce domaine, l'identification de protéines par spectrométrie de masse rencontre un certain nombre de difficultés liées, entre autres, à la nature complexe du protéome.

Notre travail a consisté à créer et implémenter une nouvelle méthode d'identification et de caractérisation de peptides portant des modifications post-traductionnelles non attendues, voire inconnues, à partir de données de spectrométrie de masse en tandem (MS/MS). L'enjeu était particulièrement motivant. En effet, déterminer la présence de modifications et mieux encore, préciser leur emplacement sur la séquence protéique ainsi que leur type, est d'un intérêt majeur pour la recherche en biologie à cause de l'implication des modifications dans de nombreux processus cellulaires, physiologiques et pathologiques.

Notre travail avait donc deux objectifs: fournir un algorithme d'identification qui soit robuste à la présence de modifications non prévues sur les peptides et, pour autant que les spectres contiennent suffisamment d'information, préciser la position sur la séquence peptidique et la nature des modifications présentes.

# TECHNIQUES DE PROTEOMIQUE

La protéomique se définit comme la science qui étudie le contenu en protéines d'échantillons biologiques (par exemple, un tissu). La préparation des échantillons inclut la cassure des parois cellulaires, l'extraction d'impuretés (comme l'ADN ou les lipides), la dénaturation des protéines, ainsi que des étapes de fractionnements et de séparations, jusqu'à obtenir, dans certains cas, une seule –ou quelques- espèces de protéines à analyser. Ces dernières sont ensuite clivées en peptides par l'action d'une enzyme de digestion et analysées par spectrométrie de masse.

## Séparation de protéines

Il existe deux classes majeures de techniques de séparation de protéines en protéomique: la chromatographie et l'électrophorèse. Le principe, pour l'une comme pour l'autre, est une migration différentielle des molécules dans un milieu gazeux, liquide ou semi-liquide (gel).

En chromatographie, la force de migration est mécanique. Une solution (phase mobile) contenant les analytes (protéines ou peptides) circule de manière continue dans une colonne remplie avec un support (phase stationnaire) dont les propriétés physico-chimiques se caractérisent par une certaine affinité avec les analytes. En modifiant de manière graduelle les caractéristiques du buffer dans la colonne, les analytes sont soit retenus sur le support, soit libérés et soumis au flux qui les dirige vers la sortie. La nature de l'interaction entre les analytes et le support est variable. En chromatographie hydrophobe, les protéines sont séparées selon leur degré d'hydrophobicité. La colonne est remplie avec des billes contenant des chaînes hydrophobes sur lesquelles les protéines viennent se coller. Sous l'action d'un détergent, les protéines se solubilisent et, selon leur degré d'hydrophobicité, se décollent plus ou moins vite, quittant ainsi séparément la colonne. En chromatographie ionique, les protéines sont séparées selon leurs propriétés électriques. Dans ce cas, la phase stationnaire est une résine chargée négativement. Les protéines, chargées positivement, sont introduites dans la colonne et se lient à la résine. En modifiant progressivement le pH de la solution, les protéines se libèrent de manière différentielle, suivant leur charge. En chromatographie par affinité, ce sont des interactions spécifiques protéines-ligands qui sont exploitées (par exemple, une enzyme et son substrat). Enfin, en chromatographie par exclusion, la séparation s'effectue en fonction de la taille des protéines. La colonne est remplie d'un support poreux dans lequel les protéines pénètrent ou non, selon leur taille. Les molécules les plus grandes sont les premières à quitter la colonne alors que les plus petites sont retenues plus longtemps.

En électrophorèse, la force de migration provient d'un champ électrique entre deux électrodes. Ainsi, en électrophorèse SDS-PAGE, les protéines dénaturées sont placées à l'extrémité d'un gel de polyacrylamide dont la taille des pores forme un gradient continu entre les deux extrémités. Le potentiel électrique provoque la migration des analytes d'une extrémité du gel vers l'autre. Selon leur taille, les analytes migrent plus ou moins vite et se dispersent dans la longueur du gel. En électrophorèse IEF, le gel contient un gradient de pH. Les protéines qui le traversent s'arrêtent à

l'endroit précis où leur charge est neutre. En utilisant des gels rectangulaires plutôt que longilignes, il est possible de combiner les deux technologies: les protéines sont séparées dans une première dimension selon leur point isoélectrique, et dans une deuxième dimension selon leur poids moléculaire. La séparation produit une constellation de points (plusieurs milliers par gel), chacun d'eux représentant une ou quelques protéines différentes.

## Spectrométrie de masse

La technique la plus utilisée en Protéomique pour identifier les protéines est la spectrométrie de masse. Un spectromètre est composé de deux éléments principaux: une source et un analyseur. La source sert à ioniser les peptides, qui sont ensuite transmis à l'analyseur. Il existe plusieurs types de sources. Les plus répandues sont la source MALDI ("matrix-assisted laser desorption ionization »), dans laquelle les ions sont créés par l'action d'un faisceau laser, et la source ESI (« electrospray ionization »), dans laquelle les ions sont formés par un effet de répulsion électrostatique due à un champ électrique. De même que pour les sources, il existe plusieurs types d'analyseurs, dont les « ion-traps » (IT), les « time-of-flights » (TOF), les « quadrupoles » (Q) ou encore les « fourier transforms» (FT). Les analyseurs permettent de mesurer la valeur masse/charge des peptides ionisés, produisant un spectre MS, caractérisé par un ensemble de pics d'intensités variables. Un spectre MS correspond donc à une protéine, et les pics représentent les valeurs masse/charge de ses peptides. L'intensité des pics est fonction du nombre de peptides détectés à une certaine valeur masse/charge. Puisque la composition en pics est spécifique pour une protéine, les spectres sont appelés « peptide mass fingerprint » (PMF) pour « empreinte par masses de peptides ». L'identification d'une protéine se fait en corrélant son spectre MS avec des spectres virtuels construits à partir de séquences protéiques provenant de banques de données (identification par PMF).

Lorsque l'échantillon analysé contient plusieurs, voir un grand nombre de protéines, la digestion résulte en une mixture complexe de peptides. Dans un tel cas, ainsi que lorsque l'on veut obtenir de l'information sur la séquence de la protéine, une deuxième étape de MS vient se superposer à la première. On parle alors de « spectrométrie de masse en tandem » (MS/MS). Des exemples de configurations de spectromètres sont le « time-of-flight/time-of-flight » (TOF-TOF), le «quadrupole/time-of-flight » (Q-TOF) ou encore le « quadrupole/ion trap » (Q-IT). La première étape de spectrométrie permet d'isoler les peptides selon leur valeur masse/charge. Puis, chacun leur tour, les peptides sont fragmentés. La fragmentation se fait généralement par collision avec un gaz rare et est aléatoire, la coupure ayant lieu plus ou moins entre les acides aminés et conduisant à des fragments ioniques. Une même position de fragmentation sur la séquence (par exemple entre le deuxième et le troisième acide aminé) peut produire plusieurs pics différents, selon l'état de charge des fragments obtenus, leur type ionique (par exemple N-terminal et C-terminal) et d'éventuelles pertes de molécules (comme des molécules d'eau ou des ions ammoniums). Les valeurs masse/charge des fragments obtenus sont mesurées et produisent un spectre MS/MS appelé «Peptide Fragment Fingerprint » (PFF). Cette fois-ci, le spectre correspond à un peptide (appelé peptide précurseur), et les pics représentent la valeur masse/charge de fragments de peptides. De la même manière que pour les spectres MS, un peptide ayant produit un spectre MS/MS peut être identifié par corrélation avec des séquences peptidiques théoriques (identification par PFF). Les peptides identifiés mènent alors à une liste de protéines identifiées.

# IDENTIFICATION MS

## Identification par « peptide mass fingerprinting » (PMF)

L'identification par PMF implique la digestion de la protéine d'intérêt (de préférence purifiée) en peptides (les coupures se font en des endroits spécifiques, selon l'enzyme utilisée), la mesure des masses des peptides par spectrométrie de masse et la comparaison du spectre MS obtenu avec des spectres virtuels. Ces derniers sont construits à partir de protéines candidates filtrées parmi l'ensemble de séquences protéiques répertoriées dans une banque de donnée choisie et digérées de manière virtuelle. Ainsi, si la trypsine a été utilisée pour digérer les protéines de l'échantillon biologique, la digestion virtuelle coupera les séquences après chaque lysine et arginine, sauf si ces dernières sont suivies d'une proline. Selon les paramètres choisis, la digestion pourra permettre de « sauter » une lysine ou arginine (« missed-cleavage »). La sélection des protéines candidates se fait généralement sur la base d'attributs spécifiques connus, tels que l'espèce ayant produit l'échantillon, la masse totale de la protéine ou son point isoélectrique. La similarité entre le spectre expérimental et les spectres virtuels est quantifiée par une fonction de score. C'est la protéine candidate qui obtient le meilleur score qui est présumée représenter la protéine ayant produit le spectre. Si tous les scores se situent en dessous d'une valeur seuil, le spectre est considéré comme non-identifié. Une situation comme celle-ci peut s'expliquer par des différences inattendues entre la protéine expérimentale et son double théorique, ou par l'absence de ce dernier de la banque de donnée (protéine inconnue).

# IDENTIFICATION MS/MS

L'identification de protéines à partir de spectres MS/MS présente plusieurs avantages. Premièrement, puisque plusieurs spectres MS/MS sont généralement obtenus pour chaque protéine, l'identification est plus robuste et moins équivoque. Elle est plus robuste, car il n'est pas nécessaire d'identifier tous les peptides de la protéine pour suspecter sa présence dans l'échantillon; et elle est moins équivoque, car l'identification de plusieurs peptides pour une protéine donnée tend à confirmer le résultat de l'identification. Un second avantage est que l'analyse par spectrométrie de masse en tandem permet d'analyser des échantillons contenant des mélanges complexes de protéines, contrairement à l'identification par PMF qui nécessite un niveau de purification élevé. Si les étapes de séparations s'en trouvent simplifiées, il faut, par contre, ajouter une étape de compilation de résultats, lors de laquelle une liste de protéines identifiées est construite à partir de la liste des peptides identifiés. Or cette étape n'est pas toujours simple, car différentes protéines peuvent produire des peptides identiques. Enfin, une caractéristique intéressante des spectres MS/MS est qu'ils contiennent de l'information sur la séquence du peptide précurseur, la séquence d'acides aminés pouvant être inférée à partir d'écarts de masses entre pics (séquençage *de novo*).

## Identification par séquençage de novo

Le séquençage *de novo* est particulièrement intéressant lorsque la séquence de la protéine d'origine est absente des banques de données ou lorsqu'il y a suspicion de remplacements d'acides aminés (mutations ou erreurs dans la banque de données) entre le peptide ayant produit le spectre et son représentant théorique dans la banque de données. La plupart des méthodes de séquençage *de novo* commencent par structurer le spectre en un graphe dont les nœuds représentent des masses de

fragments potentiels. Tous les nœuds ayant un écart de masse équivalent à la masse d'un (ou plusieurs) acides aminés sont connectés par des ponts. Deux nœuds particuliers sont créés artificiellement pour inclure dans le graphe la masse de la séquence vide et celle de la séquence complète (masse du précurseur). L'ensemble des chemins permettant de relier ces deux nœuds représente l'ensemble des séquences d'acides aminés qu'il est possible d'inférer à partir de la distribution des pics dans le spectre et ayant une masse totale similaire à celle du précurseur. Le problème du séquençage *de novo* est de repérer, parmi toutes ces séquences, celle qui correspond au peptide précurseur et donc est à l'origine du spectre. En séquençage *de novo*, les modifications éventuelles d'acides aminés sont prises en compte en ajoutant des résidus à masses modifiées dans l'alphabet des acides aminés possibles. Afin de ne pas augmenter de manière exagérée le niveau de connexion du graphe, et donc le nombre de chemins à explorer et évaluer, il est nécessaire de restreindre le nombre des résidus modifiés à quelques unités.

## Identification par «peptide fragment fingerprinting» (PFF)

Mais lorsque l'on s'attend à ce que la séquence du précurseur soit présente dans la banque de données, une approche par comparaison de spectres (PFF) est généralement préférée. Toute méthode d'identification par PFF partage les mêmes étapes-clés, à savoir le filtrage des peptides candidats, la construction des spectres virtuels et la comparaison de spectres. Les différences se situent dans la manière d'effectuer chacune de ces tâches.

Choix des peptides candidats

Comme pour l'identification par PMF, les séquences protéiques sont digérées virtuellement afin de produire des peptides. L'utilisation de filtres basés sur des attributs connus permet de limiter la comparaison à une sous-partie des peptides, ce qui réduit d'une part le temps de calcul et d'autre part le nombre d'identifications faux-positives (peptides produisant par chance des scores élevés). Généralement, on utilise comme filtre l'espèce ayant produit l'échantillon et la similarité entre la masse du précurseur et celle des peptides candidats. Ce type de filtrage est très efficace, mais il présente le désavantage d'écarter le peptide correct si sa masse ne correspond pas, pour une raison ou une autre, à celle du précurseur. Or, une telle situation est loin d'être rare. En voici quelques exemples:

a) la masse du précurseur n'est pas suffisamment précise et l'erreur paramétrée trop petite

b) le précurseur porte une modification post-traductionnelle ou est muté

c) la séquence théorique contient une erreur

d) les sites de coupure ne suivent pas les règles établies (par exemple, présence de deux « missed-cleavages », ou peptide tronqué)

Il est possible de remplacer –ou complémenter- le filtrage avec de courtes séquences, appelées « tags », extraites des spectres par séquençage *de novo*. Les peptides issus de la digestion virtuelle sont alors sélectionnés selon leur séquence (ils doivent contenir le tag) et non plus uniquement selon leur masse.

Prédiction de spectres

Les spectres virtuels sont construits en simulant le processus de fragmentation sur les séquences des peptides théoriques. Les paramètres généralement utilisés sont la séquence en acides aminés du peptide, l'état de charge du précurseur, le type de spectromètre utilisé et des hypothèses ioniques (décrivant le type ionique d'un fragment, son état de charge et d'éventuelles pertes de molécules). Les prédictions concernent d'une part l'emplacement des pics et d'autre part leur intensité.

Mesure de la similarité entre un spectre expérimental et un spectre théorique

La similarité entre deux spectres se mesure à partir du nombre de pics qu'ils ont en commun. Cela revient à compter les pics apparaissant dans les deux spectres, étant donné un seuil d'erreur autorisé. Deux mesures couramment utilisées sont la corrélation-croisée, qui permet de mesurer la similarité de deux signaux continus ou discrets, et le produit vectoriel. Pour ce dernier, les spectres sont représentés comme des vecteurs dans un espace N-dimensionel (N est le nombre de pics en communs), la direction et la longueur des vecteurs étant déterminées par les valeurs masse/charge et les intensités des pics. Si les spectres sont égaux, l'angle entre les vecteurs est nul. S'ils sont très différents, l'angle approche les 90 degrés.

Afin d'exploiter de manière plus intensive l'information présente dans les spectres, et donc de réduire le nombre d'identifications faux-positives, les méthodes d'identifications incluent souvent d'autres paramètres dans la fonction de score, comme l'erreur observée entre les fragments appariés, la présence de séries ioniques (fragmentation de positions successives sur la séquence du précurseur), la composition en acide aminés du peptide théorique, le nombre de « missed-cleavages », etc.

La présence de modifications du peptide précurseur peut être, du moins dans une certaine mesure, prise en compte par les méthodes PFF pour autant qu'elles soient prévisibles. Il suffit de générer des peptides modifiés à partir des séquences de la banque de donnée et d'une liste de modifications possibles. Plus la liste est fournie, plus l'augmentation du nombre de peptides possibles est importante. Par exemple, dans une banque de donnée comprenant 800'000 peptides tryptiques (ce qui correspond à environ 10'000 protéines), la prise en compte d'un seul type de modification pouvant survenir sur les acides aminés D, K, N, P, F et Y (comme l'hydroxylation) mène à la création d'un total de 7.4 millions de peptides si l'on autorise 0, 1 ou 2 évènements de modification par peptide. Si tous les acides aminés peuvent être modifiés, ce nombre monte à près de 45 millions. Il ne reste plus qu'à imaginer le nombre de peptides différents si plusieurs types de modifications sont pris en compte pour chaque acide aminé. Heureusement, le filtrage des peptides candidats par rapport à la masse du précurseur permet de réduire fortement le nombre de peptides finalement analysés. Néanmoins, il est tout de même nécessaire de restreindre le nombre de modifications différentes à quelques unités afin de rester dans des temps de calcul raisonnables.

# METHODES D'IDENTIFICATION A "MODIFICATIONS OUVERTES"

Mais qu'en est-il lorsque le peptide précurseur porte une modification qui n'est pas répertoriée dans la liste des modifications possibles ? Les méthodes de séquençage *de novo* se trouvent face à un graphe dont le chemin correspondant à la séquence du précurseur est coupé en deux, puisque le pont représentant la masse de l'acide aminé modifié n'existe pas. Deux cas peuvent se présenter: soit les deux sous-parties de chemin ne sont pas connectées du tout, et il est alors impossible de parcourir l'ensemble du chemin (du premier au dernier noeud). Dans ce cas, l'algorithme de séquençage *de novo* ne peut aboutir à un résultat valide. Soit les deux sous-parties sont connectées par un chemin alternatif empruntant un ou plusieurs ponts n'appartenant pas au chemin correct. Dans ce cas, l'algorithme trouve une séquence proche de celle du peptide d'origine, mais non exacte sur la partie du chemin alternatif.

Pour ce qui est des méthodes de PFF, la présence d'une modification non-attendue a deux conséquences. Premièrement, la masse du précurseur est modifiée; il n'est donc pas possible de filtrer les peptides candidats en utilisant cette dernière. Deuxièmement, la modification affecte la position des pics (en moyenne, la moitié des pics sont décalés vers la gauche si la modification provoque une perte de masse, vers la droite sinon), ce qui a pour conséquence de réduire les scores de comparaison.

Une première solution à ce problème consiste à utiliser un filtre basé sur des tags à la place de la masse du précurseur. De cette manière, le nombre de peptides candidats analysés est fortement réduit, et même si un certain nombre d'appariements de pics est perdu lors de la comparaison, le score suffit encore à départager le bon peptide des autres candidats. Le point faible de cette méthode est que ses atouts résident dans le filtrage plutôt que dans la procédure de comparaison, puisqu'elle n'essaye pas d'apparier des pics de fragments modifiés avec des pics de fragments non-modifiés.
Une deuxième méthode étend le concept de PFF en autorisant la présence de décalages de masses entre pics appariés. Ceux-ci ne sont plus appariés selon leur valeur absolue, mais selon l'écart entre leur valeur masse/charge et celle des pics précédemment appariés.
Enfin, une troisième méthode utilise des séquences *de novo* (séquences complètes produites par un algorithme de séquençage *de novo* et donc, contenant forcément un chemin alternatif). L'alignement entre la séquence *de novo* et les séquences théoriques est initié par appariement sur une courte région non ambiguë, puis l'alignement est étendu vers la gauche et la droite en se basant sur la correspondance des masses des acides aminés (simples ou en combinaison) tout en autorisant si nécessaire des acides aminés non-appariés. Les écarts de masses relevés aux endroits non-appariés sont expliqués par des modifications possibles.

# ALGORITHME DE POPITAM

Popitam est un outil d'identification de peptides à partir de spectres MS/MS. Il ressemble aux méthodes de PFF parce qu'il utilise une banque de donnée de séquences pour guider l'interprétation du spectre. Mais il emprunte aux méthodes de séquençage *de novo* la structure de graphe. Ce dernier est utilisé pour extraire du spectre tous les tags qui représentent des sous-séquences du peptide candidat en train d'être analysé. Les tags sont ensuite combinés en respectant des règles de compatibilité, conduisant à la création de scénarios d'interprétation composés d'un ou plusieurs tags séparés par des trous (appelés gaps). Puis Popitam évalue si les gaps proviennent d'un manque d'information dans le spectre (dans ce cas, ils sont dénommés *lackGaps*), ou s'ils sont dus à une modification ou à une différence de séquence (dans ce cas, ils sont dénommés *modGaps*). Pour chaque peptide candidat, plusieurs scénarios sont créés et évalués. Le candidat avec le meilleur scénario est proposé comme résultat de l'identification. Les paragraphes qui suivent décrivent les trois processus-clés de l'algorithme, à savoir l'extraction de tag, la création des scénarios et leur évaluation.

## Filtrage des peptides de la banque de donnée

Popitam utilise, pour le moment, quatre types de filtres. Le premier est taxonomique. Le second est basé sur les règles de coupures de l'enzyme utilisée pendant la digestion de l'échantillon. Le troisième est basé sur l'intervalle de masse autorisé pour les modifications (cet intervalle peut être de

plusieurs centaines de Daltons). Enfin, un quatrième filtre est basé sur une liste restreinte de protéines. Seuls des peptides provenant des protéines spécifiées sont analysés. Nous prévoyons d'ajouter prochainement un filtre basé sur une liste de tags.

## Construction du graphe

La construction du graphe nécessite de ré-exprimer les valeurs masse/charge lues dans le spectre en masses de fragments N-terminaux « standards », ou PRMs (Prefix Residue Masses). La procédure revient à énoncer un certain nombre d'hypothèses ioniques pour chaque pic. Au plus, l'une de ces hypothèses sera correcte, toutes les autres donnant forcément des PRMs erronées. Afin de pouvoir mieux distinguer les PRMs correctes des erronées, on attribue à chacune d'elles un score basé sur la probabilité que l'hypothèse ionique attribuée soit correcte. Par hypothèse, les PRMs de même masses sont supposées représenter le même fragment et sont réunies en un nœud unique (toute l'information originale est cependant maintenue dans chaque nœud). Les nœuds peuvent donc contenir une ou plusieurs PRMs. Seuls les nœuds avec au moins une PRM ayant un score élevé sont sélectionnés pour former le graphe. Enfin, les nœuds qui ont au moins une PRM dont la différence de masse correspond à celle d'un ou deux acides aminés (étant donné une erreur autorisée) sont connectés par un pont. Plus l'erreur autorisée est grande, plus le graphe sera connecté et le nombre de chemins possibles important.

## Extraction des tags

L'extraction des tags est faite indépendamment pour chaque peptide candidat de la banque de donnée. Les sous-séquences de ce dernier sont tout d'abord indexées par un arbre des suffixes. L'extraction se fait en parcourant simultanément le graphe et l'arbre. La recherche est lancée indépendamment à partir de chaque nœud du graphe et procède de manière récursive. Au fur et à mesure qu'un chemin est exploré, la séquence d'acides aminés est complétée par les acides aminés correspondant au pont parcouru. Tant que le tag correspond à une sous-séquence du peptide candidat, l'exploration continue en profondeur. Dans le cas contraire, le tag est stocké dans une liste (pour autant qu'il ait atteint une longueur minimale de trois nœuds) et la récursivité permet de remonter et dans le graphe et dans l'arbre des suffixes.

## Création des scénarios

Une fois les tags extraits, Popitam cherche les différents moyens de les combiner afin de former des scénarios d'interprétation de spectre pour la séquence peptidique analysée. Un scénario correspond à un « run-and-jump path », c'est-à-dire à un parcours qui emprunte les ponts du graphe tout en pouvant sauter d'un nœud à un autre (de masse plus élevée) sans qu'ils soient connectés. Les chemins parcourus sur les ponts sont les tags, et les sauts sont les *modGaps* et les *lackGaps*. Un certain nombre de règles de compatibilité sont énoncées et définissent si un tag peut cohabiter avec un autre dans un scénario. Ainsi, les tags chevauchants ou recouvrants sont incompatibles, de même que les tags continus. Pour déterminer les combinaisons de tags compatibles, les tags sont structurés en un graphe de compatibilité, dans lequel les nœuds représentent des tags et les ponts relient les tags qui sont compatibles. Il suffit alors de rechercher les sous-graphes complets (cliques) dans le graphe de compatibilité pour construire les différents scénarios. En comparant les masses des gaps avec celles qu'on attendrait selon la séquence du peptide candidat, Popitam les classifie soit en *modGaps* (lorsque la différence excède un certain seuil) soit en *lackGaps*.

## Evaluation des scénarios

Pour chacun des peptides candidats, zéro, un ou plusieurs scénarios sont construits. Il faut alors pouvoir repérer, parmi tous les scénarios, le ou lesquels correspondent au peptide candidat correct. Idéalement, un scénario correct devrait posséder les caractéristiques suivantes: les tags devraient couvrir une grande partie du peptide candidat; les *modGaps* devraient se limiter à un seul acide aminé; les nœuds parcourus devraient inclure des PRMs à scores élevés; les nœuds qui contiennent plusieurs PRMs devraient être favorisés par rapport à ceux qui n'en contiennent qu'une seule; les pics à l'origine des nœuds devraient avoir des intensités plutôt élevées; les erreurs observées entre les PRMs et les masses attendues calculées d'après la séquence peptidique devraient être minimisées; les scénarios incluant des nœuds de même type ionique devraient être favorisés. Enfin, différents tags ne devraient pas inclure plusieurs fois un même pic. Afin de décrire au mieux chacun de ces différents aspects, nous avons défini un ensemble de 12 sous-scores. Encore fallait-il trouver la manière de les combiner afin d'optimiser au mieux cette information et de pouvoir identifier et discriminer le bon peptide parmi la liste de tous les peptides candidats. Dans un premier temps, nous avons utilisé une fonction empirique (multiplication des sous-scores); puis nous avons utilisé la Programmation Génétique pour explorer de multiples combinaisons possibles de sous-scores afin de produire une fonction de score optimisée.

# UTILISATION DE LA PROGRAMMATION GENETIQUE POUR APPRENDRE DES FONCTIONS DE SCORE

La programmation génétique (GP) est une approche non-déterministe, inspirée de l'évolution naturelle des espèces et permettant de résoudre des problèmes d'optimisation combinatoire. La GP modélise les principes-clés de l'évolution, à savoir la reproduction différentielle et la variation des caractères héritables. Ces principes sont appliqués sur une population de solutions spécifiques au problème considéré et codées sous la forme d'arbres. A chaque génération, les solutions de la population courante sont évaluées par une fonction objective. Les solutions « se reproduisent » pour former une nouvelle population, les fonctions les mieux évaluées ayant plus de chance de transmettre leur descendance à la nouvelle génération, selon le principe de la survie des meilleurs. Pendant la reproduction, des opérateurs génétiques tels que la recombinaison (échange d'information entre deux solutions) ou la mutation (remplacement aléatoire d'un terme de la solution) sont utilisés pour générer de la variabilité. Au cours des générations, l'algorithme converge vers des solutions toujours plus adaptées et performantes. A la fin du processus, les meilleures solutions découvertes au cours des générations sont reportées comme résultat de l'apprentissage.

Dans notre cas, les solutions sont des fonctions de score de scénarios. Les nœuds internes des arbres représentent des opérateurs mathématiques et logiques, tandis que les feuilles représentent les sous-scores. L'évaluation d'une fonction se fait en exécutant Popitam avec cette fonction sur un set d'apprentissage composé de spectres dont les identifications sont connues et en « comptant » le nombre de spectres correctement identifiés. Nous avons opté pour une évaluation multi-objective, ce qui nous a permis d'optimiser simultanément trois critères : a) la capacité de la solution à placer le peptide correct en tête de liste (rang du peptide correct); b) la capacité de la fonction à discriminer le score du peptide correct des scores des autres peptides; et c) enfin, la taille des solutions engendrées

(qui doit être minimisée). L'objectif de ce troisième critère est d'éviter que les fonctions ne deviennent trop complexes, ce qui limite le risque de sur-apprentissage.

# RESULTATS

Les résultats présentés concernent deux aspects: l'apprentissage de fonctions de score de scénarios par programmation génétique, et le potentiel de Popitam à identifier et caractériser des peptides modifiés. Tout d'abord, nous montrons, par une procédure de dix runs et validations croisées, que la programmation génétique est une méthode robuste pour apprendre des fonctions adaptées à notre problème. Selon le principe de la multi-objectivité, chaque run de GP produit non pas une fonction, mais un ensemble de fonctions co-dominantes. Ces dernières peuvent être variées et présenter des tailles très différentes (d'une dizaine de noeuds à plus d'une centaine). Nous suspections les fonctions complexes d'être sur-apprises et d'avoir par conséquent perdu leur pouvoir de généralisation sur des données non vues. En montrant que ces fonctions donnent également de bons résultats sur les ensembles de test, nous avons pu écarter l'hypothèse de sur-apprentissage. Ensuite nous avons comparé la performance des fonctions apprises par rapport à des fonctions empiriques et montré que les premières donnaient de meilleurs résultats. Nous avons également relevé la difficulté croissante de trouver des fonctions performantes pour des scénarios avec un, puis deux *modGaps*.
Dans un deuxième temps, nous avons montré le potentiel de Popitam sur des spectres MS/MS provenant d'expériences indépendantes. Popitam a ainsi pu identifier et caractériser un peptide portant deux cystéines carbamidomethylées. Le spectre venait d'une étude portant sur les protéines nucléolaires humaines. Puis, à partir de spectres provenant d'une étude sur la chaîne A de l'Alpha crystalline de souris, Popitam a pu identifier et caractériser des spectres provenant de peptides modifiés (N-acétylation, caractérisée par un *modGap* en position N-terminal de 42.0106 Daltons), semi-tryptiques et transpeptidés sans qu'aucune indication ne soit donnée *à priori* à Popitam concernant le type de modification recherchée (excepté un intervalle de masse autorisées allant de –100 à 400 Daltons).

# FOREWORD

The present document is a thesis in Bioinformatics on protein identification and characterization by tandem mass spectrometry. It is the result of four years of research and learning at the Swiss Institute of Bioinformatics in Ron Appel's group, in Geneva.

The beginning of this work was specially exciting: the draft sequence of the human genome had just been published, the new sciences in "ics" were blooming, the enthusiasm was palpable and the technology boom that followed was heady. We felt like living a particular epoch and participating at something very important. One year before, in 2000, the Swiss Institute of Bioinformatics, based in Geneva and Lausanne, organized the first DEA study cursus in Bioinformatics (DEA means "Diplôme d'Etudes Approfondies") of Switzerland. This one-year cursus gave students a taste of genomics, transcriptomics and proteomics and has been a springboard to start a PhD work at the Proteome Informatics group.

We tackled a tricky subject, the automated identification and characterization of peptides with unexpected amino acid modifications or mutations using tandem mass spectrometry. Four years ago, this topic was practically virgin territory, with only an orphan publication on the subject dating from years before. This probably explains why our method, Popitam, matured slowly, roaming in a desert valley. But soon, a new interest started growing for this subject. Recently, at least two new approaches for "open-modification search" have been described. Even now the coverage of this field is poor. Nevertheless, second sight is not required to foretell that new developments will come to this field in the near future.

The thesis is organized as follows:
Chapter I gives a general introduction to the subject. It presents the context of protein identification, brings up the difficulties of identifying modified peptides and briefly presents Popitam's approach.
Chapter II gives basic notions about proteins. It describes their structure, explains how they are coded and synthesized. A particular emphasis is given to post-translational modifications of proteins.
Chapter III describes techniques applied to the analysis of proteins. In the chapter's first part, various methods for separating and purifying proteins are introduced. A second part describes the principles of mass spectrometry (MS), and more particularly shows how tandem mass spectra can be obtained by combining several mass spectrometry stages.
Chapter IV presents existing protein identification approaches using mass spectrometry. It shortly introduces MS-based identification, and then explains in greater detail different MS/MS identification approaches.
Chapter V is devoted to "open-modification search" methods that directly concern the subject of this thesis.

Chapter VI gives a detailed description of Popitam's algorithm.
Chapter VII describes Genetic Programming, an approach we used to elaborate optimized scoring functions for Popitam. As a starter, it introduces the theory of evolution, which inspired the Genetic Programming optimization approach. Then the chapter charts the methodology in detail.
Chapter VIII presents some results we obtained with Popitam and different scoring functions. The potential of the method is shown and discussed in this chapter.
Finally, Chapter IX outlines future perspectives and brings concluding remarks.

# TABLE OF CONTENT

# TABLE OF FIGURES

La curiosité est le premier pas vers l'Enfer.

Curiosity is the first step to Hell.

Figure adapted from "Monsieur Hippopotame" by Tanikawa Shuntarô

# CHAPTER I

# INTRODUCTION

This chapter introduces the subject of protein identification and its context. It depicts the techniques employed to transform proteins detected in a biological sample into computer readable data, and tackles the different approaches used to correlate the obtained data to known protein sequences. It brings up the difficulties of identifying modified peptides and introduces Popitam's approach to bypass this issue.

# I. Introduction

Proteins are essential constituents of all living organisms. They are constituted of amino acids linearly assembled to form polypeptide chains. Interactions between non-adjacent amino acids allow proteins to acquire a 3-dimensional (3-D) structure. The tasks of proteins are many and various: among other, they regulate gene expression and other protein activity, they transport molecules inside and outside the cell, they play a role of messengers as well as signal receptors, they allow muscles to contract, they are structural components maintaining the shape and integrity of organisms, they are part of the immune memory and can be used as energy stocks. Proteins are encoded by genes. When a given protein is synthesized, the gene coding for that protein is transcribed into molecules called messengers RNAs (mRNAs), which are then translated into amino acid chains. Newly produced proteins then follow several maturation steps, take their 3-D structure and are directed towards their working compartment, where they fulfill their function, until degradation. Most Eukaryote proteins undergo post-translational modification events after their synthesis. One type of modification consists in linking molecular groups (e.g. small molecules like acetates and phosphates, but also sugars, lipids, and even peptides) on amino acids. These modifications, denoted as PTMs, are mediated by specific proteins, called enzymes and allow proteins to be fully functional. Many biological processes are PTM related, including protein folding, enzymatic activity regulation, protein-protein interaction, subcellular localization and protein turnover.
Alteration of a protein function may potentially cause observable perturbations –or diseases- in the living organisms. Such alterations are often due to mutational events, either in the very genes coding for the altered protein or in genes coding for any protein implicated into the expression, maturation and functioning of the altered one. As techniques to study proteins, and more particularly PTMs, evolve, researchers start to consider diseases from a biochemist's point of view rather than from the molecular biologist's view. As a matter of fact and with no doubts, proteins are the very actors in diseases. This new interest results in new studies that emphasize the extent of PTM importance in diseases. For example, Castegna et al. (Castegna et al. 2003) showed that nitration of amino acid tyrosine is a pathological event associated with several neurodegenerative diseases, such as amyotrophic lateral sclerosis, Parkinson's disease and Alzheimer's disease. Another study (Yang 2004) showed the correlation between mutation of the gene coding for the acetyltransferase – an enzyme that promotes lysine acetylation- and leukemia.

The present study is precisely about the identification and characterization of modified and mutated peptides using tandem mass spectrometry (MS/MS). An example of "bottom-up" protein identification workflow is outlined in Figure I-1. Although variations may occur at various steps, either in the experimental protocol (e.g. no purification, no digestion and so on) or in the applied techniques, the general framework remains similar. The first step is sample preparation. It consists in breaking the cells and isolating cellular compartments of interest. Impurities, like lipids or DNA, are discarded, and the proteins are dissolved and denaturated –i.e. proteins are unfolded by breaking hydrogen bonds, hydrophobic interactions and salt linkages–. Then, one or several separation techniques are applied to reduce the sample complexity. Thus, in a systematic analysis of nucleolar constituents, the sample will contain several hundreds of proteins, while in a targeted analysis of proteins that reveal differential expression rates as a response to given stimuli, the sample will be of very few –and even only one– purified protein. A very common separation technique is 2-D gel electrophoresis, which scatters proteins in a rectangular polyacrylamide gel by applying two

separation steps in two orthogonal dimensions. Usually, the proteins are separated with respect to their charge (isoelectric point[1]) in the horizontal dimension, and with respect to their mass in the vertical one. Proteins to be identified are isolated from the gel and digested, i.e. cleaved into shorter pieces, called peptides. This is made possible by using endoproteases, which are molecular scissors able to cut protein sequences at particular amino acid patterns. The resulting peptide mixture can then be analyzed by mass spectrometry.

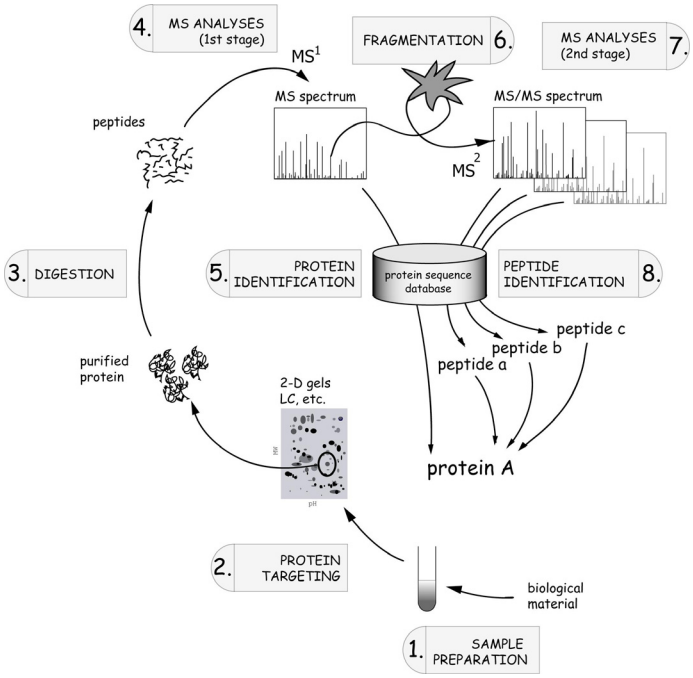



*Figure I-1: A typical workflow for "bottom-up" protein identification*
*Steps (1) and (2) is preparing the sample and targeting proteins to analyze. Then the proteins are digested (3) and the resulting peptides undergo mass spectrometry analysis (4) yielding to an MS spectrum that can be identified by an MS identification algorithm (5). The analysis can go further by isolating peptides, fragmenting them (6) and measuring the resulting fragment masses (second stage of mass spectrometry) (7), yielding to an MS/MS spectrum for each isolated peptide. The obtained MS/MS spectra are then correlated with theoretical peptide sequences using an MS/MS identification algorithm (8).*

"Top-down" identification workflows apply mass spectrometry on intact proteins without prior digestion step. This approach will not be discussed in the present document, although it represents nowadays an attractive alternative to "bottom-up" proteomics and has proven to be particularly powerful for detailed protein characterization. A review of the subject can be found in (Reid and McLuckey).

Mass spectrometers are kinds of scales for biomolecules designed to isolate analytes and determine their mass. When applied to the identification of highly purified proteins, the mass spectrometer is used to produce a raw MS spectrum (a spectrum composed of a raw measured signal), from which a list of peaks is extracted by peak detection software, leading to an MS spectrum (a spectrum

[1] The isoelectric point of a protein is the pH at which the protein has an equal number of positive and negative charges.

composed of discrete signals). Each peak in a MS spectrum represents the mass of a peptide, and, since each protein has a different sequence, the composition of all obtained masses represents a unique key -a "fingerprint"- of the protein. The identification by "Peptide Mass Fingerprinting" (PMF) (see Figure I-2) then consists in comparing the experimental spectrum with theoretical fingerprints (obtained by virtually digesting protein sequences and computing the theoretical masses of the obtained peptides). For each comparison (or match), a similarity quality is measured according to a scoring function leading to an identification score. The best scoring theoretical spectrum is then proposed as the identification. When the sample contains a protein mixture or the database is too large, the PMF identification method can often not be efficiently applied. In such cases, a first stage of mass spectrometry is applied to isolate peptides, which are subsequently fragmented. A second stage of mass spectrometry measures, for each peptide, the mass of the obtained fragments, and produces an MS/MS spectrum, in which peaks represent masses of peptide fragments. As for MS spectra, MS/MS spectra can be interpreted and correlated with theoretical peptide sequences obtained from protein databases, using a similar approach denoted as "Peptide Fragment Fingerprinting" (PFF). Then a list of identified proteins is proposed based on combinations of peptide identifications.
MS/MS spectra are more informative than MS spectra. In particularly, since a peptide preferably fragments on the peptide bonds, information about its sequence can be deduced from the pattern of peaks present in MS/MS spectra without using database information, and post-translational modifications or mutations can be precisely mapped on the peptide sequence. This explains the greater diversity in MS/MS identification approaches, expounded in Chapter IV, compared to MS identification methods.

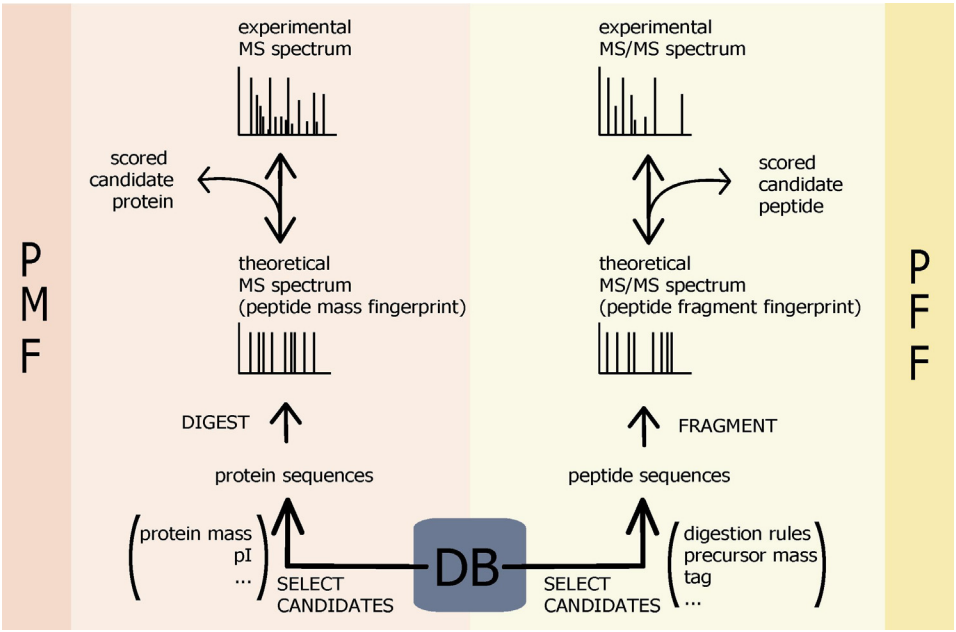
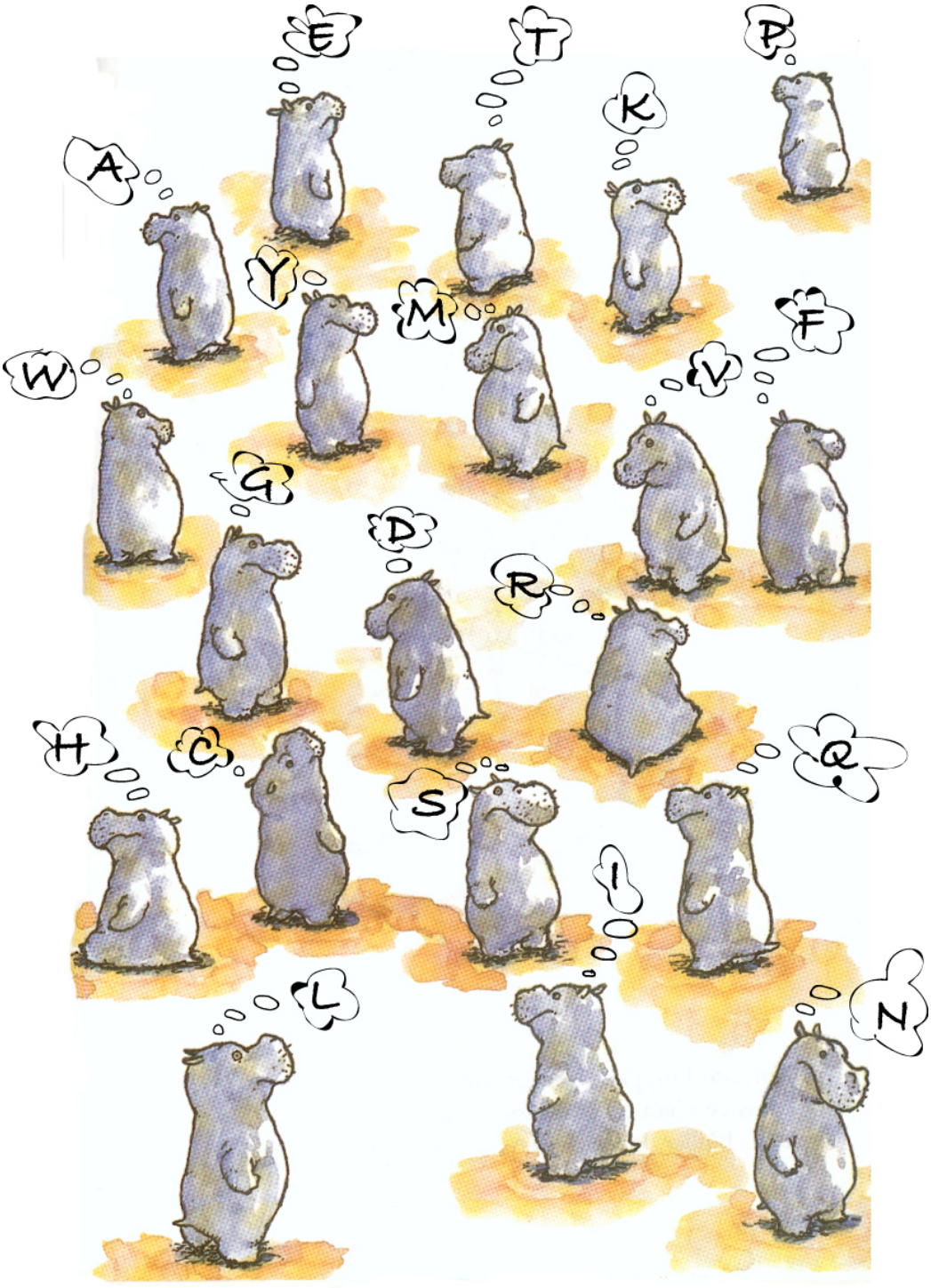



*Figure I-2: MS and MS/MS identification procedures*
*MS identification by peptide mass fingerprinting (PMF) correlates an MS spectrum (or peptide mass fingerprint) with theoretical spectra obtained from virtually digested protein sequences. MS/MS identification by peptide fragment fingerprinting (PFF) correlates an MS/MS spectrum (or peptide fragment fingerprint) with theoretical spectra obtained from virtually fragmented peptide sequences. In both cases, candidate sequences are selected according to filtering criteria like protein mass or isoelectric point (pI) in the case of PMF, and digestion rules, peptide mass or a short amino acid sequence (a tag) extracted from the spectrum for PFF.*

The identification of peptides carrying unexpected modifications is a major challenge for MS/MS identification algorithms, due to an increase of the search space and to the need for special comparison routines. One (or several) modifications or mutations will cause the mass of the peptide (called precursor mass) to be shifted compared to the corresponding theoretical peptide mass. Consequently, those algorithms that filter candidate peptides based on the experimental precursor mass will inevitably miss the correct theoretical sequence, leading to an unsuccessful identification. It is therefore necessary, for identifying peptides carrying unexpected modifications, to relax precursor mass constraints. An alternative is to extract short unambiguous sequences (called tags) from peak succession patterns that usually appear in MS/MS spectra, and to use those as a sequence-based filter (instead of a precursor mass-based filter). Even if an adapted filter is used (the correct peptide is thus presented as candidate), the comparison will probably produce low scores. The reason is that every peak in the experimental spectrum that comes from a modified fragment has a measured mass that is shifted compared to its expected position computed from the corresponding theoretical database sequence. Unexpected modifications can only be taken into account when considering all possible mass shifts between the experimental and theoretical spectra, consequently significantly increasing the comparison procedure complexity, and thus requiring new specific identification strategies. These considerations led us to develop a new identification algorithm, called Popitam[2], which can be designated as an "open-modification search" identification method. Popitam's approach allows considering any type and a reasonable number of modifications and mutations when comparing a theoretical peptide with an MS/MS spectrum. The core process of Popitam is the following: it performs, for each candidate peptide, a sequence-guided tag extraction from the spectrum. Then it analyses the various tag combinations and builds scenarios composed of tags separated by gaps. Popitam considers two types of gaps: the *lackGaps*, which are due to missing information in the spectrum and the *modGaps*, which are characterized by a mass shift and correspond to one (or several) modifications (or mutations). A scenario can then be considered as a possible interpretation of the spectrum peak patterns for a theoretical peptide. Each scenario is evaluated according to a scoring function optimized by Genetic Programming, the candidate peptide with the overall best scoring scenario being proposed as identification.

[2] Popitam has been made available through the ExPASy server (www.expasy.org/tools/popitam/)




Les acides aminés, c'est ce qui nous anime.

Amino acids animate us.

Figure adapted from "Monsieur Hippopotame" by Tanikawa Shuntarô

# CHAPTER II

# BIOLOGICAL BASES

This chapter gives basic notions about proteins. It describes their structure, explains how they are coded and synthesized. A particular emphasis is given to post-translational modifications of proteins.

# II. Biological bases

## II.1. Proteins

### II.1.1. Protein structures

Proteins are composed of chemical molecules called amino acids. Each amino acid is constituted of a carboxyl group (-COOH), an amine group (-NH2) and a lateral side chain denoted by R. There are 20 common naturally occurring types of amino acids (see Figure II-1) that differ by their side chain structure. Each of them is denoted by a three-letter and a one-letter code.

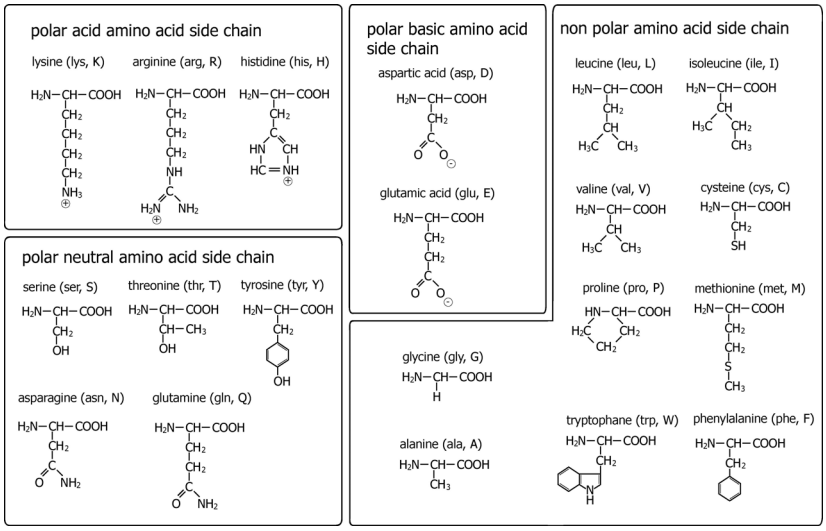



*Figure II-1: Names and structures of the 20 standard amino acids*

Amino acids can be bound by covalent bonds (called peptide bonds), and form oriented linear chains (see Figure II-2). Usually, short amino acid chains (less than 50 amino acids) are called peptides, while longer chains are called proteins.

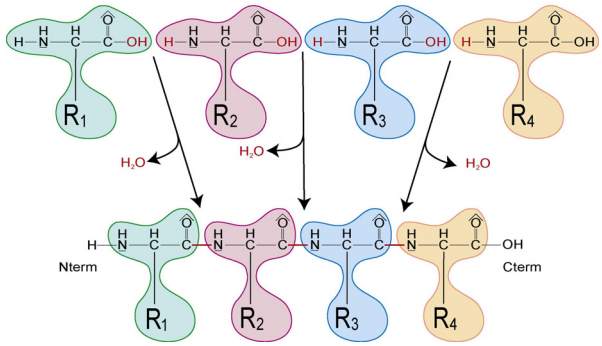



*Figure II-2: Formation of a tetrapeptide*
*Peptides are formed by linking amino acids together through peptide bonds (in red). The reaction involves the carboxyl carbon atom of one amino acid and the amino nitrogen atom of another amino acid, and the release of a water molecule.*

As each amino acid can be labeled by a letter, protein sequences can be written as a suite of alphabetic characters. For example, the insulin precursor sequence can be represented as follows:

```
MALWMRLLPLLALLALWGPDPAAAFVNQHLCGSHLVEALYLVCGERGFFY   50
TPKTRREAEDLQVGQVELGGGPGAGSLQPLALEGSLQKRGIVEQCCTSIC  100
SLYQLENYCN
```

Each type of protein consists in a unique sequence of amino acids with a length generally comprised between a few and several hundred amino acids. The first amino acid is always the N-terminal amino acid, and the last one is the C-terminal one.

## II.1.2. Protein synthesis

### II.1.2.1. Protein sequences are coded in genes

Genes are portions of chromosomes that code for a protein. They are linear molecules, located in the cell nucleus (for Eukaryote organisms) and composed of two complementary and anti-parallel strands of deoxyribonucleic acids (DNA) coiled around each other in a double helix. The subunits of DNA are the deoxyribonucleotides. Each of them is composed of a phosphate, a deoxyribose (a ribose that lost an oxygen) and a heterocyclic base that comes in four flavors: adenine, thymine, guanine and cytosine. The two DNA strands are maintained together by complementary base-pairing between adenine and thymine, and between guanine and cytosine (see Figure II-3). Polarity is given by the deoxyribose linkages. By convention, the strand 5'→ 3' is considered as the coding strand, while the strand 3'→ 5' is the complementary one (the template). The human genome contains three billions base-pairs for less than 30'000 genes. But human genes represent only a small part (one to a few percent) of genomic DNA. Non-coding DNA include for example structural material, repeated sequences and transposable elements.


*Figure II-3: Schematic representation of DNA structure*
*The two DNA strands are maintained together by base-pairing and arranged in a double helix.*

The synthesis of a protein involves two processes: **transcription** of the coding DNA strand into a ribonucleotide acid (RNA) molecule and further **translation** of the RNA into amino acid sequences. Transcription of a gene (see Figure II-4) is catalyzed by an enzyme called RNA polymerase. The process is initiated by the binding of a polymerase to a gene promoter region. Then, the polymerase opens the DNA helix and proceeds in the 5'→ 3' direction, "reading" the template strand and using the complementary base-pairing principle to correctly assemble free-swimming ribonucleotides on

the growing RNA molecule. Ribonucleotides are very similar to the deoxyribonucleotides used for DNA, except that they are attached to riboses instead of deoxyriboses, and that the thymine base is replaced by a slightly different one, called uracil (U). Elongation proceeds until a termination site on the gene is encountered, causing dissociation of the polymerase from the DNA and the relaxing of the completed RNA. The obtained RNA, called pre-mRNA, is single stranded, and its nucleotide sequence is exactly similar to the coding DNA strand it was built from (with Us in place of Ts). In Eukaryote organisms, the pre-mRNA must be processed into a mature RNA (mRNA or messenger RNA) before translation. During this process, regions from the genes that are not necessary to the proteins (called introns) are excised from the pre-mRNA, and the remaining RNA portions (the exons) are spliced together. By using alternative excision combinations (alternative splicing), a pre-mRNA can give rise to different proteins.

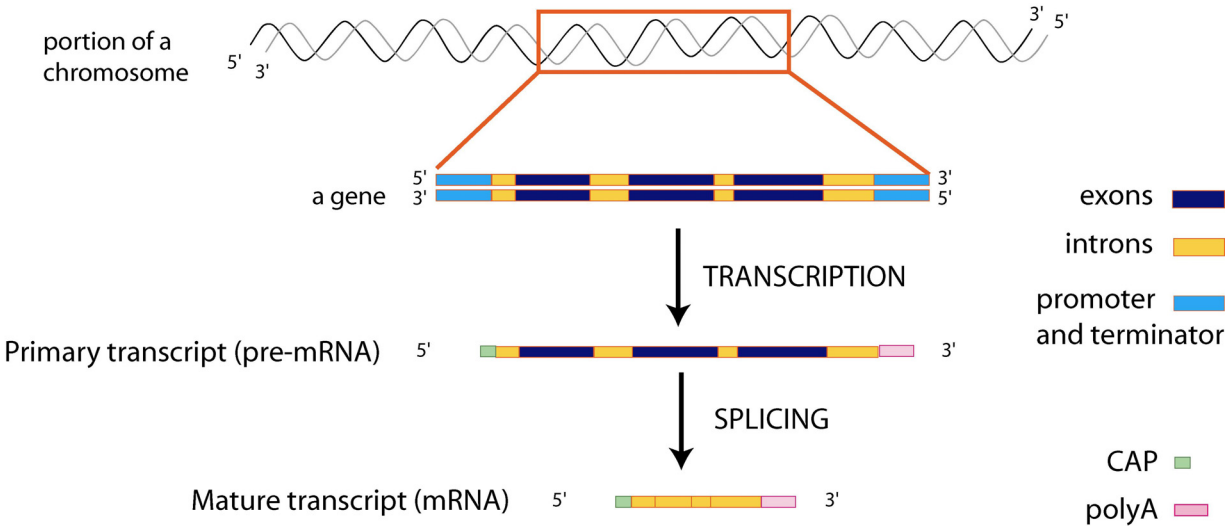



*Figure II-4: Transcription of a gene into a mature RNA (mRNA)*
*Transcription starts in the promoter region and goes on until a termination site is met. The obtained primary transcript is an exact copy of the coding DNA strand, except that it contains riboses instead of deoxyriboses, and uracil (U) bases in place of thymine bases. In addition, it has a cap that is added during the transcription, and a tail composed of 100-200 adenylic acids. The primary transcript is then processed into a mature mRNA by excision of unnecessary RNA regions (called introns).*

### II.1.2.2. Translation of mRNA into proteins

After maturation, the mRNA molecules leave the nucleus and gain the cytosol of the cell, where they are interpreted into protein sequences according to specific rules known as the genetic code (Figure II-5).

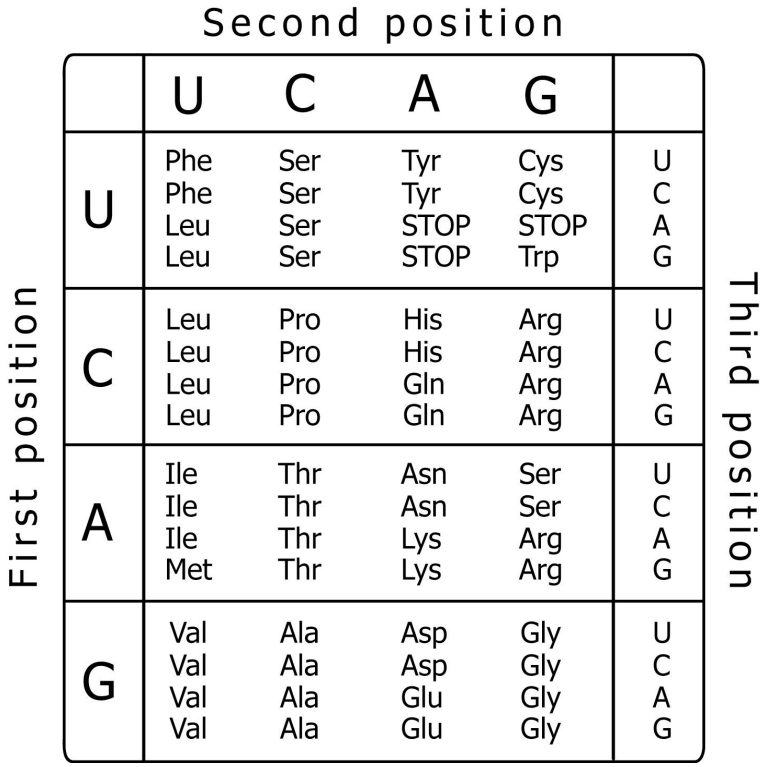



*Figure II-5: The standard genetic code*
*Nucleotide triplets (codons) in RNA are translated into amino acids according to the genetic code. There are 64 possible codons, coding for the 20 amino acids plus three codons (UAA, UAG, UGA) that code for a termination signal. (Copied from Alberts et al., Molecular biology of the cell, Garland Publishing, Inc. New York&London, 1983)*

Translation is undertaken by the ribosomes, which are complex machineries that include more than a hundred different proteins associated with structural RNA molecules (rRNAs). Interface between mRNA and amino acids is assured by adaptor molecules called tRNAs (for transfer RNAs). These molecules are composed of 75 to 95 ribonucleotides and contain two important sites: an acceptor site and an anticodon loop. The anticodon loop is made of three adjacent ribonucleotides and is used to read the mRNA sequence by complementary base-pairing. The acceptor site binds the amino acid coded by the anticodons (e.g. tRNA with anticodons UUU will recognize codon AAA, and then carries an lysine amino acid). Initiation of the translation involves the binding of a subunit of the ribosome to the mRNA. The subunit proceeds downstream the mRNA (in direction 5'→ 3') until it encounters a specific codon (AUG, coding for methionine). An initiator tRNA, carrying a methionine, binds to the complex ribosome-mRNA and initiates the polypeptide chain. From this point, the mRNA is read codon by codon until a stop codon is met. The elongation phase, illustrated in Figure II-6, consists in linking new amino acids to the growing polypeptide chain, using the complementary base pairing between codons of the mRNA and corresponding anticodons of the tRNAs. When a stop codon is met, the ribosome complex dissociates and the complete polypeptide is released.

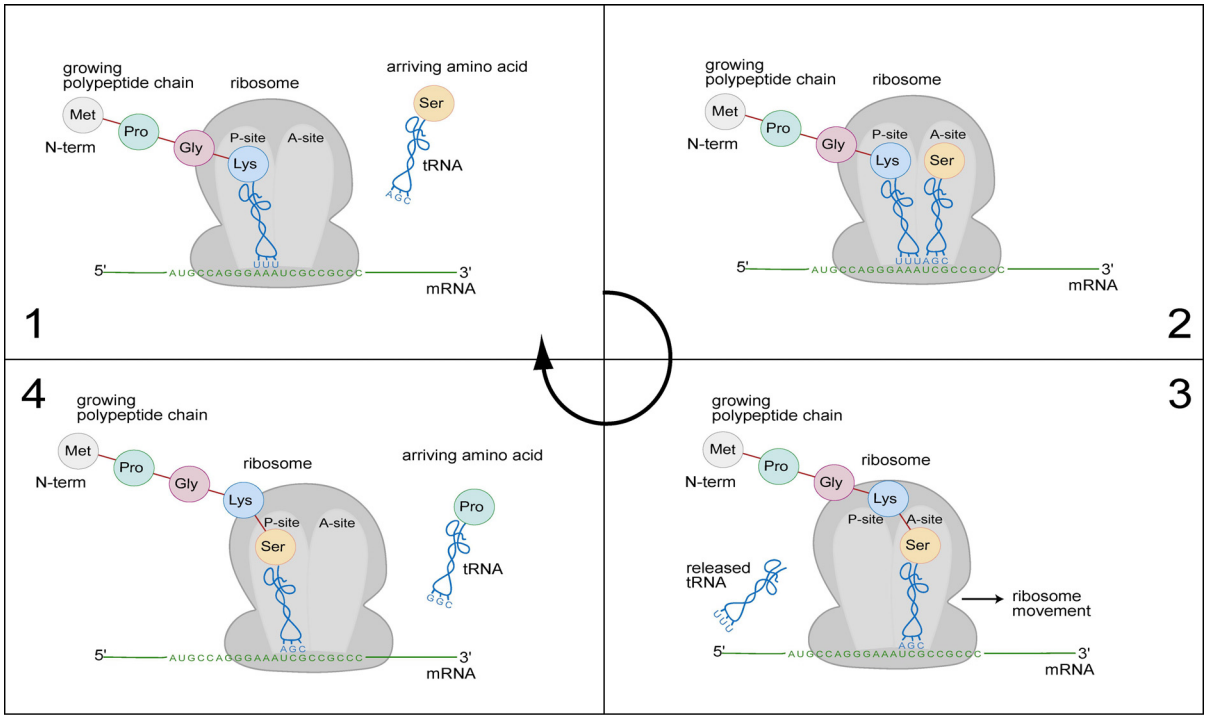



*Figure II-6: Elongation of a polypeptide chain*
*The mRNA strand is read by the ribosome from one end to the other, codon per codon. Amino acids to be incorporated are carried to the ribosome by "adaptors", denoted as transfer RNAs (tRNAs) (box 1) and bind the A-site of the ribosome (box 2). A new peptide bond is formed between the amino acid located at the P-site and the new one (box 3). Then, the ribosome moves three nucleotides towards the 3' end of the mRNA (box 4) and the cycle is repeated until a stop codon is reached. (Adapted from Alberts et al., Molecular biology of the cell, Garland Publishing, Inc. New York&London, 1983).*

After its synthesis, the polypeptide chain undergoes post-translational modifications (see Section II.2), including possible cleavages and adjunction of chemical groups to given amino acid residues, and takes its 3-D shape. Then, the mature protein travels to its destination and fulfills its function until it is targeted for degradation. All its amino acids are then cleaved and re-used for the synthesis of new proteins.

## II.2. Post-translational modifications

### II.2.1. Introduction

Many proteins are permanently or reversibly modified during or after their synthesis. Co- and post-translational modifications include proteolytic cleavage events, inter- or intra-peptidic linkages and adjunction of chemical groups at specific amino acid residues.

## II.2.2. Cleavages of polypeptide chains

### II.2.2.1. N-terminal methionine removal

As the translation is initiated by an AUG codon, the first amino acid incorporated in the protein is methionine. Most proteins undergo the removal of this amino acid during their synthesis.

### II.2.2.2. Signal sequence and transit peptide cleavage

Proteins to be secreted out of the cell are labeled by a signal sequence, which directs them inside a particular cell compartment, called endoplasmic reticulum (ER). Once inside the ER, the signal sequence becomes useless and is cleaved. A similar targeting mechanism (called transit peptide) allows proteins to be directed into chloroplast, mitochondria as well as other cell compartments. About 20% of human proteins are annotated in the Swiss-Prot database (Section II.4 introduces the concept of protein databases) as containing a signal sequence, and about 2.5% human proteins are annotated as containing a transit peptide.

### II.2.2.3. Internal peptide sequence processing

Proteins may also have internal portions –called propeptides- cut off. A very known example of such an event is maturation of insulin (see Figure II-7), which involves signal sequence cleavage, as well as propeptide excision and formation of disulfide bonds between cysteines. The function of insulin is to regulate the glucose concentration in the blood. Its precursor is produced in pancreatic cells and, since the protein is to be secreted, it contains a signal sequence that targets it into the ER. Once in the ER, the signal sequence is cleaved, and an internal 31 amino acid long stretch is excised. The mature form of insulin is then stored in secretory granules that accumulate in the cell cytosol. In case of high glucose concentration, it is secreted and binds specific receptors on the cell surface causing an increase of the cell permeability to monosaccarides.

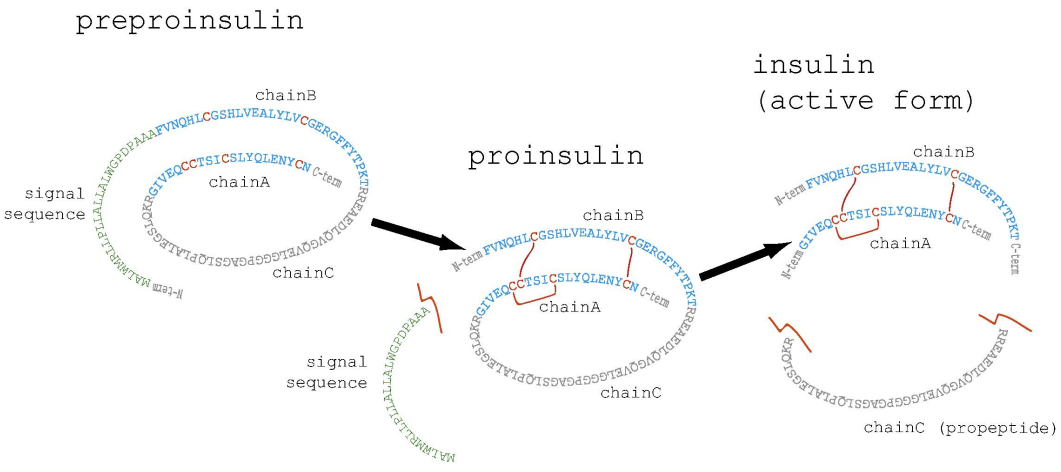



*Figure II-7: Post-translational processing of the preproinsulin*
*The maturation of the preproinsulin includes removal of the signal sequence, folding of the protein by formation of disulfide bonds and excision of a 31 amino acid peptide from the center of the chain.*

## II.2.3. Addition of chemical groups

Post-translational modifications that consist in adding chemical groups to amino acid residues are common events in Eukaryotes. We shall from now on denote this kind of events PTMs. The Swiss-Prot database reports predicted or experimentally observed post-translational modifications of protein sequences and classifies them into three classes: MOD_RES (adjunction of a small chemical group), CARBOHYD (adjunction of a glycan) and LIPID (adjunction of a lipid group). Table II-1 reports some statistics collected from these annotations. The statistics show that, if PTMs are quite rare events in bacteria, they are much more common in Eukaryotes.

| | BACTERIA (82'573 entries) | | EUKARYOTA (80'665 entries) | | HUMAN (12'211 entries) | |
|---|---|---|---|---|---|---|
| | entries with at least 1 occurence | # occurences per protein | entries with at least 1 occurence | # occurences per protein | entries with at least 1 occurence | # occurences per protein |
| MOD_RES | 2'038 (=2.46%) | ~1.12 | 9'494 (=11.77%) | ~2.11 | 1'715 (=14.04%) | ~1.88 |
| CARBOHYD | 15 (=0%) | ~1.9 | 16'612 (=20.59%) | ~3 | 3'143 (=25.7%) | ~4 |
| LIPID | 856 (=1%) | ~2 | 2'351 (=2.9%) | ~1.32 | 435 (=3.5%) | ~1.31 |

*Table II-1: Post-translational modifications in Swiss-Prot*
*Statistics about various post-translational modifications annotated in release 46.5 of Swiss-Prot. Three types of residue modifications are taken into account: a) adjunction of a small chemical group (MOD_RES), including for example acetylation, hydroxylation or phosphorylation; b) attachment of a glycan (mono- or polysaccharides) (CARBOHYD); and c) binding of a lipid group (LIPID). It should be noted that most annotations are not based on experimentally proven findings but rely on predictive cirteria (e.g. the modification has been experimentally observed in other protein family members).*

Glycosylation primarily concerns cell surface associated proteins as well as secreted proteins, and consists in the attachment of oligosaccharide chains to certain amino acids. It affects processes like protein folding, targeting, turnover and antigenicity (Blom et al. 2004). Lipidation allows proteins to anchor in the cell membrane. Anchored proteins can act as signal receptors and transmit information from outside the cell to the cytoplasm, thus the signal does not have to enter the cell (Casey 1995). But a great number of modifications imply less complex molecules than lipids and carbohydrates. One of the most studied (and certainly most common) such PTM is phosphorylation. Phosphorylation is considered as a fundamental regulatory cellular mechanism. It consists in a reversible attachment of a phosphate moiety to an acceptor residue (serine, threonine, tyrosine and –rarely- histidine). The reaction is catalyzed by two antagonist enzymes: the protein kinase, which binds the phosphate to the residue and the protein phosphatase, which removes them from the residue. Another example of PTM is acetylation. In Eukaryotes, acetylation is known to play an important role in gene expression. The mechanism involves acetylation of lysines and N-terminal groups (see Figure II-8) of histones, the proteins that assure DNA compaction. Acetylation induces the loosening of the histone-DNA

interaction, thus making accessible binding sites on the DNA for transcription factors, and then the genes to be transcribed.

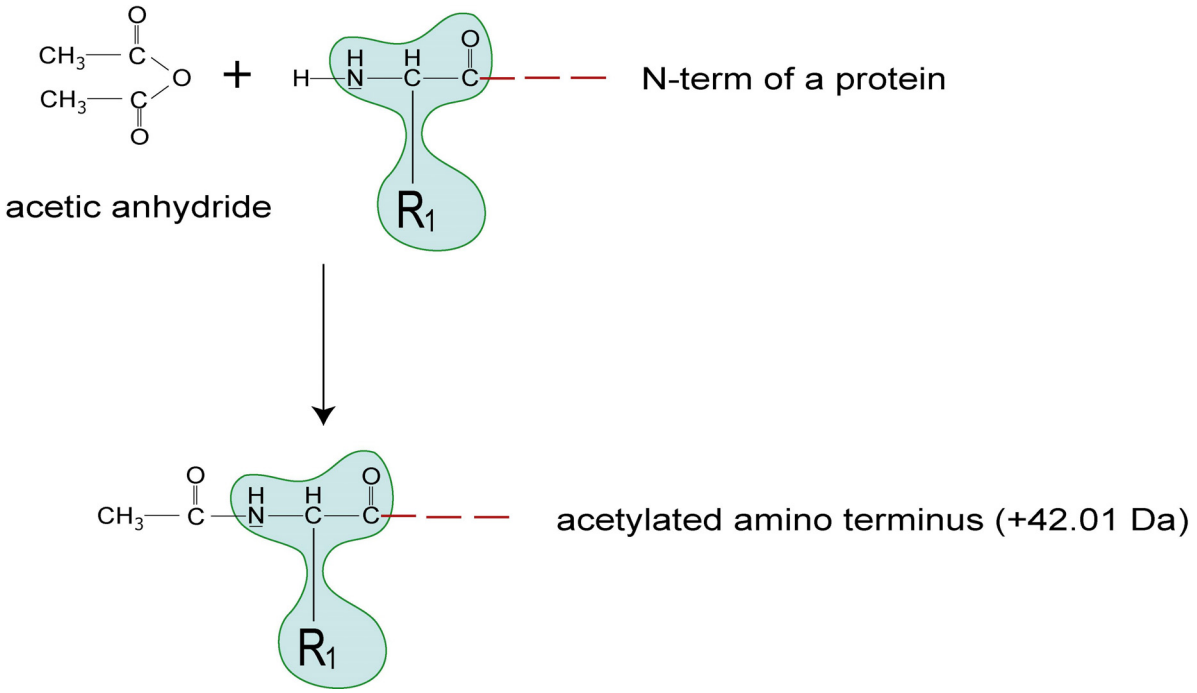



*Figure II-8: Acetylation of a protein amino terminus*

Other common modifications are given in Table II-2. Several databases documenting known modification types are available through the web. Delta Mass (www.abrf.org/index.cfm/dm.home) lists about 350 modifications and associated mass shifts (as integer values). The RESID database (Garavelli 2003) (http://pir.georgetown.edu/pirwww/dbinfo/resid.html) contains 344 entries (release 35.00, 30-Sep-2003) and provides detailed information about the modification (i.e. amino acid sites, position on the sequence, elemental composition, and so on). A third protein modification database, UniMod (Creasy and Cottrell 2004) (www.unimod.org), is specifically designed for mass spectrometry applications. It records accurate mass shifts, derived from elemental compositions of the modifications, and some useful knowledge, as amino acid modification sites and positions. The database being implemented as an open access, anybody can add new records using a web form.

Not all modifications behave the same during fragmentation. Some are very labile and fall off, while others remain linked to the peptide sequence. Examples of stable modifications are N-terminus acetylation (+42 Da) and arginine methylation (+14 Da). Examples of labile modifications are O-linked N-acetylglucosamine (GlcNAc, +203 Da) and sulfation (+80 Da) (Mann and Jensen 2003). When the modification breaks apart, it may be seen as a signature in the spectrum, yielding information about the nature of the modification, but not its location on the peptide sequence.

**Table 1. Some common and important post-translational modifications**

| PTM type | ΔMass[a] (Da) | Stability[b] | Function and notes |
|---|---|---|---|
| Phosphorylation | | | Reversible, activation/inactivation of enzyme activity, modulation of molecular interactions, signaling |
| pTyr | +80 | +++ | |
| pSer, pThr | +80 | +/++ | |
| Acetylation | +42 | +++ | Protein stability, protection of N terminus. Regulation of protein–DNA interactions (histones) |
| Methylation | +14 | +++ | Regulation of gene expression |
| Acylation, fatty acid modification | | | Cellular localization and targeting signals, membrane tethering, mediator of protein–protein interactions |
| Farnesyl | +204 | +++ | |
| Myristoyl | +210 | +++ | |
| Palmitoyl | +238 | +/++ | |
| etc. | | | |
| Glycosylation | | | |
| N-linked | >800 | +/++ | Excreted proteins, cell–cell recognition/signaling |
| O-linked | 203, >800 | +/++ | *O*-GlcNAc, reversible, regulatory functions |
| GPI anchor | >1,000 | ++ | Glycosylphosphatidylinositol (GPI) anchor. Membrane tethering of enzymes and receptors, mainly to outer leaflet of plasma membrane |
| Hydroxyproline | +16 | +++ | Protein stability and protein–ligand interactions |
| Sulfation (sTyr) | +80 | + | Modulator of protein–protein and receptor–ligand interactions |
| Disulfide bond formation | –2 | ++ | Intra- and intermolecular crosslink, protein stability |
| Deamidation | +1 | +++ | Possible regulator of protein–ligand and protein–protein interactions, also a common chemical artifact |
| Pyroglutamic acid | –17 | +++ | Protein stability, blocked N terminus |
| Ubiquitination | >1,000 | +/++ | Destruction signal. After tryptic digestion, ubiquitination site is modified with the Gly-Gly dipeptide |
| Nitration of tyrosine | +45 | +/++ | Oxidative damage during inflammation |

[a]A more comprehensive list of PTM Δmass values can be found at: http://www.abrf.org/index.cfm/dm.home
[b]Stability: + labile in tandem mass spectrometry, ++ moderately stable; +++ stable.

*Table II-2: Some common post-translational modifications.*
*Taken from (Mann and Jensen 2003)*

## II.3. Changes in protein sequences: mutations and polymorphisms

A mutation is defined as any change of a DNA sequence. Mutations may appear in somatic cells and lead for example to aging diseases. If they appear in germ cells, they can be inherited by the offspring, leading for example to hereditary diseases (i.e. Huntington's disease, a neurodegenerative disease due to a mutation in a protein called Huntingtin and whose various symptoms include uncontrolled movements and loss of intellectual faculties), but also to evolution and adaptation of species to new environments. Mutations can affect a single nucleotide base pair (point mutations) or add or delete entire stretches of DNA (chromosomal mutations). Point mutations may be induced by chemical substances or radiations, but they also spontaneously arise during DNA replication, at a rate of about 1 in every 50 million nucleotides. This means that every new human cell contains about 120 new mutations, most of them luckily occurring in non-coding DNA regions. If it occurs inside a gene, a mutation may –or may not- affect the protein encoded (see Figure II-9). Thus, when the mutated codon codes for the same amino acid than the original one ("silent" mutation), the protein is not affected at all. "Missense" mutations occur when the mutated codon codes for a different amino acid

than the original one. In this case, the protein function may be maintained (for example, when the new amino acid has similar biochemical and/or physical properties than the original one), or may be modified (for example, when the mutated amino acid destabilizes the 3-D conformation of the protein). “Nonsense” and “frameshift” mutations are usually much more dramatic. In the first case, the mutation changes the original codon into a stop codon, causing premature release of the polymerase, and therefore a truncated protein. In the second one, the deletion or insertion of one base (or of “any but non multiple of three” number of bases) results in changes of large portions of the amino acid sequence.

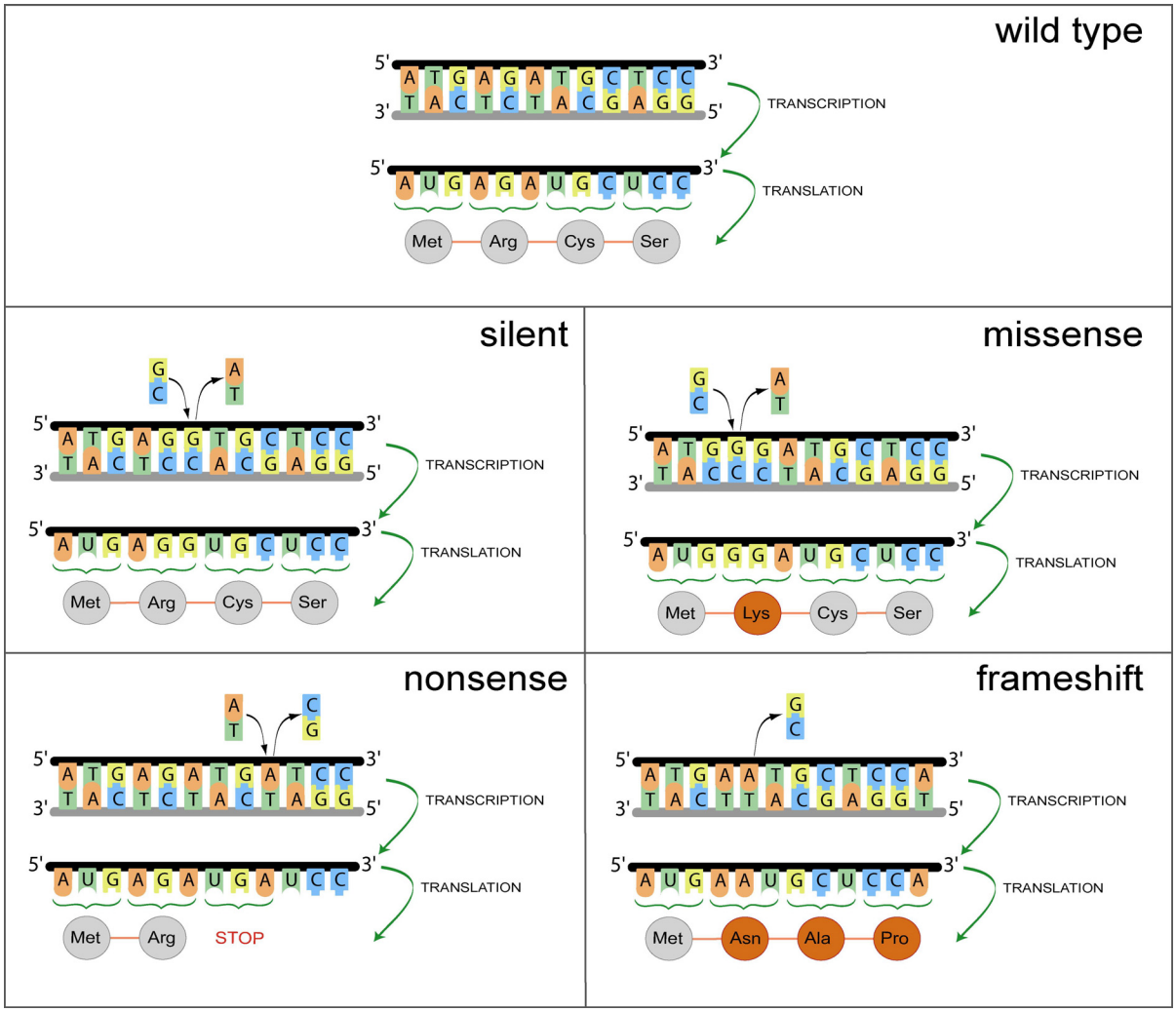


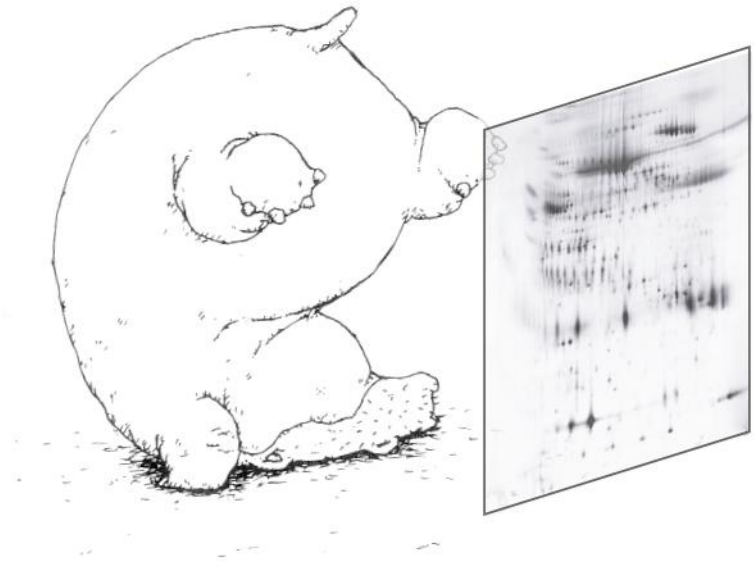

*Figure II-9: Point mutations and their consequences*
*-“ silent” mutations do not affect the amino acid sequence*
*- “missense” mutations convert an amino acid into a different amino acid*
*- “nonsense” mutations convert an amino acid codon into a stop codon, causing truncated proteins*
*- “frameshift” mutations occur when a nucleotide is deleted or added, causing changes in a large portion of the protein*

For various reasons, including individual fitness and population size, a mutation can become, with time, more common. When a mutation is present in more than one per cent of a population, it is considered as a sequence variation (one of the multiple forms on an allele) and is denoted as

polymorphism. Examples of polymorphisms are hair color or height. Swiss-Prot counts a total of 23'313 sequence variations in 3'605 human proteins (over a total of 12'861 human proteins) as of Release 48.0 of 13-Sep-2005.

## II.4. Protein databases

The first protein sequence database was published in 1965 by Dayhoff in a book named "*The First Atlas of Protein Sequences and Structure*" (1965-1978) and contained, in its initial edition, 65 protein sequences. Nowadays, various protein sequence databases exist and are accessible via internet, ranging from simple sequence repositories to non-redundant highly curated protein knowledgebases. The creation, and then exponential development of protein databases occurred because of two factors: first, the development of 2-D gel electrophoresis, which sparked off the desire to catalogue and describe all expressed proteins of living organisms, and second, the various genome sequencing projects, which fuelled protein sequences with kilometers of DNA to translate. In (Apweiler et al. 2004a), Apweiler et al. review the major protein sequence databases. The oldest of them is PIR-PSD (George et al. 1986) (http://pir.georgetown.edu/home.shtml). It grew up from the Dayhoff's Atlas and its final release (80.00) contains about 280'000 annotated entries. The Swiss-Prot knowledgebase (Boeckmann et al. 2003) (http://www.ebi.ac.uk/swissprot/), is garnished with 194'317 sequence entries (Release 48.0) highly annotated and manually curated entries from 9479 different species. Annotations include information about the function of the protein, post-translational modifications, domains, similarities to other proteins, sub-cellular localization, alternative splicing, polymorphisms, diseases associated with deficiencies of the protein, and more. Before being manually validated and incorporated into the Swiss-Prot database, protein sequences obtained from automatic translation of coding genomic sequences are computer-annotated and stored in a "buffer" database, trEMBL (Boeckmann et al. 2003) (http://www.ebi.ac.uk/trembl/). Release 31.0 of trEMBL contains 2'105'517 entries. Recently, PIR-PSD, Swiss-Prot and TrEMBL have been combined into a single resource, UniProt (Apweiler et al. 2004b) (http://www.expasy.uniprot.org/), thus providing scientists with the world's most complete and comprehensive protein knowledgebase.



Il parait que c'est la carte de mon fluide cérébrospinal.
Mais où est donc le Nord?

It is supposed to be the map of my cerebrospinal fluid.
But where is the North?

Figure adapted from "Monsieur Hippopotame" by Tanikawa Shuntarô

# CHAPTER III

# PROTEOMIC TECHNIQUES

This chapter tackles two major techniques involved in proteomic experiments: protein separation techniques and mass spectrometry.

# III. Proteomic techniques

## III.1. Introduction

Proteomics can be defined as the systematic study of the protein content –the proteome- of a given cell, tissue or organism, at a given time and under specific conditions (Wilkins et al. 1997). Proteomic research encompasses the identification, characterization and quantification of proteins and involves various techniques, including biochemical techniques, mass spectrometry and protein chips. In addition, important computing resources are generally required, associated with the use of specialized software to analyze, store and make available the obtained data. Plebani describes in (Plebani 2005) four applications of proteomics: a) protein mining; b) protein expression profiling; c) protein network mapping; and d) protein modification mapping. Protein mining is the cataloging of proteins present in a sample; protein expression profiling involves the identification of proteins differentially expressed as a response to chemical or physical stimuli, or to pathophysiological states; protein network mapping tries to describe how proteins interact with each others; and finally, mapping of protein modifications

Studying proteins is a particularly challenging task. Proteins are neither homogenous nor static, like genes. Genes are confined in the cell nucleus (for Eukaryote organisms), while proteins are present in all cell compartments, on the cell surface and even in extracellular fluids. Gene quantity is constant, while protein concentration may change according to a multitude of variables including developmental, pathological and physiological conditions. Genes are composed of 4 different types of subunits, called nucleotides, while proteins are composed of 20 different types of subunits, called amino acids, and exhibit more diversified biochemical properties than genes. One gene can give rise to several different protein sequences by mechanisms like alternative splicing and post-translational cleavages. Protein sequences are dressed with various ornaments that act upon their 3-D conformation, their function and their turnover. In addition, proteomic analysis techniques must cope with limited sample amounts, as proteins cannot be amplified using polymerase chain reaction-type technologies. Despite these difficulties, Proteomics is a very attractive research field. Much work needs to be done to catalogue protein sequences, structures and functions of numerous species and tissues. Great hopes are placed in Proteomics for discovering potential diagnostic markers and therapeutic targets. For example, techniques like 2-D gel separation, associated with mass spectrometry, allow for the rapid identification of proteins associated with particular disease states.

## III.2. Protein and peptide separation procedure

### III.2.1. Introduction

In proteomic studies, the samples to analyze can contain a huge number of different proteins in various abundances. Protein separation can help in targeting proteins of interest, and therefore alleviates further analysis procedures. In shotgun Proteomics, separation techniques are applied on large mixtures of peptides and are coupled on-line with a mass spectrometer, so that peptides are

continuously introduced into the machine and analyzed. When necessary, multiple separation procedures, based on different physico-chemical properties of proteins and peptides, can be successively performed to obtain better separation capacities. This approach is referred to as 2-D, 3-D and multi-dimensional separation, according to the number of separation steps. The next sections introduce two major classes of separation techniques: chromatography-based techniques, where the analytes migrate due to a mechanical flux, and electrophoresis-based techniques, where the analytes are submitted to an electrical field and migrate because of a difference of potential. A more complete presentation of separation methods can be found in (Scopes 1993).

### III.2.2. Chromatography-based separation techniques

In column chromatrographic separation methods, a solution containing the protein or peptide mixture passes continuously through a column (this is called the mobile phase). The column is packed with material having some binding properties for the proteins (this is the static phase). Changing experimental conditions in the solution (e.g. pH) differently retains or releases one or another type of proteins and causes proteins to elute separately from the column. The nature of interaction between the proteins and the packed material can be of various types. For example, in reversed phase chromatography, proteins are separated based on their hydrophobicity. In this case, the tube is filled with small beads carrying long hydrophobic carbon chains. When the protein solution passes through the tube, proteins stick more or less tightly to the surface of the beads depending of their hydrophobicity. By continuously adding an organic solvent, proteins of higher and higher hydrophobicity get detached and separately leave the column. In ion-exchange chromatography, proteins are separated based on electrical properties. The stationary phase is a negatively charged resin, and the mobile phase is a buffered aqueous solution. Positively charged proteins bind to the resin and are successively eluted out by varying the buffer's pH. In affinity chromatography, proteins are isolated by exploiting specific binding interactions between a protein and a ligand (e.g. an enzyme and its substrate, an antibody and an antigen, and so on). The ligand, immobilized in a matrix is the static phase. The column is loaded with the sample containing the protein mixture. The target protein specifically binds to the ligand while the other proteins leave the column. The bound protein can finally be collected by changing experimental conditions to favour its desorption. In size exclusion chromatography, the proteins are separated according to their size. They migrate through a tube filled with porous material. Large proteins pass easily and are the first to elute, while smaller ones get trapped in the pores and get delayed.

### III.2.3. Separation based on electrophoresis

Electrophoresis is a separation technique that involves the generation of an electric field between two electrodes to move charged molecules through a matrix. One of the most common electrophoresis techniques used with proteins is Sodium Dodecyl Sulfate – PolyAcrylamide Gel Electrophoresis (SDS-PAGE). SDS is a detergent employed to denature proteins, and the polyacrylamide gel is an inert polymer, whose pore size can be controlled to create a continuous gradient into the gel. Proteins are loaded at one side of the gel and migrate, more or less fast and more or less far, through the gel to the opposite side. A possible variation is isoelectrofocusing (IEF) gels, in which the gel contains a pH gradient. In such gels, proteins travel until they reach a place where their net charge is zero. Two-

dimensional gel electrophoresis combines both technologies and separates proteins in a first dimension according to their isoelectric point, and in a second dimension according to their molecular weight. The procedure results into a constellation of spots (up to a few thousands), each of them representing one or a few purified protein types.

## III.3. Mass spectrometry

### III.3.1. Introduction

Mass spectrometry (MS) is a technique whose beginnings date back to the nineteenth century and which is designed to measure mass-to-charge ratios (*m/z*) of gas-phase ions. Mass spectrometers are composed of three elements: a source, a mass analyzer and a detector. The source is the component where the analytes are introduced into the spectrometer and are ionized. The mass analyzer separates the ions with respect to their mass-to-charge ratios, and the detector reports the number of ions that emerge from the analyzer. The next sections describe the different parts of a mass spectrometer. The reader will find more details in Lane's review (Lane 2005).

#### III.3.1.1. Ion sources

To be analyzed by MS, peptides have to be ionized. Two soft ionisation methods, that suit well for non-volatile large biomolecules, are used in mass spectrometry-based proteomics: electrospray ionization (ESI) (Fenn et al. 1989) and matrix-assisted laser desorption ionization (MALDI) (Karas and Hillenkamp 1988).

In a MALDI source, the analytes are embedded into a crystalline matrix. Pulses of UV laser light (typically a nitrogen laser at 337 nm) are absorbed by the matrix, causing vibrational excitation and ejection of the analyte molecules surrounded by matrix components. As the matrix evaporate, analytes are liberated and ionized.

In an ESI source, the sample, in aqueous phase, flows through a needle subjected to high-voltage (1-6 kV). A difference of potential is applied between the needle tip and the inlet of the mass spectrometer, leading to an accumulation of charges of same polarity. Due to electrostatic repulsion, the solvent containing the analytes blow apart into a fine spray of highly charged droplets. The flow is directed through a counter current flow of heated gas, causing the solvent to evaporate and the droplets to shrink, increasing the charge concentration at the droplet's surface. As the electrical charge reaches a critical state (known as the Rayleigh limit), the droplets explode into smaller and lower charged particles. The process of shrinking and explosion is repeated until individually charged analyte molecules are left over. Since the sample is introduced in a liquid state at atmospheric pressure, ESI sources can easily be associated with liquid-phase separation techniques, such as liquid chromatography (LC). Contrary to MALDI, which generates singly charged ions, ESI technique generates singly- and multi-charged ions.

Preference for MALDI or for ESI mass spectrometry depends on several factors. Traditionally, MALDI was used to analyze proteins separated by 2-D gel electrophoresis and combined with TOF mass analyzers. It leads to spectra that are easier to interpret, because they contain only singly charged ions. Another advantage of the technique is that a very small quantity of sample is consumed during the analysis, thus permitting repeated analyses of the same spot. On the other hand, the susceptibility of ESI to produce multiply charged analytes allows the detection of peptides with masses exceeding the operating mass range of the mass spectrometer. An advantage of ESI source is that it involves the generation of peptide ions from aqueous solution and therefore can be easily coupled to liquid chromatography and capillary electrophoresis separation systems.

### III.3.1.2. Mass analyzers

Mass analyzers are instruments that separate charged molecules according to their mass-to-charge ratio *m/z*, where *m* is the mass of the molecule and *z* is its number of elementary charges. Four basic types of mass analyzers are currently in use in proteomic research: ion trap (IT), time-of-flight (TOF), quadrupole (Q) and ion cyclotron resonance (ICR) analyzers.

Ion traps mass analyzers are devices that can store charged molecules for a long time. Ions are trapped in time varying electric potentials $\sim U+V\cos(\omega t)$, produced by a ring electrode and two end cap electrodes in a space of 2-3 $cm^3$ filled with dilute helium. Ions of different *m/z* values enter the trap at one of the cap electrode and remain trapped, oscillating at frequencies that are related to their *m/z* values. By changing *U*, *V* and ω, ions of certain *m/z* become turn-to-turn exited and are ejected from the opposite end cap.

In time-of-flight mass analyzers, ions are accelerated by a potential down a field-free flight tube until they impact a detector. All ions in the source are given the same initial amount of energy, but the time needed to travel the distance between the source and the detector is dependent on their *m/z* values, which can be calculated using the kinetic energy equation, given the tube length and the measured flight times.

Quadrupole mass analyzers (Figure III-1) consist of four parallel and symmetrically arranged metallic rods. One couple of opposite rods has a positive electrical potential $+(U+V\cos(\omega t))$ while the other couple of opposite rods has a negative potential $-(U+V\cos(\omega t))$. Ions that traverse the field along the central axis of the rods oscillate. Depending on the values *U*, *V* and ω, ions of a certain *m/z* are either destabilized and deviated, or enter in resonance and follow a stable trajectory through the quadrupole to the detector.

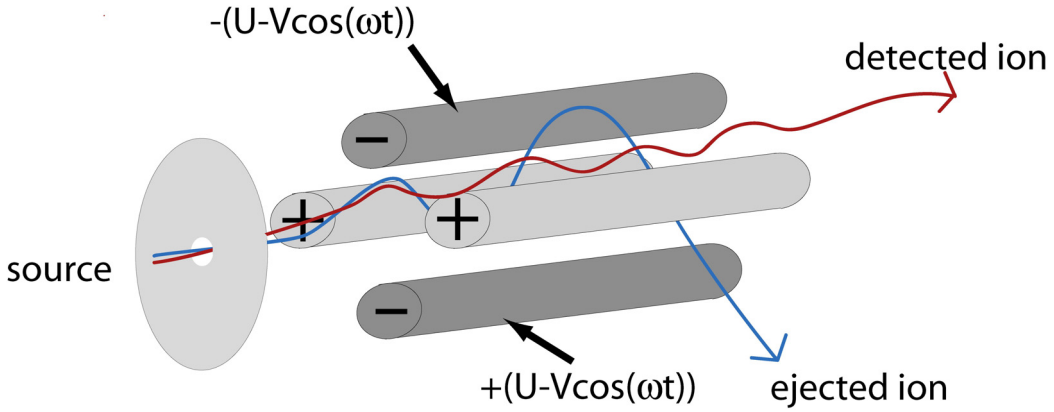



*Figure III-1: Schematic representation of a quadrupole rod system*

Ion cyclotron resonance mass spectrometers consist of a cubic cell inside a strong magnetic field. Injected ions rotate around the magnetic field with a frequency typical for their *m/z*. By varying the electric fields, changes in the ion frequency of rotation can be measured and converted into *m/z* using Fourier transformation.

These analyzers differ in speed, operating mass range, resolution, accuracy and cost. Resolution indicates the ability of the instrument to discriminate between ions with different m/z values. It is defined by the m/z divided by the peak width at half its maximum height. Mass accuracy refers to the extent to which a mass analyzer reflects the correct m/z values. It is usually measured in daltons or, better, in parts per million (ppm).
The most accurate mass spectrometer is the FT mass spectrometer, with a resolution of more than 100'000 and an accuracy of a few parts per million for a mass range of 250-1000 [Da]. Unfortunately, FT mass spectrometers are also more expensive and technically demanding. Modern TOF and quadrupole instruments have a resolution of 10'000 and can easily reach accuracies around 10 ppm. Despite a modest resolution and accuracy (they barely resolve the isotopes of doubly charged ions, which are separated by 0.5 Da), ion traps mass spectrometers are particularly appreciated, because they are robust and relatively inexpensive.

### III.3.2. MS/MS analysis

Tandem mass spectrometry is achieved by performing two mass analyses, either "in space" or "in time". The first MS analysis step is used to measure ions according to their *m/z*. Then, ions (called precursors) are selected and fragmented. The resulting fragment ions are separated in the second mass analysis step with respect to their *m/z* values and are recorded as a fragment ion spectrum (MS/MS spectrum). For "in space" configurations, such as triple quadrupole (TQ), quadrupole/time-of-flight (Q-TOF) or time-of-flight/time-of-flight (TOF-TOF), the primary and secondary analyses are performed sequentially as ions travel through the instrument. For example, in TQ configuration, the first quadrupole selects the peptides to fragment according to their *m/z*. The second quadrupole is used to fragment peptides by collision with a dilute inert gas (e.g. He). Finally, the fragment masses are scanned by the third quadrupole. A possible variation consists in replacing the last quadrupole by

a TOF analyzer (Q-TOF). For “in time” configurations, such as quadrupole ion trap (Q-IT), they are performed consecutively within the same analyzer (see Figure III-2).
For a detailed review, see (Aebersold and Mann 2003).

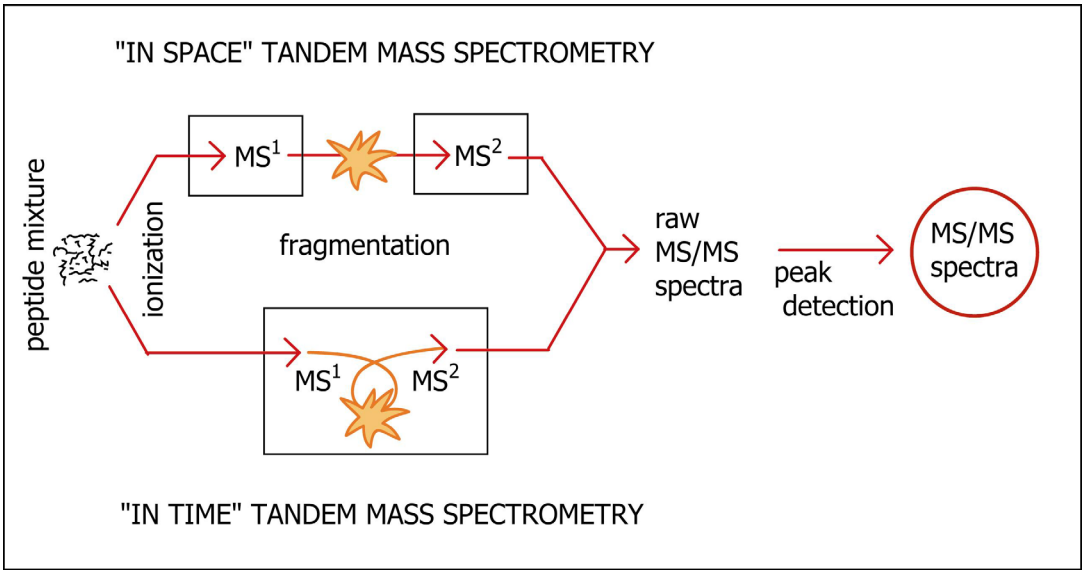



*Figure III-2: Tandem MS: “in time” and “in space” configurations*

### III.3.2.1. Fragmentation

Several techniques to fragment ions are available. According to Roepstorff’s nomenclature (Roepstorff and Fohlman 1984), the product ions are denoted as *a, b,* and *c*, when the charge is retained on the N-terminal side of the fragmented peptide, and *x, y* and *z* when the charge is retained on the C-terminal side. As shown in Figure III-3, ion types differ by the position of the fragmentation in respect with the peptide bond.
In “Collision-induced fragmentation" (CID), peptides are fragmented by collision with an inert gas yielding to mainly a, b and y ions. An alternative is "electron capture dissociation" (ECD), which uses a beam of electron and produces mainly c and z ions. An advantage of ECD is that labile modified groups are not dissociated upon fragmentation of the peptide backbone, thus allowing their mapping (Zubarev et al.).
Additional fragment ions may also be generated, including internal fragments formed by breakage of two peptide bonds, as well as side-chain specific ions (denoted as d, v and w) formed by the loss of all or parts of side-chains (Johnson et al. 1988).

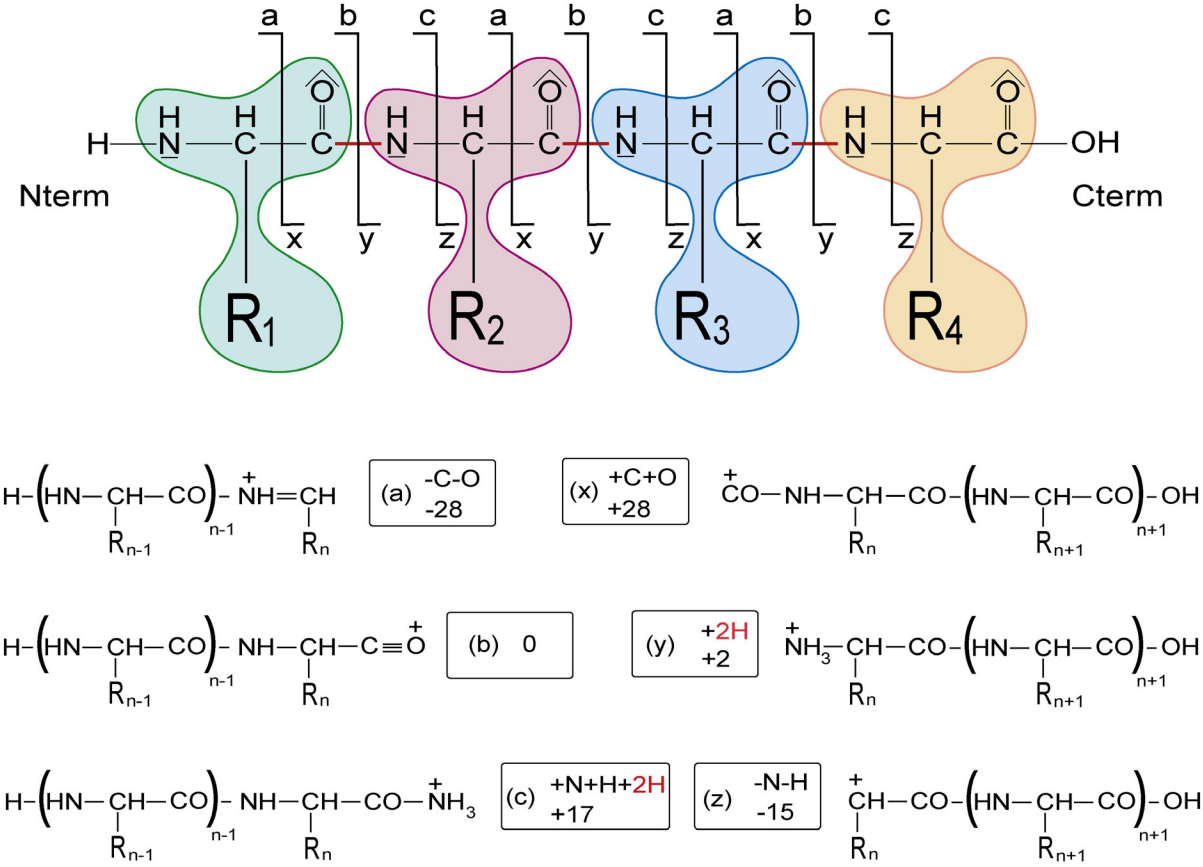



*Figure III-3: Fragmentation*
*Various possibilities of fragmentation for a tetrapeptide (top) and the produced fragment structures (bottom). The mass of an ideal N-terminal fragments is computed by adding the mass of the N-terminal group (H) and all amino acid nominal masses before the cleavage position. The mass of an ideal C-terminal fragments is computed by adding the mass of the C-terminal group (OH) and all amino acid nominal masses after the cleavage position. Each ionic type is characterized by an offset that represents the mass difference in Daltons between the observed mass and the corresponding N- or C-terminal ideal fragment. For example, b-ion type offset is 0 [Da] because the mass of a b-ion type corresponds exactly to an ideal N-terminal fragment, while a-ion type offset is –28 [Da] because a-ion types lost a carbonyl and an oxygen atom compared to an ideal N-terminal fragment.*

Numerous studies have investigated the fragmentation pathway and support a model called “mobile proton hypothesis” (Dongré et al. 1996) (Wysocki et al. 2000) (Figure III-4). This model states that under low-energy collisional activation conditions, most fragmentation pathways are triggered by protonation of the amide nitrogen or carbonyl oxygen at the cleavage site (Jonsson 2001).

The presence –or absence– of mobile proton greatly influences the fragmentation quality of the peptide. Basic amino acids in the peptide act as traps for protons. Consequently, when the number of protons on the peptide is lower or equal to its number of strong basic amino acids (Lys, Arg, and His), no mobile proton will be available to trigger fragmentation, resulting in poor quality spectra. On the contrary, when the number of charges is larger than the number of basic sites, the supplementary protons migrate along the backbone, possibly initiating fragmentation at every amide site, thus giving

rise to highly informative spectra. In addition, some amino acid residues may greatly influence the fragmentation at their N-terminal or C-terminal side. For example, enhanced cleavage is observed at proline C-terminal side (due to basic properties) while it is strongly decreased at its N-terminal side (due to steric hindrance) (Tabb et al. 2003c).

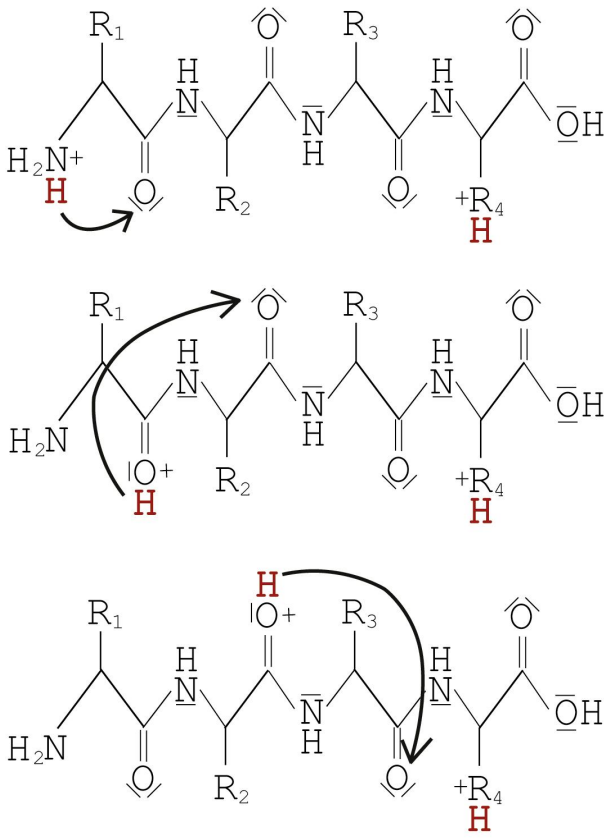


*Figure III-4: Migration of a mobile proton*
*The mobile proton migrates from the N-terminal amine group to an amide carbonyl oxygen along the peptide backbone.*

### III.3.2.2. MS/MS spectra

MS/MS spectra represent peptides that are produced by proteolysis of a protein prior to MS/MS analysis. Raw MS/MS spectra are composed of a raw signal and need to be processed to transform the signal into generic peak lists. Peak detection includes centroiding (reducing the raw signal into discrete values), noise filtering, calibration (shift of the *m/z* scale), and deisotoping (removal of isotopic peaks). Examples of algorithms are given in (Gentzel et al. 2003). Processed MS/MS spectra are finally composed of the mass and charge state of the precursor peptide, as well as of a list of peaks. Each peak is characterized by two values: the measured fragment mass-to-charge ratio (*m/z*) and an intensity value that represents the number of detected fragments of the given *m/z*. The number of peaks composing an MS/MS spectrum varies from about ten to several hundreds depending on factors like peptide length, fragmentation quality, mass spectrometer type or parameters used to extract the peaks from the raw spectrum. Interpreting a spectrum is not a straightforward procedure,

as many parameters have to be taken into account. Thus, the measured *m/z* of a peak depends among other things on its ionic type, on the number of charges carried by the fragment and on its isotopic distribution. In addition, a fragment can lose specific molecules (like ammonium and water molecule), or its interpretation can be complicated by the presence of supplementary molecules (such as post-translational modifications). Also, the calibration and internal error of the mass spectrometer have to be taken into account. Finally, certain peaks may represent noise and disrupt the spectrum interpretation by diluting informative peaks, while some fragmentation positions may not be represented at all.

Spectra obtained by tandem mass spectrometry contain series of peaks that come from successive fragmentation positions in the peptide sequence. This is a key property of MS/MS spectra, since information about the peptide sequence can be inferred from the mass differences between peaks (Figure III-5).

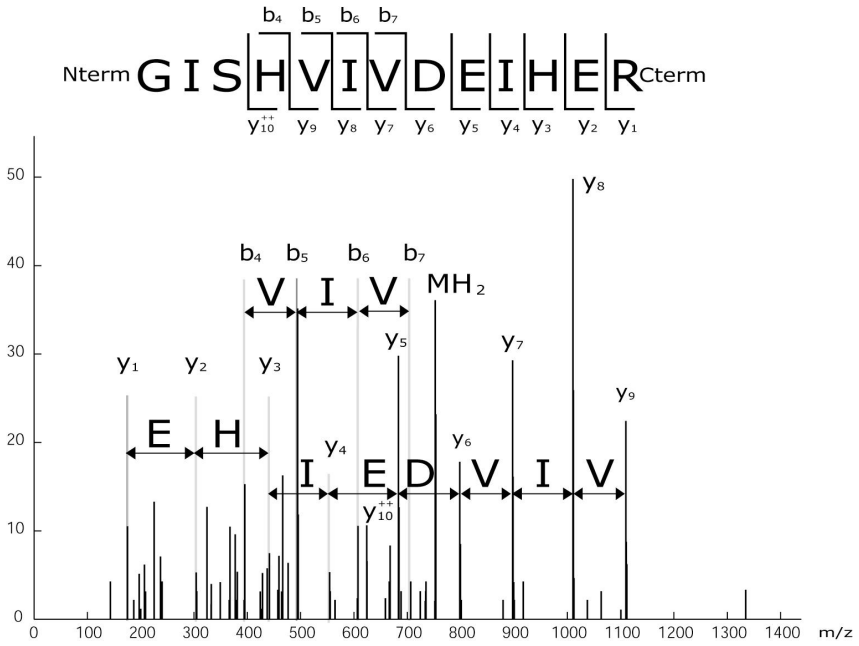
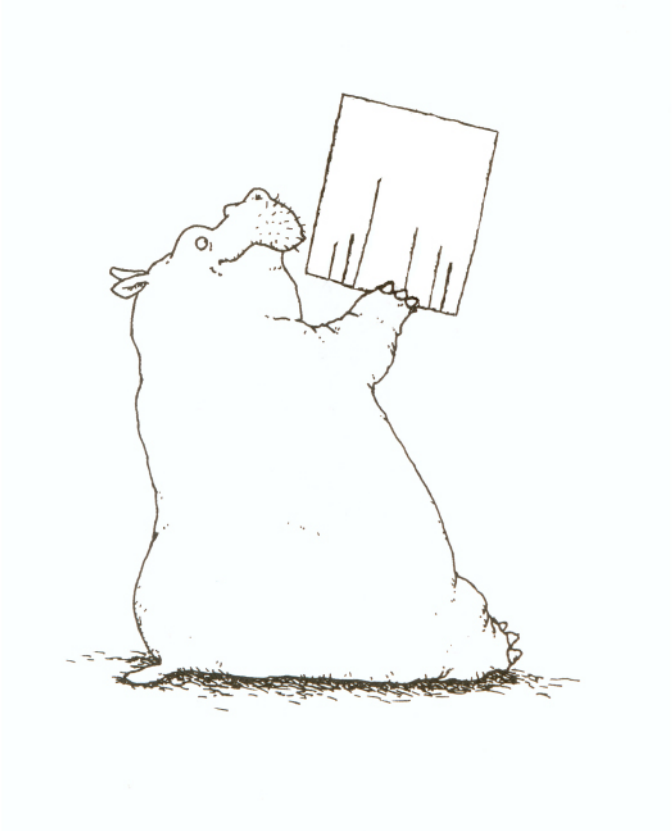



*Figure III-5: An annotated MS/MS spectrum*
*Information about the peptide sequence can be inferred from peak differences. Peaks that are not labeled should nevertheless not be systematically associated with noise. Many of them originate from the peptide sequence and could be correctly interpreted by taking into account more ion types. A peak series is a set of peaks of similar ion-types that represent consecutive fragmentation positions on the peptide sequence (e.g. the peaks b4, b5, b6 and b7 represent a b-ion series).*



On m'a demandé un jour la différence entre un spectre et des brins d'herbe.
J'ai répondu: "L'un est la nourriture du corps, l'autre de l'esprit".

One day somebody asked the difference between a spectrum and grass blades.
I answered: "Both are food. One for body and one for soul."

Figure adapted from "Monsieur Hippopotame" by Tanikawa Shuntarô

# CHAPTER IV

# MASS SPECTROMETRY BASED IDENTIFICATION

This chapter presents existing protein identification approaches using mass spectrometry. It shortly introduces MS-based identification, and explains in greater detail various MS/MS identification methods.

# IV. Mass spectrometry-based identification

## IV.1. Introduction

Thanks to advances in mass spectrometry, new approaches appeared in the last 20 years for the identification of proteins or peptides present in biological samples. First, in the eighties, strategies were developed for the sequencing of peptides from MS/MS spectra; then, the growth of protein and genomic databases, brought forth two major identification approaches: "peptide mass fingerprinting" (PMF) in the nineties, and soon after that, "peptide fragment fingerprinting" (PFF). These three mass spectrometry-based identification approaches are now routinely used.
The next sections introduce the PMF identification method, and review in detail *de novo* and PFF identification approaches.

## IV.2. MS protein identification

The PMF approach (see Figure IV-1) has been described for the first time in 1993 by five independent groups (Henzel et al. 1993;James et al. 1993;Mann et al. 1993;Pappin et al. 1993;Yates, III et al. 1993). It involves digesting a protein of interest (purified from a mixture) using a site-specific proteolytic enzyme and then measuring the masses of the obtained peptides by MS. One of the favorite enzymes used for mass spectrometry-based protein identification is trypsin. This protease can be found naturally, for example in the small intestine of Mammalians, where, together with other proteases, it participates to the degradation (digestion) of proteins into amino acids. Trypsin specifically cleaves protein sequences at carboxyl side of lysine and arginine residues. The median length of the produced peptides is about 15 residues. This corresponds to a mass of about 1600 Daltons that suits well the optimal operating mass range of many mass spectrometers (typically up to 4000 Daltons for most instruments). The masses of the obtained peptides are reported as an MS spectrum. The experimental spectrum is compared with theoretical ones computed from protein sequences stored in databases and "in silico" digested using the same cleavage specificity of the protease employed in the experiment. This procedure roughly amounts to counting overlapping masses between the experimental and theoretical spectra, leading to "similarity scores" for each candidate protein. The candidate proteins are then sorted according to their score. The top-ranked protein is considered as the identification of the spectrum.

The key step of the procedure lies in the scoring function. The latter must take into account many factors to produce a robust score, like dissimilarities in the peak positions due to internal or calibration errors, expected peak intensities, noise, contaminant or missing peaks, presence of post-translational modifications, and so on. A variety of different scoring functions have been implemented in various algorithms. Appendix A2 gives the names and URLs of a certain number of available PMF tools. FragFit (Henzel et al. 1993), PeptideSearch (Mann and Wilm 1994), peptIdent (Wilkins et al. 1997) and PepFrag (Fenyo et al. 1998) use a simple score based on the number of common masses between the experimental and theoretical spectra. MOWSE (Pappin et al. 1993) exploits a scoring which accounts for the non-uniform distribution of protein and peptide molecular

weights in databases. Similar score schemes are exploited in MS-Fit (Clauser et al. 1999), Mascot (Perkins et al. 1999) and ProFound (Zhang and Chait 2000). SmartIdent (Gras et al. 1999) uses Genetic Algorithms to learn the scoring parameters and Aldente exploits Hough transform to determine the mass spectrometer deviation, to realign the experimental masses and to exclude outliers.

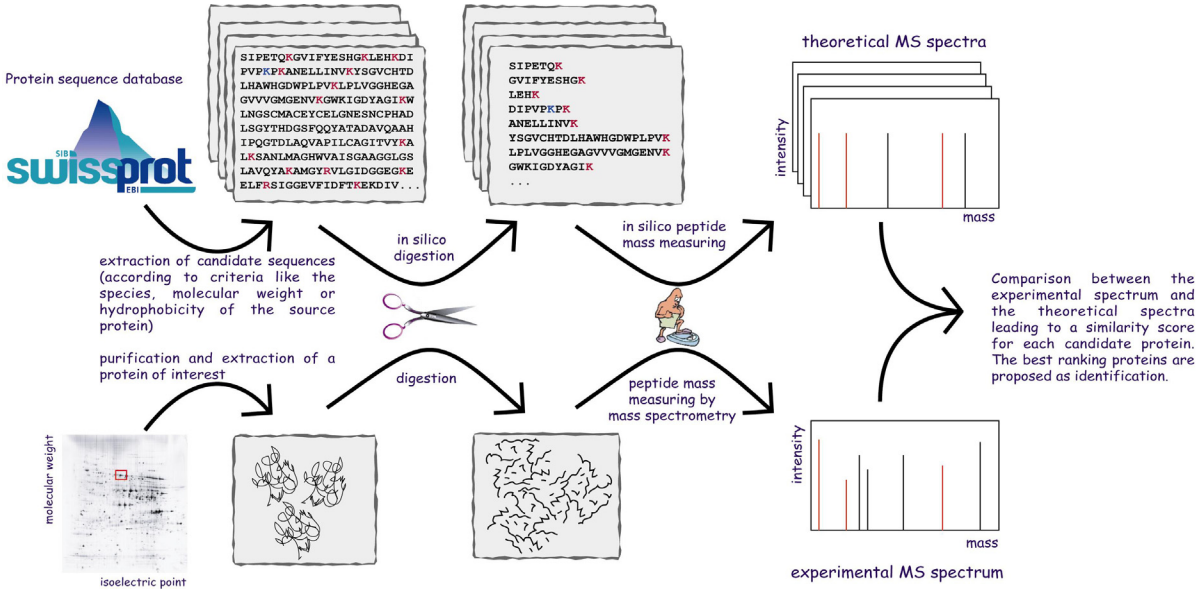



*Figure IV-1: Protein identification by peptide mass fingerprinting.*
*Candidate protein sequences extracted from a database are digested in silico according to a protease specificity. Theoretical MS spectra are constructed and compared to the experimental MS spectrum, leading to a similarity score for each* candidate *protein. The candidate proteins are then sorted according to their score. The top-ranked protein is considered as the identification of the spectrum.*

## IV.3. MS/MS protein identification

### IV.3.1. Introduction

PMF identification is sometimes not appropriate. For example, the sample to be analyzed must contain one (at worst a few) purified protein (typically, a 2D gel spot). A protein mixture gives an MS spectrum with several mixed signals and it is difficult to determine which protein each peak belongs to. Moreover the proteins at the origin of the spectrum (or homologous proteins with nearly 100% identical sequences) have to be represented in the database. Consequently it is not recommended to search genomic databases (especially databases of "expressed sequence tags") because they contain errors and often only partial sequence information.

An alternative to PMF is the tandem mass (MS/MS) analysis that produces peptide specific spectra in which the information about the peptide sequence is present. MS/MS-based identification presents several advantages over PMF. It is possible to work with complex peptide mixtures, to search in homologous databases and, when enough coverage is obtained, it provides detailed information about the peptide sequence and about possible modifications and mutations. Last but not least, MS/MS

identification does not require all the peptides of a given protein to be confirmed to achieve confident identification. But MS/MS identification is also the subject of difficulties, notably in every situation where the peptide sequence does not correspond exactly to any candidate peptide from the database. Actually, a great majority of MS/MS spectra collected during an experiment cannot be confidently matched to theoretical peptides. Possible reasons are listed below.

a) Non-peptide spectrum
The spectrum may originate from a non-peptide contaminant when using, for example, contaminated material during separation.

b) Co-eluting peptides
The spectrum may originate from several co-eluting peptides and may therefore contain the signal of more than one peptide, disturbing identification algorithms.

c) Spectrum quality
The spectrum may be too noisy, or may be issued from unusual fragmentation (due for example to the non availability of a mobile proton to trigger fragmentation).

d) Incorrect precursor mass/low accuracy precursor mass
Non-accurate or incorrect precursor mass may cause the identification to fail, particularly if the identification algorithm applies a precursor mass-based filter to select candidate peptides from the database.

e) Novel protein/alternative splicing
The spectrum may derive from a novel protein that is not present in the database or from a protein issued from alternative splicing that is not annotated in the database.

f) Missed or exotic cleavage sites
During the digestion process, cleavage-sites may be skipped by the protease, leading to fragments composed of several peptides linked together end to end (missed-cleavages). Or the protease may cut at a wrong place, resulting in peptides with unpredictable ends (non specific cleavages).

g) Transpeptidation
Transpeptidation involves the grafting of a peptide fragment on another one, probably as a consequence of a side activity of trypsin. Schaefer et al. (Schaefer et al. 2005) recently reported different examples of transpeptidation observed during a mass spectrometry-based proteomic experiment. They notably observed one and two N-terminal amino acid addition, as well as combination of two peptides originally located in different regions of a protein. Such events greatly complicate the task of identification algorithms.

h) Mutations, polymorphisms, amino acid modifications and errors in databases
Unexpected modifications and mutations, as well as non-annotated polymorphisms (allelic variations) in the protein sequences to be analyzed may complicate the identification procedure, because they cause part of the peaks to be shifted in the spectrum compared to their expected positions. Similarly,

when the candidate protein sequences are obtained by automatic translation of genomic data, DNA sequencing errors may more or less strongly affect the obtained protein sequence, from amino acid replacements ("missense"-type errors) to changes of large portions of the protein sequence (frameshift-type errors) or to truncated protein sequences (nonsense-type errors).

Luckily, software tools have been developed for most of these issues (see Appendix A3), although none of them is currently able to handle all issues at once. Some are specialized in reducing the number and complexity of MS/MS spectra while increasing their quality; others have been specifically designed to handle unexpected modifications or mutations; and some split the identification into several stages and combine different approaches.

### IV.3.2. Increasing spectra quality before identification

Preprocessing procedures are often applied to the peak lists before performing the identification algorithm, with the aim to increase the quality of the spectrum. Such procedures include filtering background noise and removing isotopic peaks. If present, the peak representing the precursor ion is also deleted from the peak list. The precursor charge-state can be confirmed using information from the fragment ions (Sadygov et al. 2002) (Colinge et al. 2003a). Spectra that are likely to derive from the same peptide can be merged into a higher quality consensus spectrum. This allows avoiding multiple analyses of a same peptide, while improving signal-to-noise ratio and mass accuracy. Nevertheless, it should be noted that information is lost during the clustering procedure. The criterion used to decide if two spectra originate from a same peptide is based on a similarity measure. For example, NoDupe (Tabb et al. 2003a) represents each spectrum as a vector in a multidimensional space and uses as similarity measure the "contrast angle" (also called "dot product") between pairs of spectra with similar precursor masses. Angles near zero degree are found for similar spectra. Another example of similarity score applied to spectrum clustering was proposed by Pevzner et al. in 2000 (Pevzner et al. 2000). This method applies the Needleman and Wunsch sequence alignment algorithm to spectral masses. As it is also suited for open-modification searches, the algorithm will be described in more details in Chapter V.

### IV.3.3. « De novo sequencing » versus « peptide fragment fingerprinting »

Two main approaches are taken to identify an MS/MS spectrum (Figure IV-2). The first one, named *de novo* sequencing, consists in inferring knowledge about the peptide sequence independently of any information extracted from a pre-existing protein or DNA database. Then, the inferred complete or partial sequences are compared to theoretical sequences using specifically developed sequence similarity search algorithms.

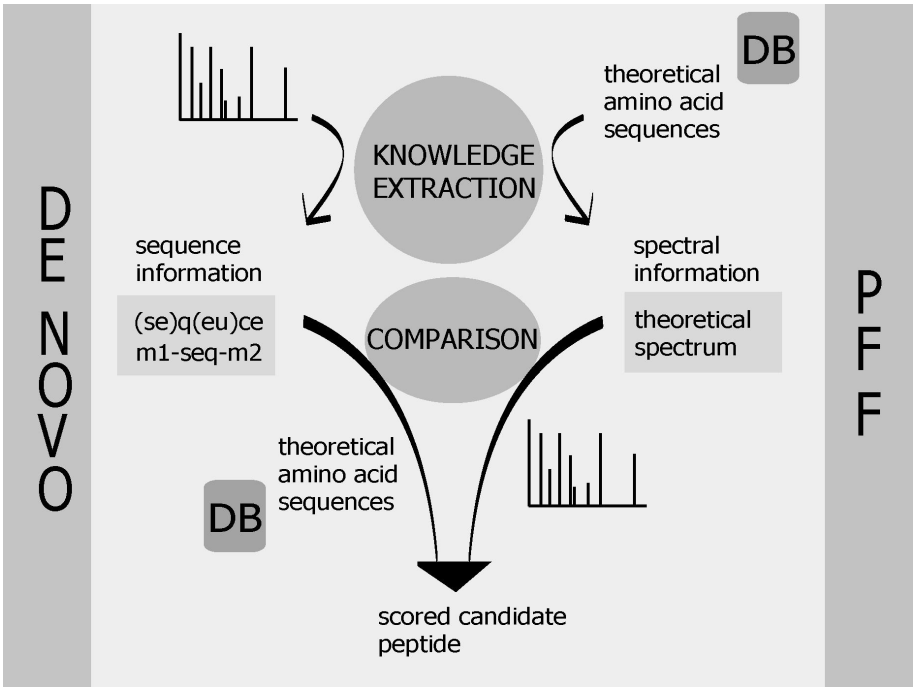



*Figure IV-2: Conceptual representation of de novo and PFF approaches.*
*Both methods contain a knowledge extraction phase and a comparison phase. In the de novo approach, knowledge is directly extracted from the spectrum as de novo complete or partial "sequences" that are then compared to theoretical sequences using a sequence similarity search algorithm. The PFF approach extracts knowledge from the database by building theoretical spectra and then compares the theoretical spectra to the experimental spectra.*

The second major way to identify a spectrum has recently been designated as "peptide fragment fingerprinting" (PFF) (Blueggel et al. 2004), by analogy to "peptide mass fingerprinting" (PMF). In the PFF approach, spectrum analysis is specifically performed for candidate peptides extracted from a database by building theoretical spectra from the peptide sequences and measuring the degree of similarity between the experimental and the theoretical spectra. For a given scoring function, the highest scoring theoretical peptide is then taken as the one amongst all candidates that best represents the experimental spectrum.

### IV.3.4. Methods based on *de novo* sequencing

*De novo* sequencing-based identification starts by inferring sequence information from the experimental MS/MS spectrum. Before the nineties, when protein and genomic databases were still at an embryonic stage, the obtained *de novo* sequences were used to design oligonucleotide probes for gene cloning. But with the growing number of sequenced genomes, software tools have been conceived to correlate *de novo* sequences with theoretical sequences for identification purpose.
Since they do not use database information during spectrum interpretation, *de novo* sequencing algorithms work in a search space composed of the set of all possible sequences that can be represented by the spectrum without any other restriction than peak disposition. Due to the size of this search space, *de novo* sequencing methods are disadvantaged compared to PFF methods. They require

spectra of higher quality with smaller fragment errors and a more or less continuous signal, or at least high-quality signal for several adjacent amino acids. Spectra with unusual fragmentation will be very hard or even impossible to analyze. Despite these disadvantages, *de novo* methods may overcome PFF methods, notably when searching genomic databases subjected to sequencing errors, when searching databases composed of homologous sequences, and when analyzing a spectrum that originates from a mutated protein or variant. In effect, *de novo* algorithms naturally extract sequences from the spectrum that include the amino acid replacements, which are then handled by the similarity search algorithm by allowing mismatches between the *de novo* sequences and the database sequences. But *de novo* sequencing algorithms are not well adapted to deal with the presence of modifications on the peptide. For each putative modification, the corresponding modified amino acid mass must be added to the pool of existing amino acids. This results into an expansion of the search space, and thus, this solution can only be used for a couple of modifications of interest, or for chemical modifications that were deliberately introduced during sample preparation, such as cysteine carbamidomethylation. *De novo* sequencing algorithms can be separated into two classes: the first one applies a "pseudo" PFF approach using a database of generated sequences, while the second one exploits the principle of peak succession to extract sequence information from the spectrum.

#### IV.3.4.1. The "pseudo" PFF approach

Early *de novo* sequencing algorithms (Hamm et al. 1986;Sakurai et al. 1984) consisted in building a "pseudo" sequence database on-the-fly: the sequences were generated by determining all possible amino acid compositions with a total mass matching the experimental precursor mass, and then, for each composition, by determining all possible amino acid permutations. Subsequently, as for a PFF-approach, theoretical spectra were computed from the "pseudo" sequences and common peaks between the experimental and theoretical spectrum were counted. The theoretical sequences with the highest scores are the most likely to represent the original peptide. The main drawback of this approach is combinatorial complexity as the number of possible sequences increases exponentially with the precursor mass. This issue can be handled by using additional information, notably about the amino acid composition of the peptide. Thus, in an approach by Spengler (Spengler 2004), a drastic reduction in the number of "pseudo" sequences was achieved by using more accurate precursor masses and the presence of immonium ions in the spectrum. Another strategy was chosen by Heredia-Langner et al. (Heredia-Langner et al. 2004), which proposed to build candidate sequences using a genetic algorithm rather than systematically enumerating all amino acid combinations. Finally, Ma et al. (Ma et al. 2003) computed a set of 10'000 good scoring candidate sequences using dynamic programming, and then reevaluated the candidate sequences using a more stringent scoring scheme. Their tool, named PEAKS, has the particularity to provide a confidence score to each individual amino acid of the candidate sequences.

#### IV.3.4.2. The peak succession approach

Since the mid-eighties the tendency has been to use an incremental approach: candidate sequences are built in an iterative way, amino acid by amino acid, until complete sequences that account for the precursor mass are obtained. During sequence building, only partial sequences whose extensions are validated by fragment ions in the spectrum are retained for further extension. In this way, large

subsets of permutations are discarded from analysis, contrary to the previous approach in which every possible sequence is systematically compared to the spectrum. This method is therefore much more sensitive to spectrum quality. Thus, a two amino acid gap in the spectrum (or in other words two successive non-fragmented positions on the peptide) causes the correct sequence to be discarded. Early implementations differ from each other by small variations. For example, Zidarov et al. (Zidarov et al. 1990) use the precursor mass as well as information about the amino acid composition deduced from observed immonium ions to limit the search space; Ishikawa's approach (Ishikawa and Niwa 1986) generates a pool comprising all possible permutations of three amino acids to initiate the extension process and allows multiple amino acid extensions; SEQPEP (Johnson and Biemann 1989) includes information from side-chain losses to differentiate leucine from isoleucine, two isobaric amino acids; Scarberry et al. (Scarberry et al. 1995) use a neural network to assign specific ion-types to the observed fragment ions before starting the iterative sequencing.

In 1990, Bartels (Bartels 1990) coined the term "spectrum graph", which clearly illustrates the principle of peak succession in MS/MS spectra. The graph structure has since been widely adopted and refined in several *de novo* methods (Dancik et al. 1999; Fernandez-de-Cossio et al. 1995; Hines et al. 1991; Taylor and Johnson 1997). Formally, a spectrum graph is a directed acyclic graph, whose vertices correspond to masses of putative N-terminal fragments. Vertices that differ by the mass value of one or more amino acids within a given error margin are linked by an edge labeled with the corresponding amino acids. Each path in the graph defines a sequence that is consistent with the peak succession in the spectrum. Therefore, contrary to early "pseudo" PFF approaches, only a subspace of all possible amino acid arrangements is considered for the search. Candidate sequences are built by traversing the graph from low mass vertices to high-mass vertices following available edges. As a path exploration progresses, a score representing the adequacy between the parsed subsequence and the spectrum is computed. Various algorithms were proposed for parsing the graphs and scoring the paths. Hines et al. (Hines et al. 1991) use a recursive parsing procedure. The authors of SeqMS (Fernandez-de-Cossio et al. 1995) propose using the Dijkstra algorithm "or any other single-source, shortest-path algorithm". They restrict the parsing to promising area of the graph by using a score criterion involving the maximum scores of all paths leading to each vertex. By this mean, they drastically reduce the execution time, since they avoid the calculation of a huge number of sequences. Dancik et al. (Dancik et al. 1999) describe a scoring method based on the likelihood ratio between two models: the first one assumes that the peaks are a result of a peptide's fragmentation, and the second one assumes that they are the result of a random process. Soon after, Chen et al. (Chen et al. 2001) provide a dynamic programming algorithm to extract the highest scoring sequences from the spectrum graph, as well as sub-optimal ones (Lu and Chen 2003b).

It should be noted that *de novo* algorithms typically try to infer the whole peptide sequence. Thus, methods based on peak succession build the inferred sequence in an iterative way until the precursor mass is reached. When a spectrum graph is used, it is typically parsed from the first node (the empty sequence) to the last node (the complete sequence). Missing fragmentation positions are handled by using combinations of two or three amino acids. But when more than a few consecutive fragmentation positions are missing in the spectrum, or in case of unexpected modifications, the correct path is split into two (or more) sections, which are (or are not) connected by alternative paths. The extraction process may therefore stop before completing the sequence, or produce a sequence that contains wrong amino acid sections.

### IV.3.4.3. Database-search algorithms for data obtained by *de novo* sequencing

Once sequence information has been extracted, the *de novo* sequence is correlated with theoretical sequences using database-search algorithms. Without surprise, there are numerous programs that propose the use of sequence information to extract theoretical peptides or proteins from a database. MS-Seq (Clauser et al. 1999) works on a list of masses corresponding to given ion type series (the masses do not have to be all contiguous). MS-Pattern, from the same authors, performs a text-based search with regular expression syntax. It accounts for mutations and database errors by allowing mismatches between the input sequence and the theoretical ones. PeptideSearch (Mann and Wilm 1994) also accepts regular expressions, but in addition it can search the database with "sequence tags", allowing either one of the flanking masses, or the sequence to mismatch. Several tools are based on the well known sequence similarity search algorithms BLAST (Altschul et al. 1990) and FASTA (Pearson and Lipman 1988) and modified BLOSUM (Henikoff and Henikoff 1992) and PAM (Dayhoff et al. 2005) matrices accounting for indistinguishable amino acids; MS-Blast (Shevchenko et al. 2001) for example is a BLAST-based protocol for using BLAST with *de novo* sequences, while FASTS (Mackey et al. 2002) is based on the FASTA algorithm; MS-Blast takes as input n *de novo* sequences assumed to be of the same protein and concatenates them into n! query sequences. Each permutation is then individually submitted to the BLAST program. FASTS is similar and allows searching the database with *de novo* sequences of unknown order. CIDentify (Taylor and Johnson 1997) was specifically written for database searching with *de novo* sequences obtained with Lutefisk97 and is also based on the FASTA algorithm. The *de novo* sequences reported by Lutefisk97 are aligned to the database sequence, and the best alignment is then reprocessed to resolve possible isobaric combinations of amino acids. OpenSea (Searle et al. 2004) is designed to align sequences reported by PEAKS. It initiates the alignments by matching tags composed of unambiguous amino acids to theoretical sequences, and then extends the alignments using a "breadth-first-search" approach based on mass correspondence between matching amino acids or groups of amino acids. Figure IV-3 shows two mass-based alignment examples, one obtained with CIDentify, the other one with OpenSea.

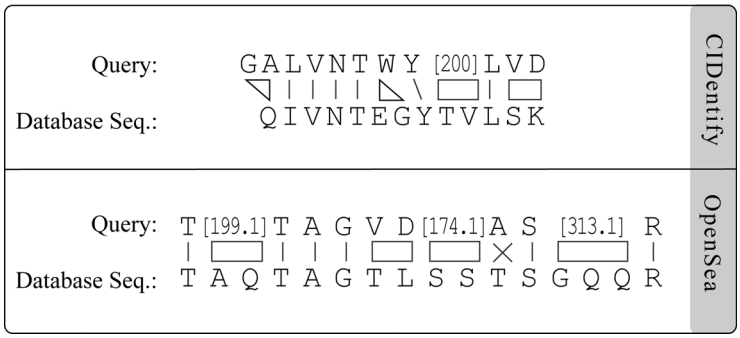



*Figure IV-3: Alignments by CIDentify and OpenSea*
*Both programs handle mass equivalencies: leucine is matched with isoleucine, and combinations of two amino acids are matched with single amino acids or with other combinations of two amino acids. Bars represent amino acid matches, boxes are matches of combinations of amino acids, and crosses represent amino acid mismatches (Figure adapted from (Taylor and Johnson 1997) and (Searle et al. 2004)).*

Mascot (Perkins et al. 1999) requires each submission to start with the precursor mass of the experimental spectrum, to which additional high-confident information are associated, such as partial sequence(s), amino acid composition or ionic fragment(s). Finally, MS-Shotgun (Huang et al. 2001) and MultiTag (Sunyaev et al. 2003) have been designed to analyze the output of multiple sequence database searches with the aim to identify homologous proteins.

## IV.3.5. Methods based on peptide fragment fingerprinting (PFF)

The apparition and expansion of protein sequence databases made *de novo* sequencing in most situations unnecessary. Why waste time and resources by considering all possible amino acid sequences when databases report the sequences that actually occur in Nature? The peptide-sequencing problem became soon changed to a database-matching problem. By exploiting information from the database during the spectrum interpretation, PFF methods restrain the search to a subset of the search space. This results into better exploration capacities and consequently, the method is generally more efficient than the *de novo* approach, as it allows interpreting the spectrum specifically (then optimally) for each candidate peptide. PFF methods strongly rely on a scoring function that evaluates the correlation between the experimental spectrum and the theoretical peptides. Many identification algorithms based on a PFF approach have been developed. Variations can be found at every step: in the way the candidate peptides are chosen from the database or the virtual spectra are modelled from theoretical amino acid sequences, as well as the way to score the similarity between the experimental spectrum and the virtual spectra, or to validate the confidence in the resulting identifications.

### IV.3.5.1. Choosing candidate peptides from the database

PFF algorithms typically produce candidate peptides by in silico “digesting” theoretical protein sequences. Cleavage rules depend on the type of enzyme used for proteolysis during sample preparation. For example, if the enzyme was trypsin, the algorithm cleaves the protein sequence after each lysine (K) and arginine (R), unless the next amino acid in the sequence is proline. Generally, the user can choose to allow skipping one or two cleavage sites to account for peptides with one or two missed-cleavages. Non-specific cleavage can be taken into account by presenting as candidates all possible peptides that match the spectrum precursor mass within a given range, without taking into account any specific cleavage rule, although this results into a significant increase in computing time (Craig and Beavis 2004). Lu and Chen (Lu and Chen 2003a) proposed a method for database searching without cleavage site restrictions. They used a generalized suffix tree structure to index the whole set of database sequences, thus allowing direct access to all possible subsequence of any protein without enzyme specificity restriction.

Most often, PFF algorithms apply filtering criteria on the database sequences. The aim is to reduce the analysis to a small fraction of the database that contains the correct peptide with high probability. Peptide sequences that go through the filter are named “candidate peptides”. Database filters reduce the computing time, allow for more sophisticated and computationally intensive scoring schemes, and diminish the possibility of a high score being achieved by chance. Various filters may be used with PFF methods. The user may specify the species of the sample, thus avoiding parsing the whole taxonomy range of the database. The measured precursor mass of the spectrum is also commonly

used as a filter, but this may hamper the identification in the following cases: a) when the precursor mass is incorrect due to a false assignment of the precursor charge state; b) when the precursor mass error is higher than the selected threshold; and c) when the precursor mass does not match the mass of the corresponding database sequence, due to the presence of a modification or a mutation on the peptide or to an error in the database sequence. Another possibility is to use short amino acid sequences –called tags- extracted from the MS/MS spectrum, using *de novo* sequencing approaches. As this filter is mainly used for open-modification searches, it will be described in greater detail in Chapter V. A third filtering method is to perform several analysis steps, and to use proteins identified in previous runs as a restrained database for subsequent runs.

#### IV.3.5.2. Modeling virtual MS/MS spectra from theoretical sequences

PFF methods try to score the similarity of a given theoretical sequence with an experimental spectrum. Since comparison can only be applied to similar entities, PFF methods build theoretical spectra from amino acid sequences. This procedure, referred to as spectrum prediction, consists in predicting the fragmentation of a peptide given its amino acid sequence, its charge state, as well as experimental conditions. The aim is to build a spectrum that approaches the corresponding experimental one. Most of the identification tools use basic rules for spectrum prediction: each cleavage position in the theoretical sequence is translated into several expected peaks according to a list of possible ion types, charges and molecule losses. So far little attention has been given to modeling the intensity of the peaks, or to measuring the influence of neighboring amino acids. Recently though, several authors carried out work aimed at characterizing sequence-dependent fragmentation patterns using statistical observations from identified MS/MS spectra. For example, Kapp et al. (Kapp et al. 2003) measured the extent to which specific amino acid residues promote or dampen the cleavage on their N- or C-terminal side, and they analyzed the influence of basic residues on fragment ion peak intensities. Elias et al. (Elias et al. 2004) built from a set of identified MS/MS spectra a probabilistic decision tree, and used it to estimate the intensity distribution of a fragment ion given an extended list of peptide and fragment attributes. As a last example, Zhang (Zhang 2004) developed a mathematical model based on classical kinetic rules and on the mobile proton hypothesis. There is no doubt that such studies are of great help to improve the performance of spectral interpretation methods by increasing the precision of spectrum prediction. They will also allow making scoring functions more efficient by incorporation of additional knowledge on the fragmentation process.

#### IV.3.5.3. Scoring the similarity between experimental and theoretical spectra

In the present section, we shall discuss how PFF algorithms score the similarity between an experimental and a theoretical spectrum. Ideally, a scoring scheme should provide a low number of both false positive and false negatives as well as a good discrimination between true and random matches. The three basic “peak matching” measures are i) the counting of common masses, ii) the cross-correlation and iii) the dot product.

i) Counting common masses

The simplest way to account for the similarity between two spectra is to simply count, within a given margin error, masses that are shared by both spectra (see Figure IV-4).

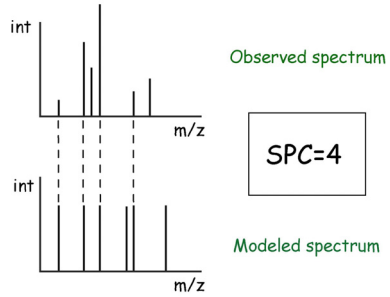



*Figure IV-4: The basic shared-peak-count (SPC) score between an experimental spectrum and a spectrum computed from a theoretical peptide is the count of the matching peaks.*

An example of SPC-based scoring scheme is implemented in the PEP_PROBE method (Sadygov and Yates, III 2003), which assigns to each candidate peptide a score value –log(P) that depends on the probability P that the SPC obtained between the candidate peptide and the spectrum is random, given a distribution model. The authors chose to use an hypergeometric distribution, which models a random sampling (without repetition) in a finite population of objects of two distinct types, given three parameters: N, the size of the population, K the number of items with the desired characteristic in the population, and $N_1$, the number of samples drawn. Figure IV-5 establishes the correlation between the hypergeometric model and the PFF identification context.

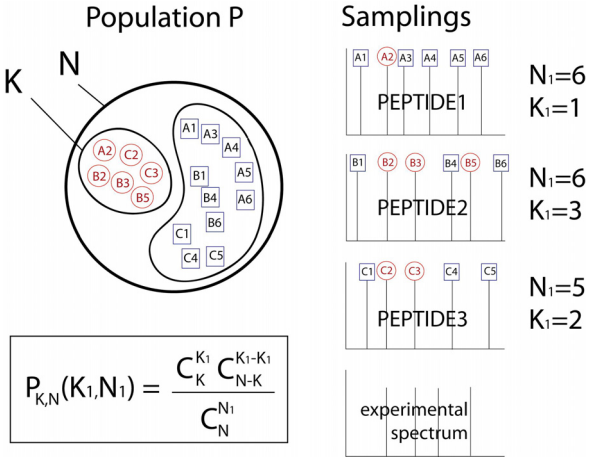



*Figure IV-5: PEP_PROBE score. The population P is the pool of all fragment masses obtained from candidate peptides (filtered according to their molecular weight); P is composed of N fragments, which are divided in two sets: the sets of "matched fragments", composed of all fragments that match peaks in the experimental spectrum, of size K, and the set of "non-matched fragments". Comparing a candidate peptide to the experimental spectrum amounts to sampling $N_1$ elements of P. The hypergeometric distribution $P_{K,N}(K_1,N_1)$ gives the probability to obtain a SPC of $K_1$ matched fragments among the K possible matches, given the sample size $N_1$. In other words, it gives the probability P that the SPC obtained between a candidate peptide and the spectrum is random.*

ii) Cross-correlation

A cross-correlation is a mathematical method to evaluate whether two signals exhibit common features. When the signals are continuous, it can be calculated according to the following equation.

$$C_{xy} = \int_{-\infty}^{+\infty} x(t)y(t+\tau)dt$$

*where*

*x(t) and y(t) are the two signals and*

*τ is a displacement value between the signal.*

It the signals are equal, the correlation function should maximize at τ=0. Sequest (Eng et al. 1994), which was the first published MS/MS identification tool based on a PFF approach, precisely uses this kind of similarity measure. Sequest, as the algorithm was published in 1994, starts by selecting candidate peptides from the database using the precursor mass of the spectrum, without any enzyme cleavage specificity. Fragments of b- and y-ion types are computed for each peptide, and the latter are pre-scored according to a scoring scheme $S_P$ including several different criteria, as the number of matching fragments, the presence of consecutive fragment ion matched and of immonium ions:

$$S_P = \frac{\left(\sum i_m\right) n_i (1+\beta)(1+\rho)}{n_t}$$

*where*

$i_m$ *is the intensities of the matching fragments,*

$n_i$ *is the number of matching fragments*

$\beta$ *is a value initially set to 0.075 that is incremented for each consecutive fragment ion matched,*

$\rho$ *is a value initially set to 0.15 that is incremented for matched immonium ions*

$n_t$ *is the total number of predicted fragments*

The 500 top-scoring candidate peptides undergo the cross-correlation analysis. The experimental spectrum is processed to remove the mass-to-charge ratio of the precursor ion and to normalize the peak intensities. Then, approximate spectra are constructed for each high-scoring candidate peptide and cross-correlations of pairs of spectra are performed. Sequest reports as similarity score the value of the cross-correlation when τ = 0 minus the mean of the cross-correlation when τ varies from –75 to +75.

iii) Spectral angle contrast

A third type of score that accounts for mass correspondence between two spectra is the spectral angle contrast method, which was implemented in Sonar (Field et al. 2002) and GutenTag (Tabb et al. 2003b). In this method, both spectra are represented as vectors in a N-dimensional space, N being the number of matched peaks. The length and direction of the vectors are determined by the *m/z* and

intensities of the peaks. The normalized spectral contrast angle between two spectra O and M is given by:

$$\cos(\theta) = \frac{\vec{O} \cdot \vec{M}}{|O| \cdot |M|} = \frac{\sum_{i=0}^{n} o_i \cdot m_i}{\sqrt{\sum_{i=0}^{n} o_i^2 \cdot \sum_{i=0}^{n} m_i^2}}$$

*where*

$\vec{O}$ *is the vector constructed from spectrum O composed of n matching peaks* $o_i \ldots o_n$,

$\vec{M}$ *is the vector constructed from spectrum M composed of n matching peaks* $m_i$ *to* $m_n$, *and* $|O|$ *and* $|M|$ *are their respective length*

θ ranges from 0 to 90 degrees. Zero degrees means that the two spectra are not discernible. As the angle grows, less and less similarities are found between the spectra, until the angle reaches 90 degrees, indicating a maximal spectra differentiation. Figure IV-6 illustrates the spectral angle computation between two spectra composed, for simplicity, of two matching peaks.

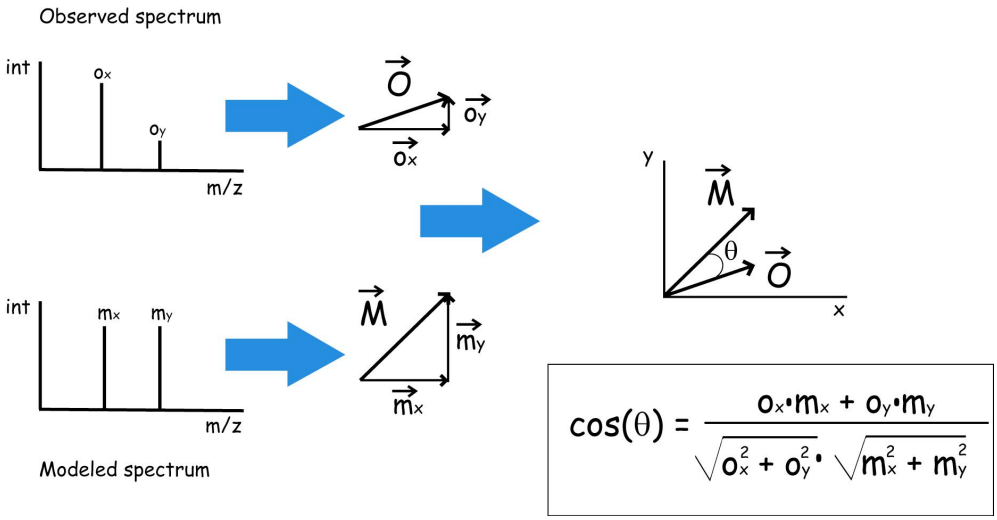


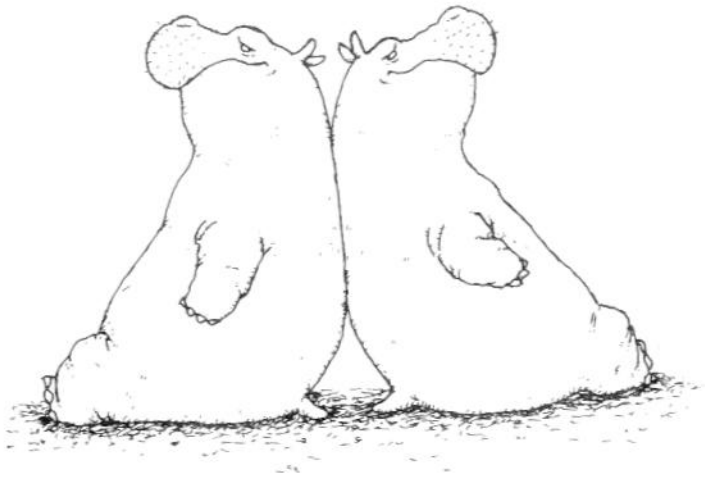

*Figure IV-6: Spectral angle contrast score*

Actually, all these similarity measures are very close to each other, as Fu et al. recently pointed out (Fu et al. 2004). Additional information, like probabilities of ion series or the number of consecutive fragment matches may nevertheless complement the basic "peak matching" measure. Thus, Fu et al. (Fu et al. 2004) extended the spectral angle contrast method to exploit the correlative information among fragments. In addition, most scoring schemes are cast into a probabilistic framework. Thus, Mascot (Perkins et al. 1999) evaluates a probability P that the observed similarity score occurs by chance and reports as final score -10log(P)[3]. PEP_PROBE (Sadygov and Yates, III 2003) follows a similar approach using a hypergeometric distribution. SCOPE (Bafna and Edwards 2001), ProbId

[3] unfortunately, no details on the exact probability model used by Mascot have been published so far

(Zhang et al. 2002) and Phenyx (based on the Olav scoring system) (Colinge et al. 2003b) also use probabilistic models, incorporating more or less the same parameters. Phenyx differs in that it includes physico-chemical properties of fragments into what is called an “extended” match. During the comparison between the spectrum and the theoretical peptides, information is used about matching fragments, mass errors, ion series, peptide amino acid composition, presence of modifications, number of missed cleavages and so on. In this way, the information present in the experimental spectrum is more extensively exploited, resulting into a more precise probabilistic model and thus reducing the number of false positive identifications.

一頭のカバともう一頭のカバの区別がつかない奴がいる。
そういうのを観念的というのだ。



Il y a des gars qui ne peuvent pas distinguer
un hippopotame d'un autre hippopotame.
On les appelle "conceptualistes".

There are guys who cannot distinguish between
a hippopotamus and a hippopotamus
They are called "conceptualists".

Figure adapted from "Monsieur Hippopotame" by Tanikawa Shuntarô

# CHAPTER V

# OPEN-MODIFICATION SEARCH METHODS

This chapter describes the concept of "open-modification search" approach. Methods based on such an approach are specifically designed to identify and characterize post-translationally modified or mutated peptides. More generally, "open-modification search" method can take into account any event that modifies the sequence or residue mass of a peptide analyzed by mass spectrometry. We show how a post-translational modification may change the peak pattern in an MS/MS spectrum and describe different strategies to account for such events during the comparison of the spectrum with candidate peptides.

# V. “Open-modification search” methods

## V.1. Introduction

Many events may occur that result into differences between an MS/MS spectrum and its corresponding theoretical sequence found in databases. Section IV.1 introduced well-known examples of such events: PTMs, mutations, database errors, polymorphisms, transpeptidation, missed or exotic-cleavage and alternative splicing. An “open-modification search” is an MS/MS identification procedure that takes into account any type and number of differences between an MS/MS spectrum and theoretical peptides from a database. “Open-modification search” algorithms are more specifically designed to identify and characterize post-translationally modified peptides.
The chapter is organized as follows: Section V.2 illustrates the effect of a residue modification in an MS/MS spectrum; the next topic is the combinatorial issue inherent to PFF methods that simulate modifications on the database sequences; and the remaining part of the chapter describes “open-modification search” MS/MS identification methods.

## V.2. MS/MS spectra obtained from peptides with residue modifications

Many reported PTMs are characterized by the attachment of a chemical group on an amino acid residue. Residue modification may also be introduced deliberately (e.g. cysteine carbamidomethylation) or may happen as an artifact during sample preparation (e.g. oxidation of methionine). Most PTMs are site-specific and/or position specific. For example, oxidation is observed on M, H and W; phosphorylation, a much studied PTM, occurs on any S, T and Y, and methylation may occur at any N-terminal site and on internal C, H, K, N, Q, R (reference: UniMod).

Residue modifications change the amino acid molecular mass, and consequently, the peptide mass. According to the modification, the mass difference varies from a few to several hundred Daltons. Thus, for example, deamidation of N and Q is characterized by a mass shift of 0.984 Daltons (loss of an hydrogen and an nitrogen, and gain of an oxygen) while glycine and lysine myristoylation is characterized by a mass shift of 210.198 (gain of 26 hydrogens, 14 carbons and 1 oxygen). The difficulty when analyzing MS/MS spectra produced from modified peptides is that the fragmentation pattern is affected by the presence of the modification. The modification may, for example, hamper the fragmentation mechanism because of biochemical or physical properties (e.g. steric hindrance). But it also modifies the position of the peaks (in respect with their expected position computed from a theoretical non-modified sequence). Figure V-1 illustrates how a modification may affect a spectrum.

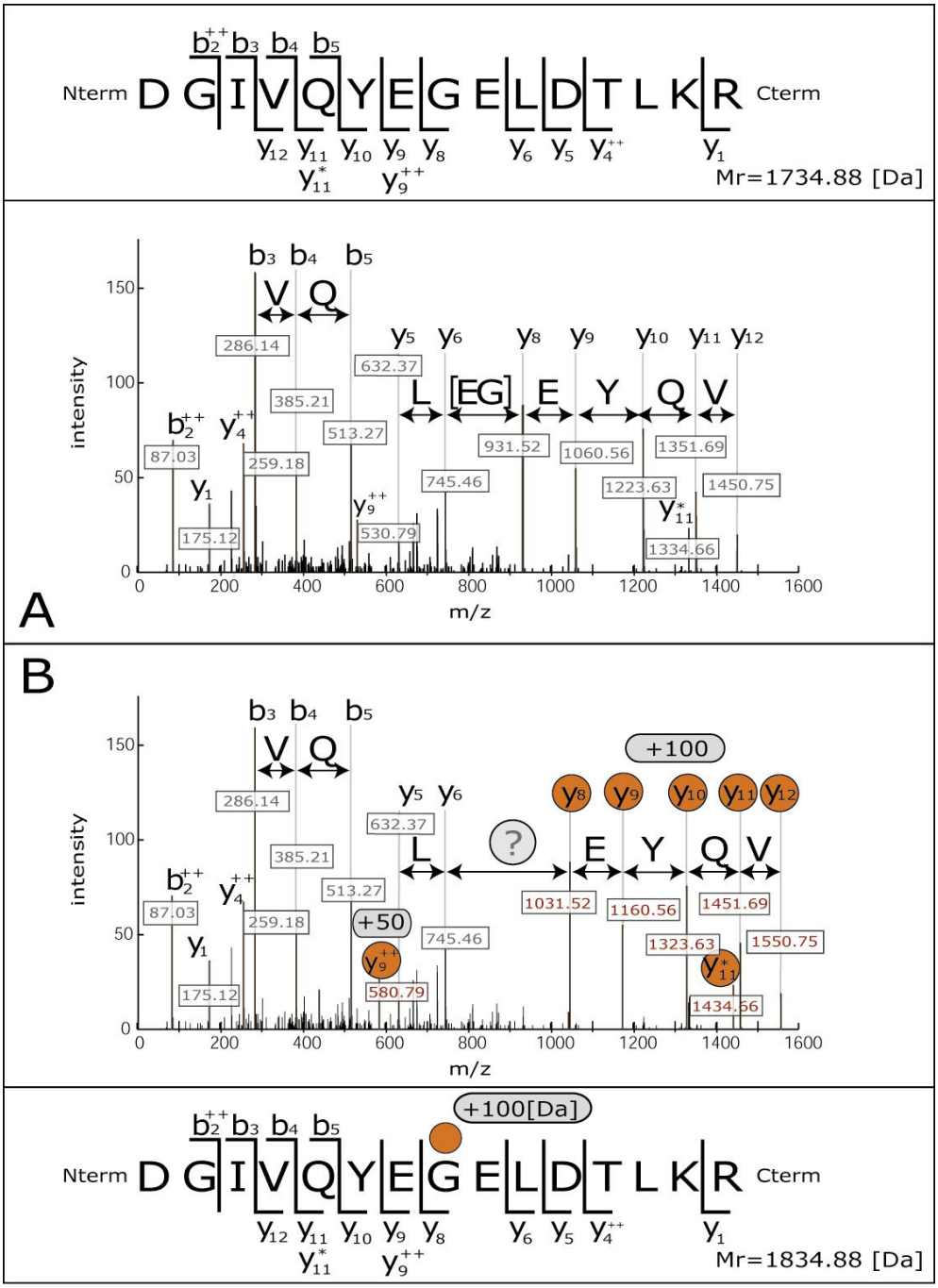



*Figure V-1: Effect of a modification on an MS/MS spectrum*

*Box A shows an annotated spectrum of peptide DGIVQYEGELDTLKR. Major ion assignments are indicated, as well as the corresponding peak masses and sequence information that can be inferred from peak differences. Box B shows the same spectrum, in which we simulated a 100 Dalton modification on the second glycin of peptide DGIVQYEGELDTLKR. Peaks marked with a circle represent fragments that carry the modification and are thus shifted by a delta value from their expected position. It should be noted that the shifted peaks are not necessarily grouped in the spectrum and that the delta mass is not constant, since this depends on the number of charges in the fragment. Moreover, according to the type of modification (gain or loss of atoms), peaks may be shifted either to the left or to the right. In box B, sequence information is partly lost, as the amino acids GE can no more be read from the mass difference between ions y6 and y8.*

## V.3. PFF approach and combinatorial issue

PFF algorithms score theoretical peptides by modeling virtual spectra and comparing them with the experimental one. If a modified spectrum is compared with its corresponding non-modified theoretical peptide, a number of peak matches are lost (on the average 50% of the masses are shifted), reducing the confidence in the identification score. In addition, since typical PFF approaches filter the database using the precursor mass, the correct theoretical peptide may merely not be selected as candidate for the identification. One possibility to handle modifications using a PFF algorithm is to specify a list of anticipated modifications (and associated masses). The identification algorithm generates virtual spectra of all PTM variants of the database sequences using the supplied list. This comes down enumerating the modifications for all candidate peptides and, for each variant, replacing the standard amino acid mass by the corresponding modified amino acid mass before computing the virtual spectrum. Certain tools, such as Phenyx (Colinge et al. 2003b) and InsPect (Tanner et al. 2005), allow the user to define his/her own types of modifications and/or to make use of additional information from databases (annotated PTMs). Generally, the user can also input information about the frequency of occurrence of a given modification. Thus, modifications that occur with high frequency (e.g. deliberately introduced during sample preparation) are handled in so-called "fix" mode. In such a case, all possible sites for a given modification are modified. When the frequency of a modification is low, the latter is handled in "variable" mode. In this mode, the program must compute all possible occurrence combinations according to the number of modification sites on the theoretical peptide.
Variable modifications result into a huge increase of the number of peptides obtained after digestion and modification simulation, and then in a dramatic increase in computing time. More importantly, the number of random matches increases also because more candidate peptides are generated. For these reasons, PFF algorithms typically limit the number of different variable modifications to a small number (less than 10). In addition, mutations in the peptide and errors in the database cannot be taken into account because of combinatorial explosion.

The present section illustrates the combinatorial complexity faced by MS/MS identification approaches that want to account for modifications.
To evaluate the complexity arising from the introduction of PTMs into the database, we need to compute the probability P of having k possible sites of modification in a sequence of length L. This probability is given by a binomial distribution B(L, p, k), where p is the occurrence probability for a considered modification. Such distributions are observed in situations of the general "k successes out of L trials" type. For the sake of simplicity, we assume the modification sites to be independent. According to the binomial distribution, the number $N_k$ of peptides containing k sites for a possible modification with occurrence probability p that are expected to arise in an n sized sequence database is given by:

$$N_k = n \cdot \mathcal{B}(L,p,k) = n \cdot C_L^k \cdot p^k \cdot (1-p)^{L-k}$$

Figure V-2 shows distributions of $N_k$ obtained for different modification occurrence probabilities.

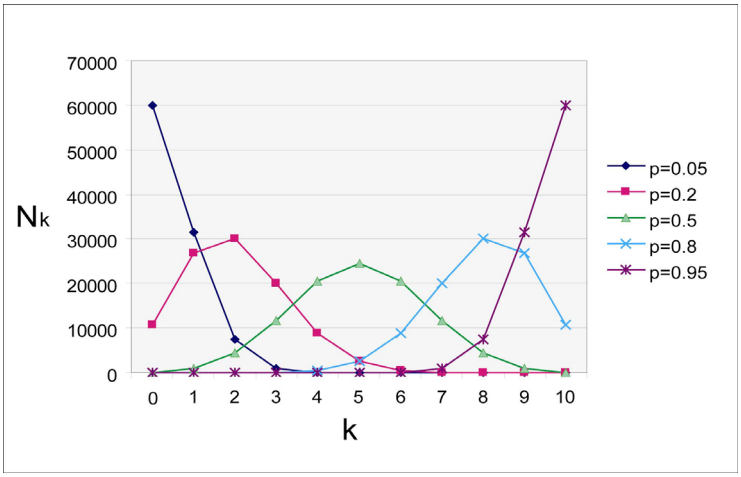



*Figure V-2: Expected number of peptides with k modification sites*
*Simulation of the expected number $N_k$ of peptides with k modification sites for different modification occurrence probabilities in a 100'000 sequence database (sequences are assumed to be of length 10, with independent amino acids). In the case of a modification that occurs on very few amino acids (e.g. p=0.05), one expects that most peptides will contain no modification site (k = 0), while in the case of a modification that occurs on many possible amino acids (e.g. p=0.95), one expects that most peptides will contain many modification sites (k near 10).*

When a peptide contains more modification sites than modification events, the PFF algorithm will have to compute the different possible permutations to find all variants. Let $S_M$ be the number of possible permutations for M modification events.

$$S_M = \sum_{k=M}^{L} C_k^M \cdot N_k$$

*where*

*k is the number of possible modification sites*
*$N_k$ is the expected number of peptides with k modifications sites*
*M is the number of modification events*

An example of complexity evaluation is given below.
Let us assume a user runs a PFF algorithm. The search is restrained to Human sequences in the Swiss-Prot database. This corresponds to over 10'000 proteins, that produce, according to tryptic cleavage rules, about 800'000 peptides of size 3 to 30. A mean length of 11 amino acids is considered for the simulation (L=11).
The user specifies only one modification type for the search: hydroxylation, which may occur as PTM on amino acids D, K, N and P, and as artifact on amino acids F,Y, without restriction on the sequence position. We roughly estimate the occurrence probability of hydroxylation to 6/20. In addition, the user sets the modification as variable (so that a maximum of 2 modification events on the peptides is taken into account).

Let M represent the number of modifications. When k is greater than M, the modification sites have to be permuted to produce all possible variants. Thus for example, a peptide with 5 modification sites (k=5) allows for 5 permutation possibilities in case of 1 modification event (M=1), and for 10 possibilities in case of 2 modification events (M=2). The total number of peptides obtained after simulating modifications is given by the total sum of Table V-1. In this simulation, it is equal to 7.4 millions.

| | k=0 | k=1 | k=2 | k=3 | k=4 | k=5 | k=6 | k=7 | k=8 | k=9 | k=10 | k=11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $N_k$ | 15818 | 74573 | 159800 | 205457 | 176106 | 105663 | 45284 | 13862 | 2970 | 424 | 36 | 1 |
| $S_{M=1}$ | / | $1*N_1$ | $2*N_2$ | $3*N_3$ | $4*N_4$ | $5*N_5$ | $6*N_6$ | $7*N_7$ | $8*N_8$ | $9*N_9$ | $10*N_{10}$ | $11*N_{11}$ |
| $S_{M=2}$ | / | / | $1*N_2$ | $3*N_3$ | $6*N_4$ | $10*N_5$ | $15*N_6$ | $21*N_7$ | $28*N_8$ | $36*N_9$ | $45*N_{10}$ | $55*N_{11}$ |

*Table V-1: Expected number of peptides after modification simulation*

As the PFF method will be able to filter the 7.4 millions theoretical peptides using the precursor mass, the final number of candidates will stay more or less reasonable as long as the number of modification types taken into account remains low. If the user wishes to include more modification types, the number of candidate peptides must be restrained using more stringent filtering techniques. Much effort has been made in this direction over the last years. Notably, several programs, including Phenyx (Colinge et al. 2003b), Mascot (Creasy and Cottrell 2002), TANDEM (Craig and Beavis 2004) and VEMS (Matthiesen et al. 2004) now split the identification procedure into two runs and use the first run as a filter for the second one. Typically, the first run is performed with low combinatorial parameters (regular cleavages, a few number of fixed modifications). Relatively high-scoring peptide matches allow isolation of proteins that are likely to be represented in the experimental mixture, and which are grouped in a database of limited size. The second run is then performed with loose parameters on the limited database. During the second run, more modifications can be considered, as well as possible mutations and non-specific cleavages. This approach is based on the assumption that all proteins represented in the experimental mixture contain at least one tryptic unmodified peptide and have been fished out during the first run. Another possibility to restrain the number of candidate peptides is to combine a *de novo* approach with a PFF approach. Thus, the algorithm InsPecT (Tanner et al. 2005) focuses on a very efficient filter, by selecting candidate peptides that match stretches of two or three amino acids extracted *de novo* from the spectrum. In addition, InsPecT applies dynamic programming to find out which candidate peptides can match the precursor mass, given the set of allowed modifications. By this mean, explicit enumeration of all peptide variants is avoided. With such enhanced filtering procedure, InsPecT can support more modification types during the search.

But the procedure that consists in building the theoretical modified peptides from a set of possible modifications is simply not applicable in an “open-modification search” strategy. This would amount in creating, for each theoretical peptide, a huge number of modified peptides, in which every amino acid type, at every position, could be modified by any delta value. This clearly illustrates the tricky problem of the search space size for “open-modification searches”. Since the development of the first MS/MS identification approaches, very few methods aimed at identifying peptides without the use of a list containing expected modifications have been developed. However the subject is appealing. First, because it represents a challenge, asking for new comparison strategies. Second, because it may help reduce the number of non-identified spectra presenting good peak statistics. In most experiments,

the proportion of unidentified spectra is very high and could even reach 90% (Meyer, oral communication). Third because it meets the need of tools aimed at characterizing PTMs to better understand cellular pathways and investigate possible relations between the presence or absence of PTMs and disease states.

The next sections detail the state-of-the-art of "open-modification search" algorithms. We present in this chapter all methods that have been claimed to deal with any type of residue modifications during the identification procedure. We do not include methods aimed at cross-species identification since they only deal with mutation and polymorphism.

## V.4. State of the art of "open-modification search" approaches

### V.4.1. A pioneer work: the "sequence tag" approach

As explained in Section III.3.2.2, a key property of MS/MS spectra is that they contain series of peaks resulting from successive fragmentation positions in the peptide sequence (see Figure V-3). This principle is exploited by *de novo* sequencing methods, which typically infer whole peptide sequences by running a reconstructed spectrum (in which every peak is associated to an ionic hypothesis). Mann and Wilm proposed, in 1994, to use *de novo* extracted sequence information as a substitute filter that would allow selecting candidate peptides independently of the presence of modified residues. Their idea was to exploit regions in the spectrum containing high-quality peak signal to infer partial sequence information rather than complete peptide sequences. They called these inferred partial sequences "sequence tags" and defined them as short stretches of amino acid sequences flanked by two "docking" masses representing the start mass (or prefix region mass) and the end mass (or suffix region mass) of the tag (see Figure V-3).

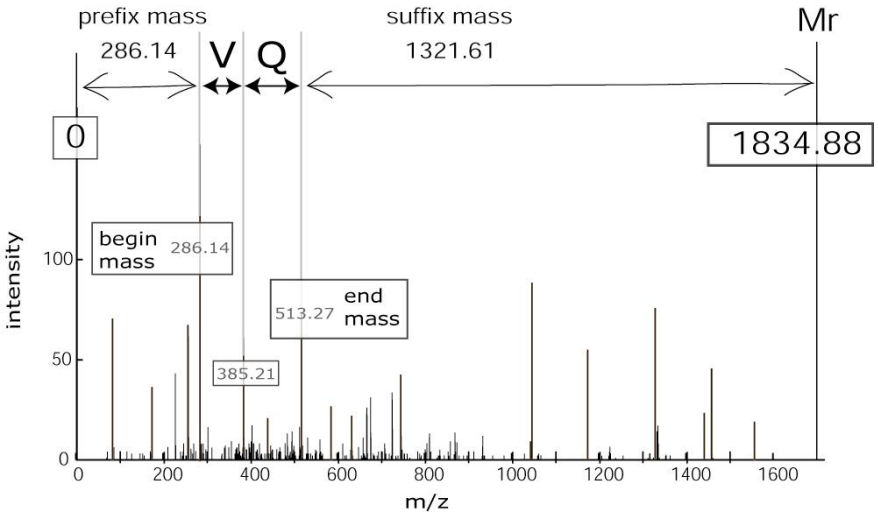



*Figure V-3: A "sequence tag" inferred from an MS/MS spectrum*
*The tag is composed of two flanking masses –called prefix and suffix masses- and of a sequence. Masses of the prefix and suffix regions are easily computed from the first and last masses composing the tag (denoted as begin mass and end mass) and from the precursor mass (Mr).*

The idea of Mann and Wilm was to use tags generated by *de novo* spectrum interpretation to scan the database using a pattern matching-type algorithm and the flanking masses instead of a precursor mass-based filter. Then, candidate sequences that matched one of the tags were subsequently scored using an SPC-type procedure. Their method dealt with modifications –even unexpected ones- by allowing one of the regions to mismatch during the candidate extraction procedure (see Figure V-4).

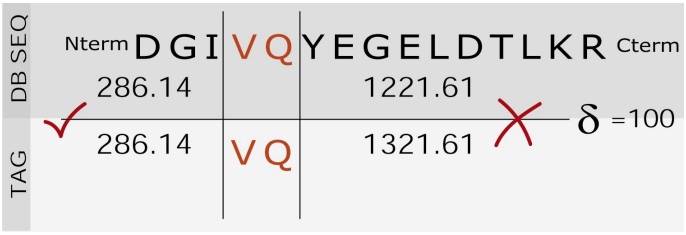



*Figure V-4: Candidate peptide selection using a tag filter*
*The tag extracted in Figure V-3 is composed of three attributes: a) the prefix mass 286.14, b) the sequence VQ and c) the suffix mass 1321.61. A minimum of two attributes must match for a theoretical sequence to be considered as a candidate (in this case, the sequence and the prefix mass). The delta value observed between the theoretical and observed suffix masses can be explained by the presence of a modification in the suffix region of the tag.*

If both flanking masses match, the experimental peptide is supposed to be unmodified and the complete theoretical sequence can be evaluated against the spectrum using a PFF approach. If only one flanking mass matches, the comparison focuses on the matching part of the sequences, without attempting to include additional information by taking into account shifts between peaks of the experimental and theoretical spectra. Consequently, in case of modifications or mutations, as it leaves out one part of the candidate peptide in the scoring phase, Mann and Wilm's algorithm does not maximally capture the similarity between the experimental spectrum and the corresponding theoretical one. Luckily, with a careful manual tag extraction, the number of candidate peptides to evaluate is drastically reduced, and the scoring scheme does not have to be very robust to more or less efficiently distinguish the correct peptide from all the other ones. Nevertheless, the manual tag extraction makes Mann and Wilm's method unsuitable for automated MS/MS identification.

### V.4.2. GutenTag: an enhanced version of the "sequence tag" approach

In 2003, Tabb et al. (Tabb et al. 2003b) implemented a similar approach (schematized in Figure V-5), named GutenTag, with an automatic tag extraction procedure and an enhanced scoring scheme. In this method, the spectrum is represented as a spectrum graph, which is recursively parsed to extract the tags. The size of the spectrum graph is controlled by carefully preprocessing the peaks, by considering all peaks as y-ion types, and by limiting the "sequence tag" lengths to a given size-range. Each extracted tag is then scored using a combination of two subscores: the *m/z* score, that takes into account mass errors between peak masses included in the tags and their expected position given the tag sequence, and an intensity score. Then, the best-scoring tags are searched in parallel against the sequence database using a "trie" structure (Aho and Corasick 1975), which allows locating all

occurrences of any of a finite number of keywords in a character string. The masses of the flanking sequences are determined for each matching peptide, and every sequence with at least one matching flanking mass is retained as candidate peptide and scored against the spectrum using the spectral angle contrast scoring scheme (see Section IV.3.5.3).

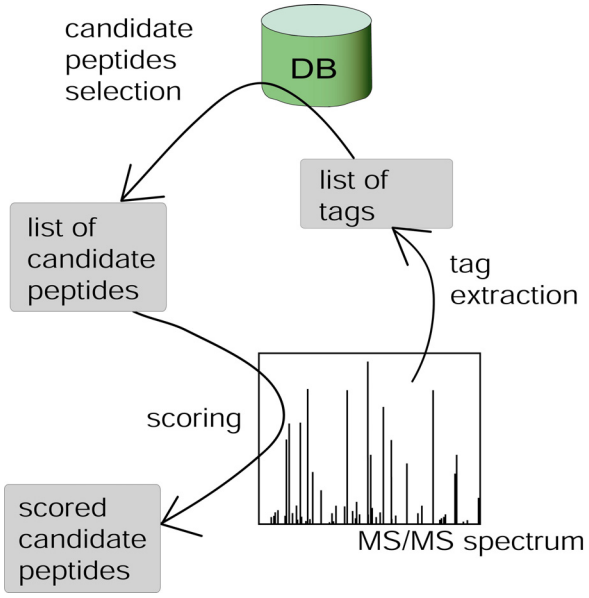



*Figure V-5: GutenTag's approach*

### V.4.3. PEDANTA: spectral alignment

The "spectral alignment" method, from Pevzner et al. (Pevzner et al. 2000) can be considered, together with the Mann and Wilm's "sequence tag" approach, as a pillar for "open-modification search" approaches. The method, called PEDANTA, extends the PFF matching concept by taking into account the possible existence of shifts (or gaps) that would allow a better peak matching.

The algorithm consists in aligning the masses of the theoretical and experimental spectra by storing all possible matches (without considering mass similarities) in a matrix, and then searching for the path that best explains the similarity between both spectra using dynamic programming. Figure V-7 shows the alignment procedure for two hypothetical spectra. In short, the procedure splits one of the two spectra so as to compare one spectrum with sub-regions of the other one (as illustrated in Figure V-6). There is a strong analogy to the global sequence alignment method of the Needleman and Wunsch algorithm (Needleman and Wunsch 1970).

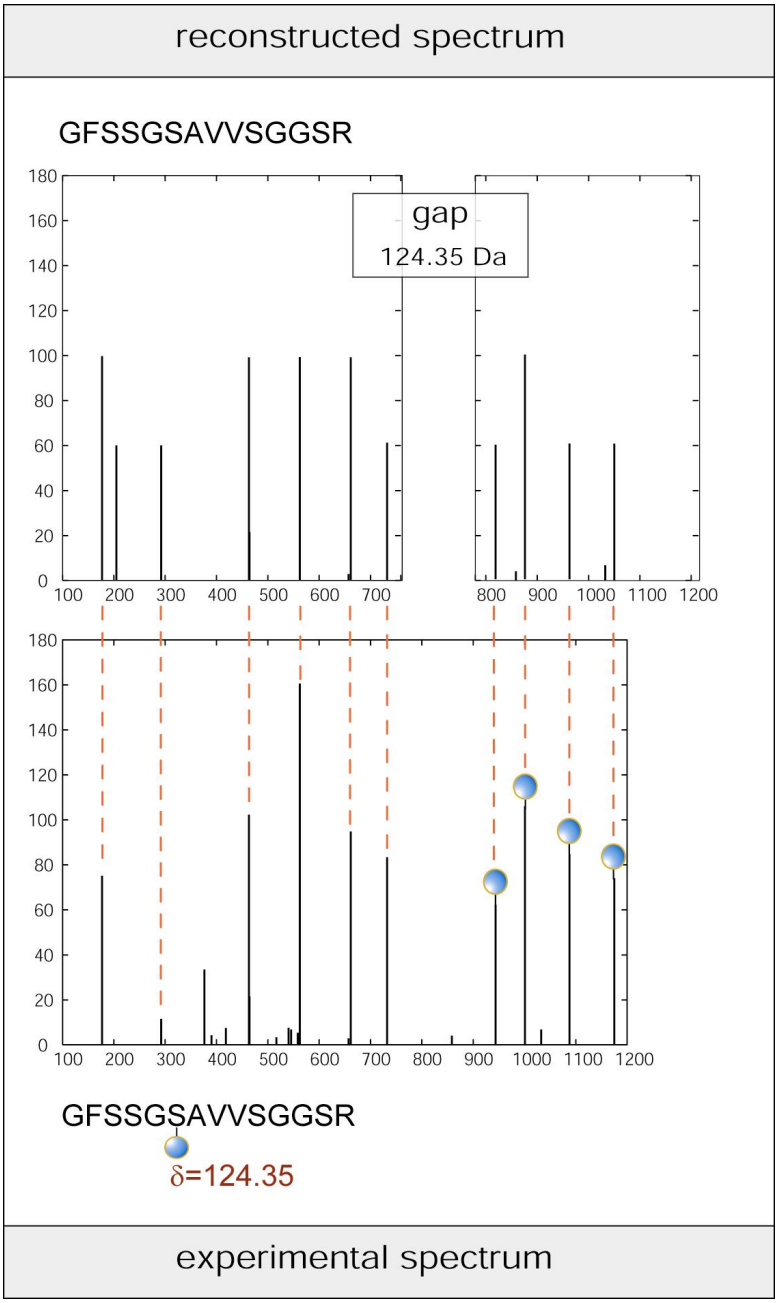
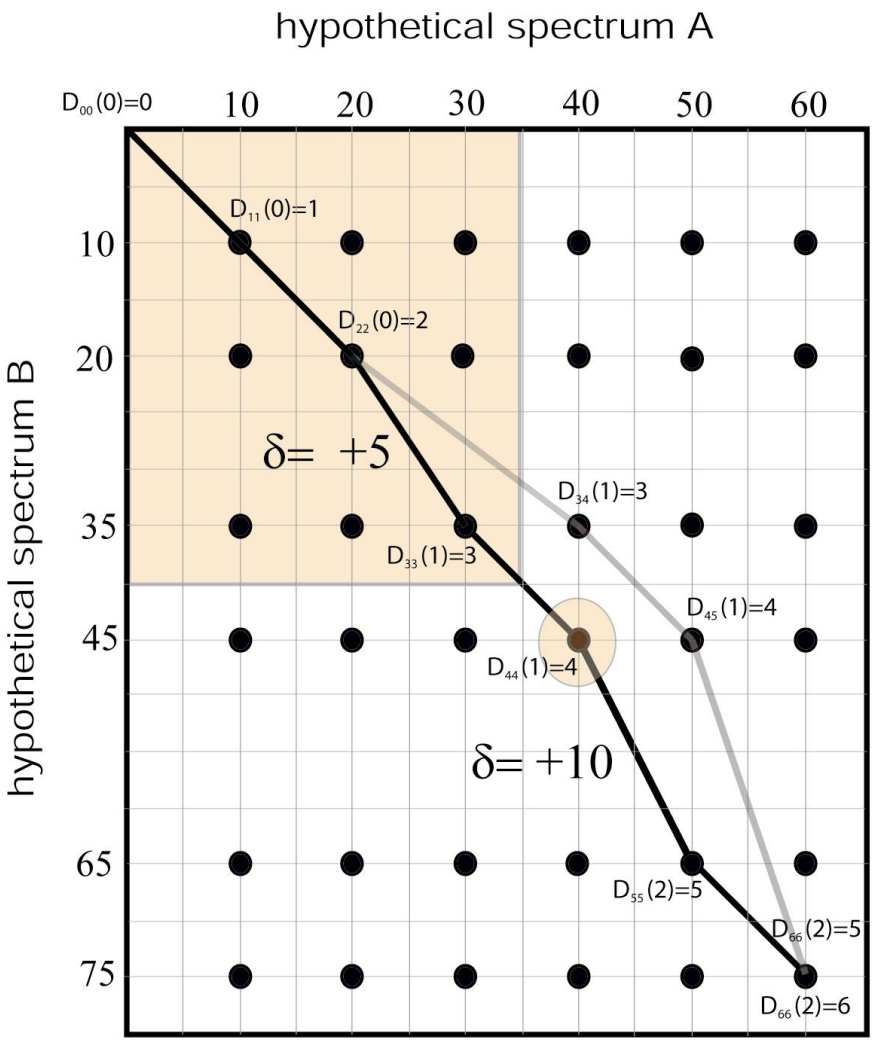



*Figure V-6: Principle of spectral alignment*

*Spectra alignment is an extension of the PFF matching concept by allowing groups of peaks to be shifted to increase the number of matched peaks.*




$$D_{ij}(k) = \max_{(i',j')<(i,j)} \begin{cases} D_{i'j'}(k)+1 & \text{if (i',j') and (i,j) are co-diagonal} \\ D_{i'j'}(k-1)+1 & \text{otherwise} \end{cases}$$

*Figure V-7: Pedanta spectral alignment procedure*
*Alignment of two hypothetical spectra A = {10, 20, 30, 40, 50, 60} and B = {10, 20, 35, 45, 65, 75}. All possible matches, i.e. pairs ($a_i$, $b_j$), are represented in a matrix $M_{ij}$ by a dot. The number of dots that are covered by the diagonal starting from the origin of the two axes is the SPC score when no shift is allowed. $D_{ij}(k)$ is defined as the k-similarity between $A_i$ and $B_j$ (the spectra are similar under the assumption that they are k modifications apart) and corresponds to the maximum number of dots on a path to ($a_i$, $b_j$) taking k+1 distinct diagonals. The dynamic programming recurrence for computing $D_{ij}(k)$ is indicated below the matrix. Two paths are shown. The optimal one, with a score D(k=2)=6; and a sub-optimal path, shown in gray, with a score D(k=2)=5. A basic SPC score would lead to a D(k=0)= 2.*

Pevzner's approach is simple and original. Shifts observed in a path allow making hypotheses about the modification types. Of course, the scoring is too basic and a tag-based filter would be appreciable

to speed up the search, but the authors of Pedanta were undoubtedly more interested, at the time of publication, in describing a new original approach than in producing an "all-in-one" identification tool.

### V.4.4. OpenSea: tag extension

OpenSea, which was introduced in Section IV.3.4.3 is dedicated to the alignment of *de novo* sequences against database sequences. In a recent publication, it has been adapted to account for known and unknown modifications. Here is given a description of the algorithm, as presented in (Searle et al. 2004;Searle et al. 2005).

OpenSea takes as input *de novo* sequences obtained by the PEAKS sequencing software. It first identifies a high-scoring unambiguous tag of two or three amino acids in the *de novo* sequences. Then it selects candidate peptides by matching the tag against database sequences using a string search procedure. The alignment between each candidate peptide and the *de novo* sequence is then extended at both extremities of the matched tag using a breadth-first procedure (see Figure V-8).

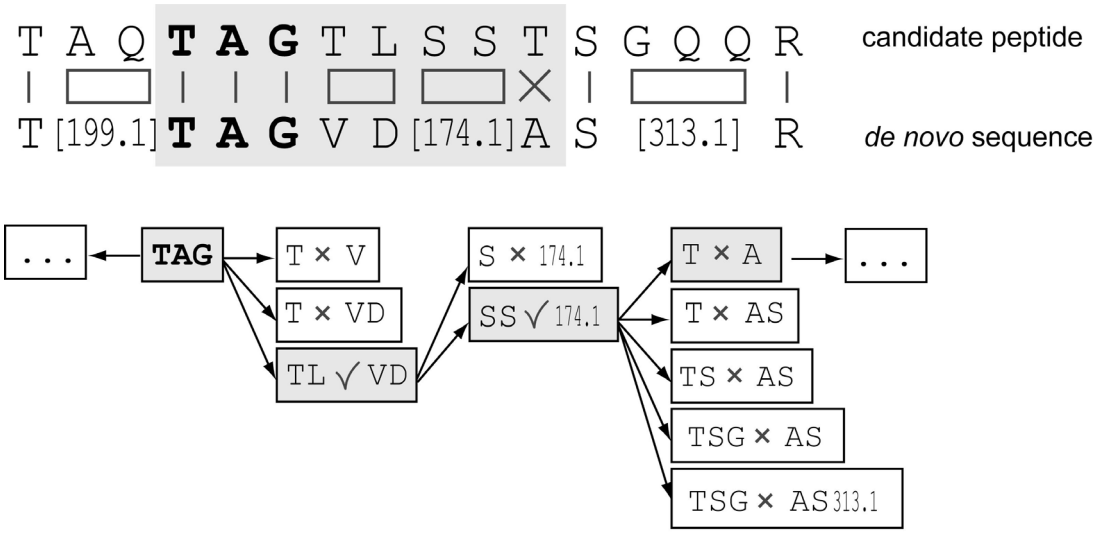



*Figure V-8: Mass-based alignment*
*Alignment procedure of a candidate peptide and the "de novo" sequence using a breadth-first search approach with a depth of three amino acids.*
*Numerical values in the "de novo" sequence correspond to masses of amino acid combinations (e.g. 199.1 correspond to the mass of an alanine plus the mass of a glutamine). Matches between single residues are represented by strokes, matches between groups of amino acids are signified by rectangular boxes and mismatches are represented by crosses. The alignment proceeds from the tag extremities (the tag is represented in bold). The path taken by the alignment follows the shadowed boxes. If no mass match is found by searching the first breadth level, the algorithm searches through the next level, until it reaches a depth of three amino acids. If no match can yet be found, a substitution is assumed, and a new alignment is initiated at the next amino acid in each sequence. (Figure adapted from (Searle et al. 2004)).*

Once the alignment is built and in case of mismatched amino acids, OpenSea runs an interpretation routine to explain the observed mass shifts either by a substitution event, or a modification. If no substitution or known modification can explain the observed shifts, OpenSea assumes it is the result of an unknown modification. Finally, the alignment is re-scored using a PFF type scoring scheme as if a new amino acid were to be identified (Searle et al. 2005).

Figure V-9 illustrates how OpenSea can identify and map a methylation on a peptide.

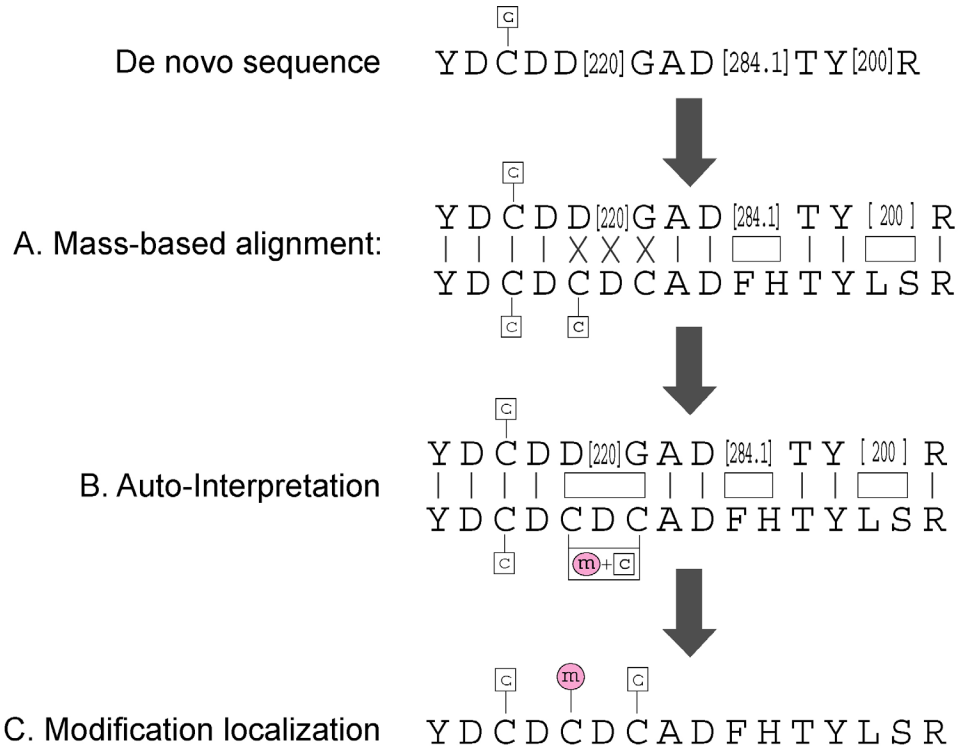



*Figure V-9: OpenSea at work*
*Identification and characterization of a methylated peptide by OpenSea.*
*A. The "de novo" sequence derived from the experimental spectrum is aligned with a candidate peptide (here, the human γS Crystallin chain). Carbamidomethylation of cysteine (CysCAM) was introduced as an expected modification. Three consecutive mismatches are observed (crosses).*
*B. The auto-interpretation process explains the mismatches of D+220+G and CysCAM+D+C by an unanticipated cysteine methylation on either the first C or the second C.*
*C. As a result of the re-scoring procedure, methylation of the first cysteine is reported as the best interpretation (Figure adapted from (Searle et al. 2005)).*

Searle et al. approach is of interest and deals with the identification and characterization of peptides carrying unexpected modifications. However, it suffers from an awkward drawback: it uses tags extracted from "complete" de novo sequences as a start for the alignment rather than tags extracted locally. We reported in Section IV.3.4.2 the difficulty met by *de novo* methods for sequencing peptides carrying unexpected modifications. *De novo* sequencing methods, whether they use the "pseudo" PFF approach or the peak succession approach, have to introduce modified residues in their

amino acid pool to specifically account for modifications. Non-expected modifications of residues in the spectrum cause the correct path of a spectrum graph to be split into two (or more) sections, which are (or are not) connected by alternative paths (see
Figure V-10).

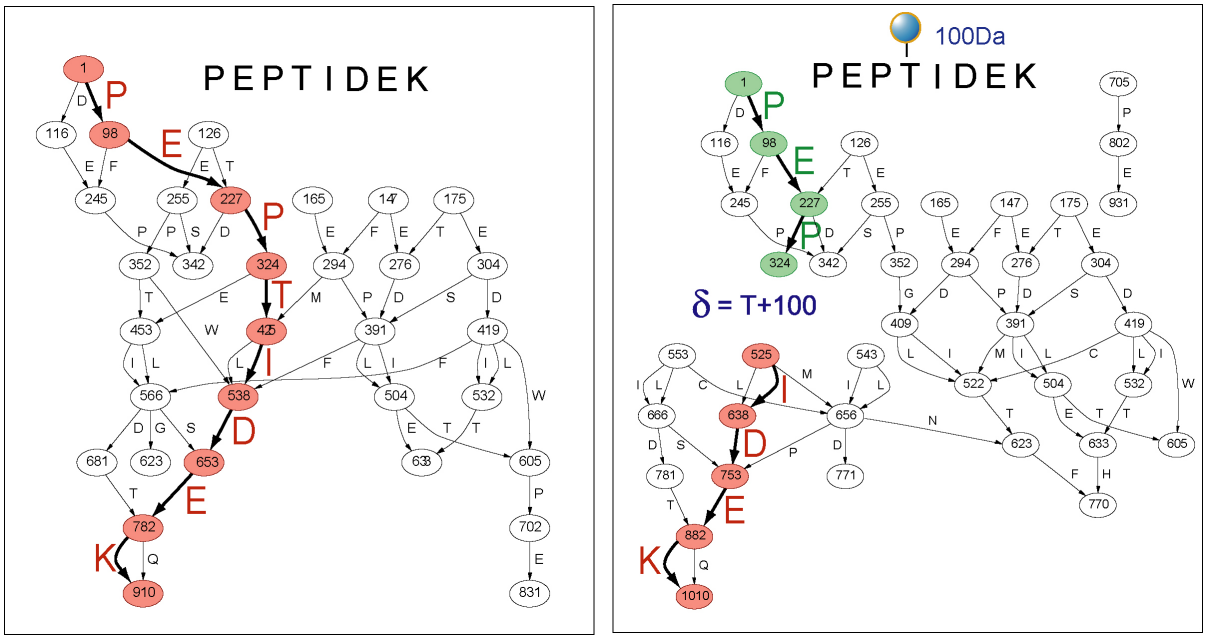
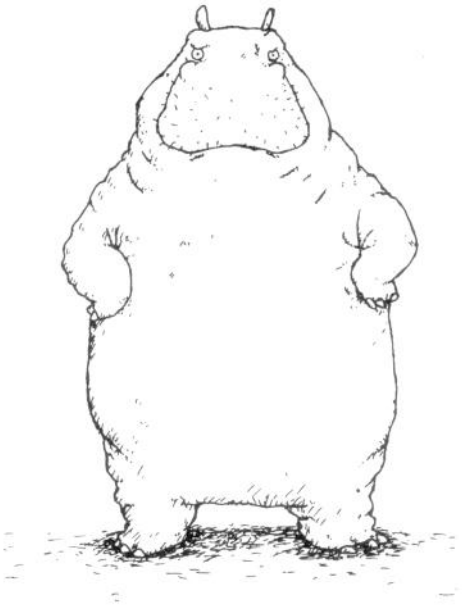



*Figure V-10: Non-expected modification and split paths*
*The figure shows how a non-expected modification causes the correct path in a spectrum graph to be split. The two original spectra were artificially created. Both spectra had complete coverage (b-ion series from position 1 to 6 and y-ion series from positions 1 to 5, leading to 11 peaks in the spectrum). The spectrum was built by Popitam. In the right graph, the path is split in two parts. There is no possibility of joining them by using existing edges (see also Figure VI-8).*

As a result of the splitting, the obtained *de novo* sequences are either truncated (these are not reported by the *de novo* algorithm because they do not match the precursor mass) or contain incorrect amino acid regions that come from an alternative path. Consequently, a large majority of the *de novo* sequences used by OpenSea contain such regions -the *de novo* sequence of Figure V-9 contains precisely three consecutive mismatches- and a significant number of modified spectra are probably not processed by the algorithm because they did not lead to any valuable *de novo* sequence.
A possible solution to this problem relies on a local tag extraction procedure. Recently, Frank et al. (Frank et al. 2005) studied tag extraction for database sequence filtering. In particular, they worked on the covering property, which states that at least one of the tags in the list must be correct (which is necessary for the correct peptide to be presented as candidate for identification) while the list should be as small as possible (which is necessary for efficient filtration). Frank et al. also highlighted the difference between global and local tags. They noted that local paths might not be extensible into an optimal global path, or into any global path at all (dead-end paths). Correspondingly, the tag PEP in the left graph of Figure V-10 represents a global path, while the same tag in the right graph of Figure V10 represents a local path.
Therefore, a more suitable approach for ”open-modification search” software like OpenSea would use local tag extraction, and then perform the mass alignment by looking directly in the spectrum. This is

the approach used by InsPecT, although the latter is not designed for an “open-modification search”, since it requires a list of anticipated modifications and does not, at the moment, include unknown modifications during the alignment procedure.



On m'appelle Popitam, parce que "Popitam l'hippopotame"
Vous trouvez ça drôle?

I am called Popitam, because "Popitam the hippopotamus".
Do you find it funny?

Figure adapted from "Monsieur Hippopotame" by Tanikawa Shuntarô

# CHAPTER VI

# Popitam's ALGORITHM

This chapter describes in details each step of Popitam's algorithm.

# VI. Popitam's algorithm

## VI.1. Introduction

Popitam, whose an earlier version was published in 2003 (Hernandez et al. 2003) uses a tag-oriented approach to perform a database-guided spectrum interpretation. It has been specifically designed to identify spectra from modified and/or mutated peptides without any a priori knowledge about the expected type of modifications. Popitam borrows the spectrum graph from *de novo* methods, but it differs from *de novo* sequencing algorithms in that the graph is specifically parsed for each candidate peptide extracted from a database. Popitam's algorithm searches the graph for all tags of the longest possible length that match subsequences of the current candidate peptide. Then, the tags are combined according to compatibility rules, in order to build plausible spectrum interpretation scenarios. Typically, a scenario is composed of one or several tags, separated by gaps. Using the flanking masses of the tags, Popitam evaluates if a gap contains a modification or a mutation (in which case it is denoted as *modGap*), or if it arises from a lack of information in the spectrum due to low peak statistics (in which case, it is denoted as *lackGap*). Finally, each scenario is scored. The candidate peptide with the highest-scoring scenario is proposed as the identification. An overview of the approach is presented in Figure VI-1. Popitam can be run in command line with a text-based output. It also allows for submission via a web interface with a parser that displays the results in html format. An example of Popitam's output is shown in Appendix A4. A user-friendlier implementation is under development.

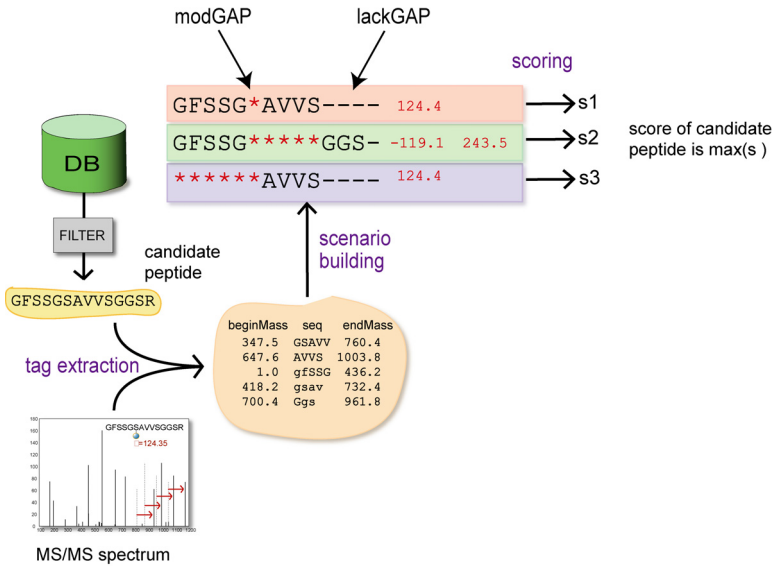



*Figure VI-1: Popitam's approach*
*Candidate peptides are selected from a database according to filtering criteria (possible criteria are taxonomy, a list of accession codes, sequence information, digestion rules, and maximum shift between the experimental precursor mass and the candidate peptide mass). For each candidate peptide, Popitam extracts all tags that are consistent with the candidate sequence and the peak pattern of the spectrum. Then it arranges the various tags according to logical rules and proposes scenarios that include modification hypotheses. Each scenario is scored. The peptide that obtains the scenario with the highest score is proposed as the identification result.*

This chapter describes in detail Popitam's algorithm. First, it introduces some terminology bases; then it describes peak preprocessing steps applied by Popitam. Section VI.5 deals with the building of the spectrum graph, followed by tag extraction, arrangement and scoring. In order to clearly illustrate each step of the algorithm, Popitam was run with an example-spectrum shown in Figure VI-2. This spectrum was chosen because it included two modified amino acids (a carbamidomethylated cysteine and an oxidated methionine). Of course, no information about the modification types had been given to Popitam. During the run, Popitam's parameters were set in such a manner that the collected data were easily exploitable for demonstration purposes: the number of nodes and edges in the graph were deliberately kept low; an AC-based filter was applied, so that only peptides that belong to the correct protein (Swiss-Prot, P36578) were presented as candidates. With a maximum of 1 missed-cleavage authorized, this corresponded to a total of 85 peptides, including the correct peptide YAICSALAASALPALVMSK. Moreover, Popitam was run in mode MODGAPNB = 2. This means that two occurrences of any type of modification were considered during the analysis.

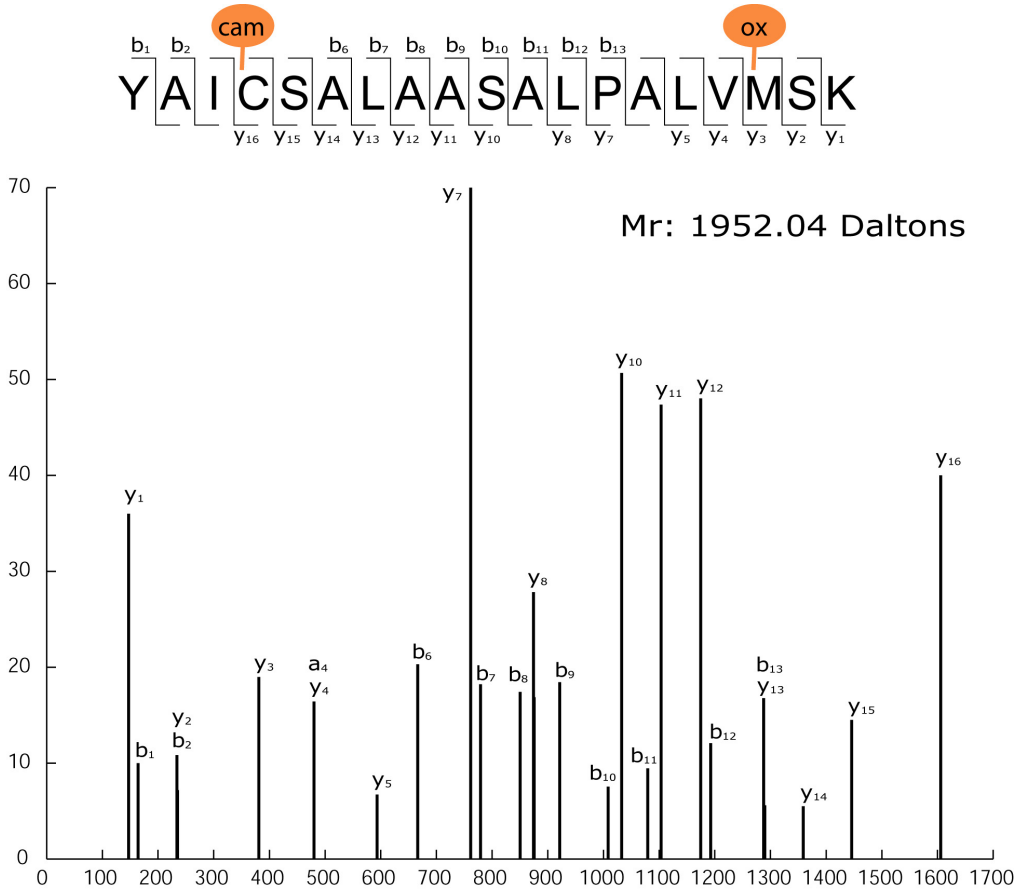



*Figure VI-2: An example-spectrum YAIC[cam]SALAASALPALVM[ox]SK.*
*This example-spectrum comes from a real experiment but it was modified by hand to remove noise peaks and to add some missing peaks. The peptide that produced the spectrum is the peptide YAIC[cam]SALAASALPALVM[ox]SK. It carries two modifications (a carbamidomethylation on the cysteine and an oxidation on the methionine).*

## VI.2. Terminology

A peptide P is a linear sequence of p amino acids taken from an alphabet Σ of size 20, with $P=\{a_1 a_2 ... a_p\}$. Each amino acid has a mass $\mu(a_i)$. The mass of an uncharged peptide P, denoted as $\mu(P)$, is the sum of all its constituent amino acids plus the mass of the N-terminal (H) and C-terminal (OH) groups.

$$\mu(P) = \sum_{i=1}^{p} \mu(a_i) + \mu(H) + \mu(OH)$$

*Equation VI-1*

In an MS/MS experiment, P can be cleaved between each amino acid. Each cleavage leads to two fragments, one with the N-terminus and one with the C-terminus (we do not consider internal fragments as they are of no –or little– use in methods using spectrum graph). In an ideal fragmentation situation (cleavage occurring exactly on the peptide bond without any atom gain or loss), the mass of a N-terminal fragment $\mu(F_n^{Nterm})$ ending at amino acid n can be computed by adding the amino acid masses located before the cleavage position and the mass of the N-terminal group.

$$\mu(F_n^{Nterm}) = \sum_{i=1}^{n} \mu(a_i) + \mu(H)$$

*Equation VI-2*

Similarly, the mass of a C-terminal fragment ending at amino acid n can be computed by adding the amino acid masses located after the cleavage position and the mass of the C-terminal group.

$$\mu(F_n^{Cterm}) = \sum_{i=n}^{|P|} \mu(a_i) + \mu(OH)$$

*Equation VI-3*

A spectrum S is composed of an observed precursor peptide mass-to-charge ratio, denoted as $\mu_{m/z}(P_{prec})$, of an integer c representing the precursor's charge state, and of a list L of peaks, with $L=\{s_1, s_2, \ldots, s_l\}$, each of them being characterized by a mass-to-charge ratio $\mu_{m/z}(s_i)$ and an intensity value $\iota(s_i)$. The charge state corresponds to the number of protons $H^+$ carried by the peptide. The non-charged molecular mass of the peptide can be computed according to:

$$\mu(P_{prec}) = \left(\mu_{m/z}(P_{prec}) \cdot c\right) - c \cdot \mu(H)$$

*Equation VI-4*

An ionic hypothesis η is a possible interpretation of a peak (each peak can have several different interpretations). Interpreting a peak means assigning it a terminus side $t(\eta)$, a number of charges $c(\eta)$, and an offset value $o(\eta)$. As we do not consider internal fragments, $t(\eta)$ is either "N-term" or "C-term". The number of charges is always > 0 as mass spectrometers only detect charged fragments.

The offset value corresponds to an ion type, as explained in Figure III-3 and represents the mass gains and losses that may occur during fragmentation of the peptide (including possible losses of molecules by amino acids, like water and ammonium). Table VI-1 gives examples of ionic hypotheses.

| η | t(η) | c(η) | o(η) |
|---|---|---|---|
| a+ | N | 1 | -28 [Da] |
| a+* | N | 1 | -45 [Da] |
| a+° | N | 1 | -46 [Da] |
| a++ | N | 2 | -28 [Da] |
| a++* | N | 2 | -45 [Da] |
| a++° | N | 2 | -46 [Da] |
| b+ | N | 1 | 0 [Da] |
| b+* | N | 1 | -17 [Da] |
| b+° | N | 1 | -18 [Da] |
| b++ | N | 2 | 0 [Da] |
| b++* | N | 2 | -17 [Da] |
| b++° | N | 2 | -18 [Da] |
| y+ | C | 1 | 2 [Da] |
| y+* | C | 1 | -15 [Da] |
| y+° | C | 1 | -16 [Da] |
| y++ | C | 2 | 2 [Da] |
| y++* | C | 2 | -15 [Da] |
| y++° | C | 2 | -16 [Da] |

*Table VI-1: ionic hypotheses*
*An ionic hypothesis η is characterized by a terminus side t(η), a number of charges c(η) and an offset value o(η). For example, a peak interpreted as a*++ is hypothesized to be a doubly charged a-ion type peak that lost an ammonium molecule. Ammonium losses are signified by a "*" while water losses are signified by a "°".*

## VI.3. Overview of Popitam's algorithm

Figure VI-3 overviews the main steps of Popitam's algorithm. Names of functions are given in italic. The sections of this chapter describe each of the eight steps numbered in the Figure.

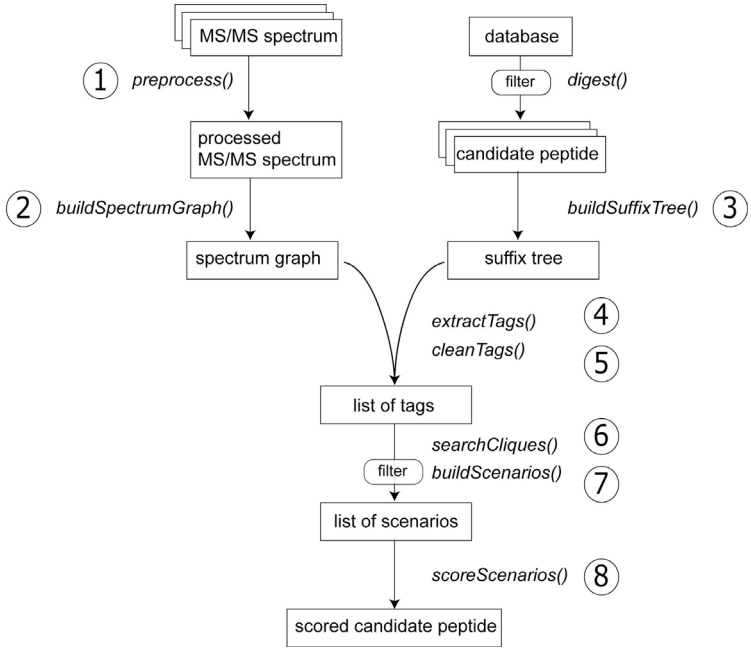



*Figure VI-3: Overview of Popitam's algorithm*
*Preprocessing (1) is tackled in Section VI.4; The spectrum building (2) is the subject of SectionVI.5. The suffix tree (3) and the tag extraction process (4) are part of Section VI.6. Section VI.7 discusses the tag cleaning (5) and Section VI.8 is about the cliques search. The scenarios are built in Section VI.9, and scored in SectionVI.10.8.*

## VI.4. Peak preprocessing

The MS/MS spectra used for identification are the result of automated signal processing algorithms that transform the raw measured signal into generic lists of peaks. Peak detection software include, with more or less success, peak centroiding, noise filtering, calibration, deisotoping and deconvolution. This low-level preprocessing is a key step that can noticeably influence the outcome of the identification (Gentzel et al., 2003). Higher preprocessing procedures are often performed on the peak lists. Although they are not systematically applied by identification software, they generally tend to enhance the identification quality. Such procedures, which are most often empirical, include removing non mono-isotopic peaks, filtering background noise, and deleting the precursor ion from the peak list, if present. The preprocessing steps performed by Popitam, which are listed below, are based on simple and empirical methodologies. As shown in Figure VI-4, they greatly reduce the number of peaks in the spectra, thus making their subsequent analysis easier. Nevertheless, we think that the procedure could (and should) be enhanced in the future. Notably, the extraction of isotopic peaks could be more developed and could use information about the expected isotope distributions for a given fragment mass.

a) Normalization of the peak intensities and removal of low intensity peaks
The function starts by normalizing the intensities of the peaks (the most intense peak is given an intensity of 100). Peaks below a given intensity threshold (e.g. 5% of the highest intensity) are removed from the spectrum.

b) Attribution of possible charge numbers to the peaks

Each peak is attributed possible charge numbers (1 or 2), according to their mass and the mass of the precursor. Peaks with masses greater than the precursor mass divided by two are considered as singly charged, while the other peaks are considered as either singly or doubly charged.

c) Isotopic peaks removal

The function sorts the processed peaks by mass and parses the list to locate isotopic peaks. For a peak $s_j$ to be considered as an isotope of a peak $s_i$, the following conditions are required:

i) $\mu_{m/z}(s_j) > \mu_{m/z}(s_i)$

ii) $(\mu_{m/z}(s_j) - \mu_{m/z}(s_i) - 0.5) < \varepsilon$ if both peaks have a possible charge number of 2

$(\mu_{m/z}(s_j) - \mu_{m/z}(s_i) - 1) < \varepsilon$ if both peaks have a possible charge number of 1

iii) $\iota(s_j) > \iota(s_i)$

*where*

*$\varepsilon$ is an error threshold specified by the user (e.g. 0.2);*

d) Merging of very close masses

It is not uncommon in a spectrum to observe peaks with very similar mass-to-charge values. When the signal extraction process is applied to the raw MS/MS spectrum, a single signal can be improperly interpreted as two different peaks (but of very similar masses). Using a greedy procedure and an error threshold, Popitam merges such peaks. When two peaks are merged, the new peak is given the average mass and the highest intensity of the two original ones.

e) Categorizing peaks into intensity ranks

Finally, peaks are sorted by intensity and are attributed a bin number according to their intensity category. The size n of the bins is directly computed from the precursor mass and represents the estimated amino acid length of the precursor peptide (rounded to the nearest integer).

$$n = \frac{\mu(P_{prec})}{\sum_{i=0}^{20} f(a_i) \cdot \mu(a_i)}$$

*where*

*$\mu(P_{prec})$ is the uncharged precursor mass*

*$f(a_i)$ is the observed frequency of amino acid i (computed for example in a reference protein database), with $0 \leq f(a_i) \leq 1$, and*

*$\mu(a_i)$ is its monoisotopic mass.*

The n most intense peaks were put into bin number 1, the n following into bin number 2, and so on until bin number 10.

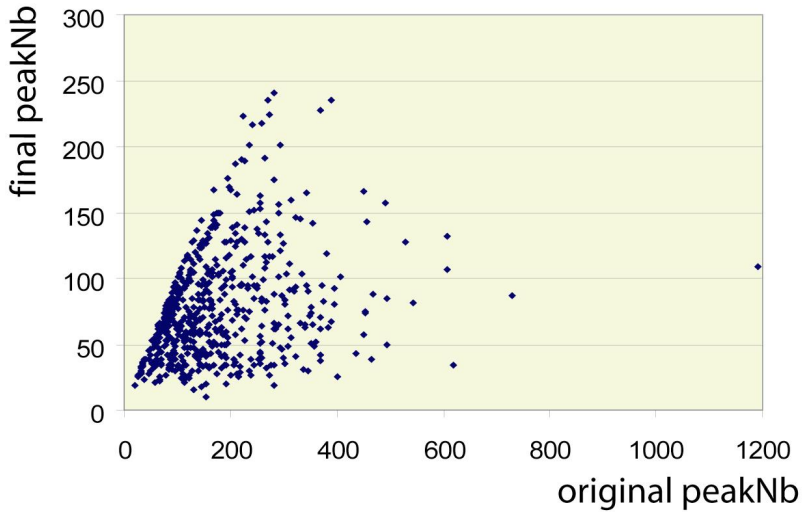



*Figure VI-4: Number of peaks before and after preprocessing*
*Scatter graph representing 597 spectra according to their number of peaks before and after preprocessing.*

## VI.5. Spectrum graph building

The spectrum graph is a widely used structure in *de novo* sequencing methods. In short, a spectrum graph is a structured representation of a spectrum. Nodes represent masses of N-terminal fragments, or "prefix residue masses" (PRM) (Bandeira et al. 2004). Given a peptide $P = \{a_1, a_2, ..., a_p\}$, the PRM of a prefix ending at position n corresponds to the N-terminal fragment mass

$$\mu(F_n^{Nterm}) = \sum_{i=1}^{n} \mu(a_i) + \mu(H)$$

Two nodes that differ by the mass value of one or a combination of amino acids (usually two) are connected by an edge labeled by the corresponding amino acid(s). Paths in a graph represent all amino acid tags and complete sequences that can be inferred *de novo* from the spectrum.
The construction of the spectrum graph in Popitam can be divided into four distinct steps. The first one is the re-expression of all peaks of the preprocessed spectrum as PRMs, according to a set of ionic hypotheses; the second one is the grouping of similar PRMs into clusters; the third one is the selection of clusters that will form the nodes of the spectrum graph; and the fourth one is the connection of the graph nodes.

### VI.5.1. Peak re-expression as prefix residue masses

In a spectrum graph, two nodes are connected when their mass differ by the mass value of one or several amino acids. This requires two node masses to be comparable, which is not the case if they represent peaks of different ionic types. Thus, the first step in the construction of a spectrum graph is to re-express all the peak masses into PRMs. This procedure, which amounts to attribute to each peak (of unknown ion type) different ionic hypotheses from a set $\Delta=\{\eta_1, \eta_2,..., \eta_{|\Delta|}\}$, is illustrated in Figure VI-5. The procedure implies the use of Algorithm VI-1 to transform peak mass-to-charge ratios $\mu_{m/z}(s_i)$ into prefix residue masses $PRM(s_i, \eta_k)$.

$$\text{if } \left(t(\eta_k) = \text{'N'}\right)$$
$$PRM(s_i,\eta_k) \leftarrow c(\eta_k)\cdot \mu_{m/z}(s_i) - \left(c(\eta_k)-1\right)-o(\eta_k)$$
$$\text{if } \left(t(\eta_k) = \text{'C'}\right)$$
$$PRM(s_i,\eta_k) \leftarrow \mu(P_{prec}) - \left[c(\eta_k)\cdot \mu_{m/z}(s_i) - \left(c(\eta_k)-1\right)-o(\eta_k)\right]$$

*Algorithm VI-1: Transforming mass-to-charge ratios into PRMs*
*Transformation of a peak mass-to-charge ratio $\mu_{m/z}(s_i)$ into a prefix residue mass $PRM(s_i, \eta_k)$, given a precursor mass $\mu(P_{prec})$ and an ionic hypothesis $\eta_k$ composed of an offset value $o(\eta_k)$, a terminus side $t(\eta_k)$ (N-term or C-term) and a number of charges $c(\eta_k)$:*

It should be noted that one of the members of the second formula being the measured precursor mass, its error, which is generally greater than the error observed for fragment masses, is transmitted to the computed PRMs. Methods exist that refine the precursor mass using complementarity between b- and y-ion type peaks (see (Dancik et al. 1999) for more details). Popitam handles the issue by using two error thresholds, FRAGMENT_ERROR1 and FRAGMENT_ERROR2. The former is used when comparing PRMs of similar terminus sides, while the latter, which is greater, is used when comparing PRMs of different terminus sides. Typical FRAGMENT_ERROR1 and FRAGMENT_ERROR2 values are 0.2 and 0.4 [Da] respectively.

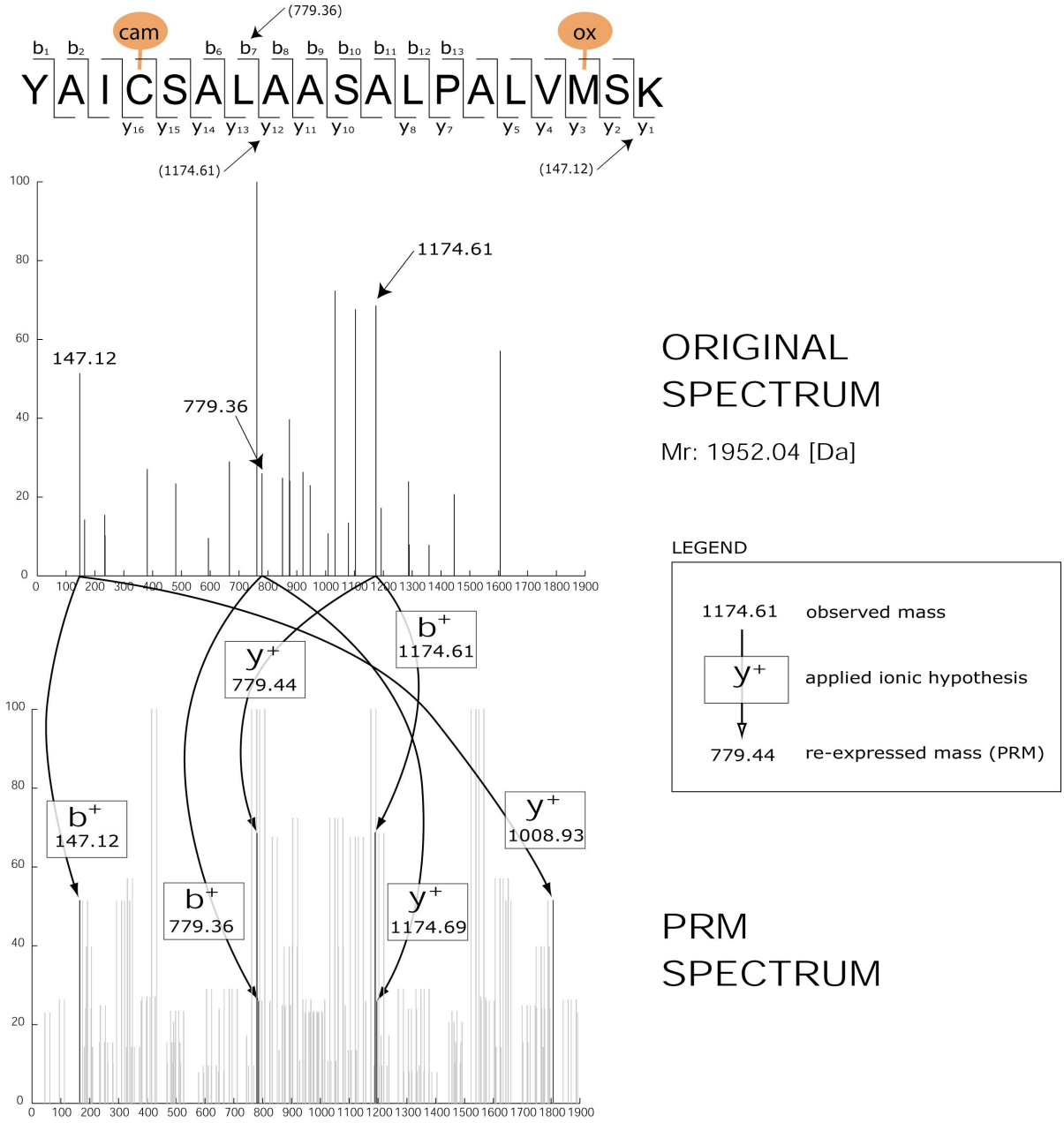



*Figure VI-5: Peak re-expression*

*$YAIC^{[cam]}SALAASALPALVM^{[ox]}SK$-spectrum before (top) and after (bottom) re-expression of the peaks as prefix residue masses (PRMs).*

*PRMs are computed from the peak mass-to-charge ratios by applying in turn ionic hypotheses of a set Δ. In this case, the 18 ionic hypotheses enumerated in Table VI-1 were applied. For a given peak, at most one of the obtained PRMs is correct, all the other ones being issued from false hypothesis assignments. Arrows allow one to follow the interpretation process for three particular peaks (at m/z 147.12, 779.36 and 1174.61), and two specific ionic hypotheses ($b^+$ and $y^+$). The first peak, 147.12 is actually the y1-fragment of peptide $YAIC^{[cam]}SALAASALPALVM^{[ox]}SK$. Then, the PRM 147.12, which is based on the assumption that the original peak is of b-ion type, is a false positive, while the PRM 1008.93 is correct. Similarly, peak 779.36 being the b7-ion, the PRM 779.36 is correct, while the PRM 1174.69 is a false positive. When two different peaks originate from the same fragmentation position, their correct PRMs converge to a unique m/z value (as can be observed for PRMs of peaks 779.36 and 1174.61). The same phenomenon happens with incorrect hypothesis assignments.*

Given |S| the number of peaks in a spectrum, and |Δ| the number of ionic hypotheses, the re-expression procedure leads to a set of |s|*|Δ| PRMs. As highlighted in the legend of Figure VI-5, at most |S| of the obtained PRMs are correct, all the other ones being issued from false hypothesis assignments. Such a situation is unfavorable, since most of the nodes in the graph will come from false assignments, thus complicating the graph interpretation. To handle this issue, we associate with each PRM a confidence score σ(PRM), on which we base the subsequent selection of PRMs to integrate in the graph. The score represents the probability that the ionic hypothesis is correct, given the intensity rank of the peak, the number of charges carried by the precursor peptide and the mass spectrometer type. These probabilities are read from pre-built tables, according to a method described in (Dancik et al. 1999) and explained below.

We used a set of 704 doubly charged MS/MS spectra obtained with a Q-TOF mass spectrometer. Each spectrum of the set was confidently correlated with a peptide sequence. For each spectrum, the peaks were ranked by intensity and grouped into bins (see Section VI.4). Masses of N-terminal fragments were computed for each cleavage position on the peptide sequence assigned to the current spectrum according to Equation VI-2. Offsets (which are specific for ion-types) between observed peak masse-to-charge ratios and the computed masses were reported and incremented in a 3D plot (see Figure VI-6, left plot). After each spectrum has added its contribution to the 3D plot, the process was repeated with C-terminal fragment masses and a second 3D plot (see Figure VI-6, right plot). Two additional plots were created for doubly charged N-terminal and C-terminal fragment masses. Finally, occurrence probabilities of a given ion type were estimated by dividing the count reported for the corresponding offset by the number of peaks in the bin (see Table VI-2).

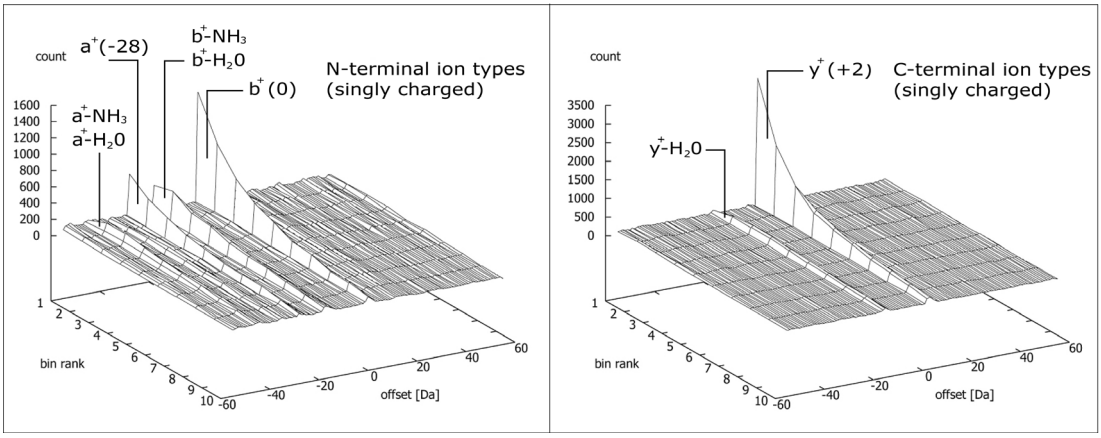



*Figure VI-6: Ion occurrence plots*
*The plots count singly charged ion types observed in a set of 704 doubly charged spectra obtained with an ESI-QTOF mass spectrometer. With such plots, one can easily spot what ion types are produced during the peptide fragmentation. Moreover the relation between peak intensity and ion-type frequency is highlighted. The distributions show that the frequencies fall with decreasing intensity, except for ions with water or ammonium molecule losses. For these ions, the tendency is reversed from bin 1 to bin 2.*

| ion type | offset | bin1 | bin2 | ... | bin10 |
|---|---|---|---|---|---|
| a | -28 | 5.24 | 4.16 | ... | 2.4 |
| a* | -45 | 0.95 | 1.5 | ... | 1.63 |
| a° | -46 | 0.76 | 1.15 | ... | 1.73 |
| b | 0 | 12.19 | 11.22 | ... | 2.12 |
| b* | -17 | 1.43 | 4.36 | ... | 2.5 |
| b° | -18 | 3.33 | 3.36 | ... | 2.6 |
| a++ | -28 | 0.48 | 0.55 | ... | 0.38 |
| a*++ | -45 | 0.33 | 0.55 | ... | 0.48 |
| a°++ | -46 | 0.19 | 0.1 | ... | 0.38 |
| b++ | 0 | 0.71 | 0.7 | ... | 0.67 |
| b*++ | -17 | 0.38 | 0.6 | ... | 0.77 |
| b°++ | -18 | 0.48 | 0.45 | ... | 0.48 |
| y | 2 | 35.52 | 18.89 | ... | 3.75 |
| y* | -15 | 0.81 | 3.46 | ... | 2.5 |
| y° | -16 | 1.24 | 2.61 | ... | 3.08 |
| y++ | 2 | 1.76 | 2.4 | ... | 1.35 |
| y*++ | -15 | 0.24 | 0.95 | ... | 0.87 |
| y°++ | -16 | 0.52 | 0.9 | ... | 1.06 |

*Table VI-2: Occurrence probabilities (as percent)*
*The table list occurrence probabilities for various ion types and peak intensities computed from a set of 704 doubly charged spectra (ESI-QTOF). In our dataset, 35.52% of the peaks in bin number 1 were of y-ion type, and 12.19% of the peaks in bin number 1 were of b ion-types. The first column sums to 66.56%. This means that 33.44% of high intensity peaks did not correspond to either of the 18 ion types.*

The methodology described above has been applied on two datasets, one from a Q-TOF and one from a TOF-TOF mass spectrometer, producing three tables: two for MS/MS data obtained with the Q-TOF (one for doubly charged spectra and one for triply charged spectra), and one table for singly charged spectra obtained with the TOF-TOF. For each spectrum to identify, Popitam uses the corresponding probability file.

### VI.5.2. PRM clustering

As shown in Figure VI-5, when a given cleavage position is represented by several different ionic peaks in the spectrum, their PRMs converge to single values. Consequently, after all peaks have been re-expressed, similar PRMs are hypothesized to represent a single fragment mass and are grouped into clusters. At the end of the process, an average PRM is computed for each cluster. The clustering algorithm used by Popitam is depicted in Figure VI-7.

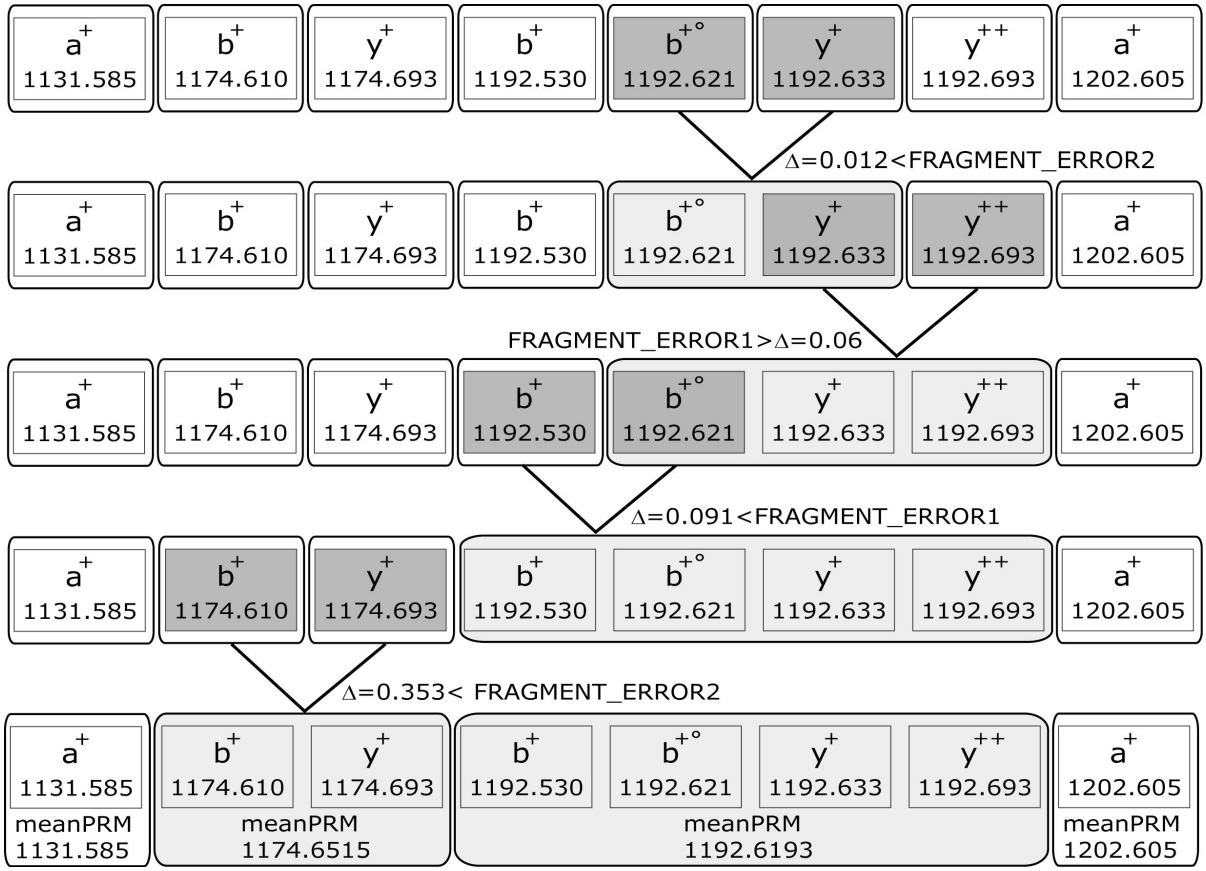



*Figure VI-7: Grouping PRMs into clusters*
*This example is based on a subset of PRMs computed from the spectrum represented in Figure VI-5. PRMs and their associated ionic hypotheses are placed in squares (PRMs scores are not shown). Clusters are shown by rounded boxes. The algorithm first associates every PRM to a cluster (composed of itself). Then, iteratively, it searches for the two closest PRMs (in different clusters and of different ionic hypotheses). If the mass difference between the two clusters is lower than a certain threshold (FRAGMENT_ERROR1 if the terminus sides are the same, otherwise FRAGMENT_ERROR2), all the PRMs of one cluster are inserted into the other one, and the empty cluster is deleted. The process stops when there are no more PRMs in different clusters and of different ionic hypotheses having a mass difference smaller than the chosen threshold.*

### VI.5.3. Node sampling

Nodes of the spectrum graph are directly formed from PRMs clusters. Therefore, nodes include all original information of the clusters. This includes the complete list of PRMs with their confidence scores, and for each PRM, indices to the original peak and applied ionic hypothesis. It is therefore very convenient to access any piece of information from a node at any moment during the identification run.

As most clusters are issued from false ionic hypothesis assignments, only clusters including PRMs with high-confidence scores are selected to be part of the graph. The procedure is the following. First, a desired number N of nodes to include in the graph is computed. N is computed according to:

$$N = \frac{\mu(P_{prec})}{111} \cdot COVBIN$$

*where*

$\mu(P_{prec})$ *is the observed precursor mass*

*111 is the average weight of an amino acid*

*COVBIN is a coverage parameter*

The first term of this equation represents the estimated length of the precursor peptide. If COVBIN is set to 4, the number of nodes is chosen so as to cover 4 times the number of possible cleavages of the peptide sequence.

Once N is computed, the clusters are sorted by decreasing order according to their highest σ(PRM). Finally, the N first PRM clusters of the sorted list are selected to form the graph. Two "virtual" nodes complete the set: the first one represents the empty sequence (PRM = μ(H)) and the second one represents the complete sequence (PRM = $\mu(P_{prec})$- μ(OH)). The former is the node with the smallest PRM of the graph, and the latter is the node with the largest PRM.

It should be noted that PRM selection could be optimized in the future to include more information. For exemple, Tanner et al. (Tanner et al. 2005) estimated the ionic frequencies for different sectors of the spectrum.

### VI.5.4. Graph connection

During the graph connection procedure, pairs of nodes $v_i$ and $v_j$ are connected by an edge $e_{ij}$ if any of the PRMs in $v_i$ differs from any of the PRMs in $v_j$ by the mass value of one or two amino acids, given an error threshold. The error depends on the ionic hypothesis terminals of the two PRMs (FRAGMENT_ERROR1 is used when the terminals are the same, FRAGMENT_ERROR2 is used otherwise). Edges therefore represent amino acids, and the set of all paths in the graph represents the set of all possible amino acid sequences that can be inferred from the peak pattern of the spectrum. "Double edges" allow jumping over one missing fragmentation position (all positions in a peptide are not necessarily cleaved during the fragmentation process). It would be possible to extend this concept to triple edges and more, but since the number of edges increases exponentially with their length, this would not be realistic. By convention, we will refer to simple edges with upper case letters, and to double edges with lower case letters. Figure VI-8 shows an example of spectrum graph obtained from an MS/MS spectrum.

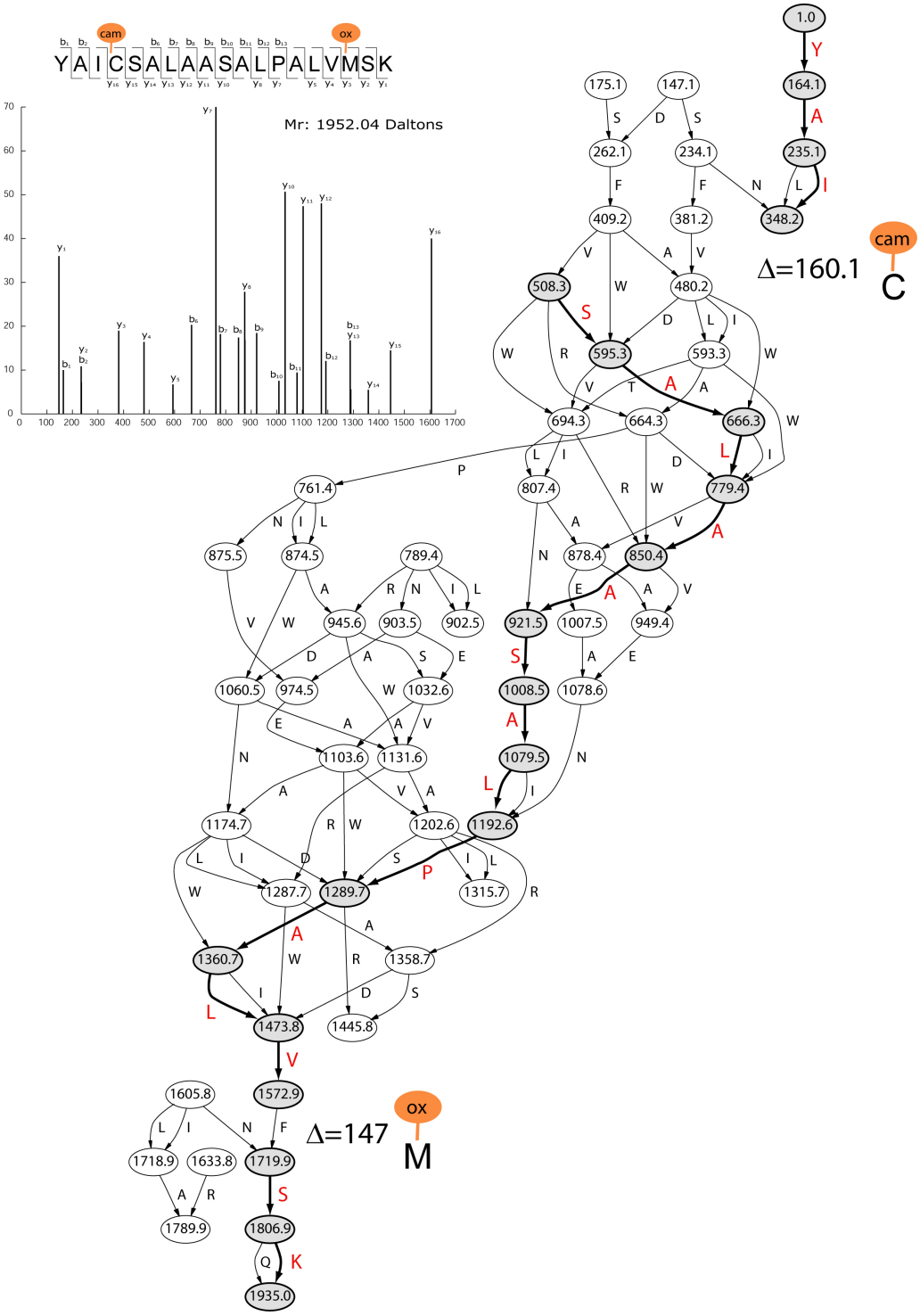



*Figure VI-8: Spectrum YAIC[cam]SALAASALPALVM[ox]SK and its spectrum graph*
*COVBIN was set two 3 (greater coverage values are generally chosen with real data) and only simple edges were considered during the graph construction. FRAGMENT_ERROR1 and FRAGMENT_ERROR2 were set to 0.2 and 0.4 respectively. For the sake of reading clarity, nodes are signified by their average PRM. The correct path (shown in bold edges and shadowed nodes) is the path that corresponds to the correct peptide sequence and that uses nodes obtained with correct ionic hypotheses. The effect of the presence of modified amino acids is clearly illustrated. The first modification (C[cam]) splits the path into two unconnected parts (it is not possible to return to the path from node 348.2). For the second modification (M[ox]), an alternative path (edge corresponding to amino acid F) can be used to go from node 1572.9 to node 1719.9.*

## VI.6. Tag extraction

A first version of the algorithm (Hernandez et al. 2003), called "Full Path Algorithm", was aimed at finding complete sections in the graph (a path starting from the first vertex and corresponding to a whole peptide sequence) that best fitted the current candidate peptide (see Appendix A5). The parsing was performed using an Ant Colony Optimization algorithm (ACO) (Dorigo and Di Caro 1999). ACO algorithms are defined as multi-agent systems inspired from real ant colony behavior. Their principle is the following: a population of ants explores, iteratively and simultaneously, different paths in the spectrum graph by moving from node to node using available edges. Ant's exploration is guided by a visibility factor (represented by the score of the nodes), and by a variable associated to edges and representing pheromone trails deposited during previous explorations. After a certain number of iterations, the ACO converges to globally interesting paths. A detailed description of the ACO algorithm can be found in (Hernandez et al. 2003), as well as some results obtained with this method. A significant drawback of the "Full Path Algorithm" was that it showed limited performance on spectra with poor peak statistics. Popitam was struggling with the identification of spectra having two or more missing fragmentation positions, because they resulted into split spectrum graphs and then forced the ants to lose the correct path. In addition, the "Full Path Algorithm" was not able to handle unexpected PTMs.

The solution came from tag extraction. By looking for local sections in the spectrum, we freed Popitam from several hindrances. First, calibration was not any more a problem, since the parsing did not necessarily include the extremities of the graph, which have arbitrary fixed masses. Then, the problem of split spectra due to poor fragmentation or to the presence of unexpected PTMs was alleviated. In (Hernandez et al. 2003), we presented results obtained with a preliminary version of Popitam's "tag algorithm". That version did not contain elaborate scoring functions and was using a factor oracle automaton (Allauzen et al. 1999) to index the subsequence of the candidate peptides. The tag extraction procedure was similar to the current version of Popitam. The spectrum graph and the automaton were simultaneously parsed, and only tags that were coherent with both the MS/MS spectrum and the current candidate peptide were extracted. As highlighted in Figure VI-9, a drawback of this structure is that available paths do not correspond to real subsequences of the candidate peptide, producing "artifact" tags. We therefore finally chose a structure based on a suffix tree (Gusfield, Cambridge University Press, New York, 2004) (see Figure VI-9).

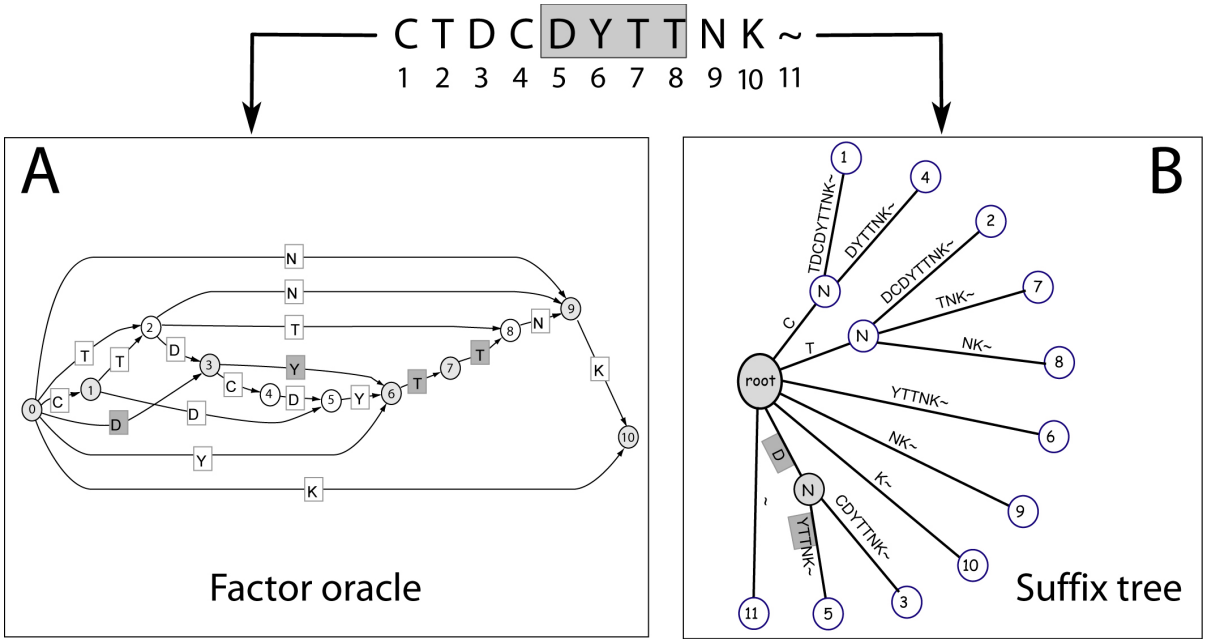



*Figure VI-9: A factor oracle and a suffix tree*
*A factor oracle of a string S of length n is an automaton with exactly n states (circles). All states can be terminal. Starting from state 0, one can build at least all possible subsequences of S. In the example A, path 0-3-6-7 corresponds to sequence DYTT and is actually a subword of CTDCDYTTNK. Path 0-1-2-9-10 corresponds to sequence CTNK, which is not a subword of CTDCDYTTNK.*
*A suffix tree of a string S of length n is a rooted directed tree with exactly n leaves, each of them representing a different suffix of the string starting at a position p. Concatenation of the edge-labels on a path from the root to a given internal node spells out a repeated factor of S.*

Tag extraction is performed by simultaneously parsing the graph structure and the suffix tree. The procedure, depicted in Figure VI-10 and Algorithm VI-2, consists in recursively parsing the graph and using the suffix tree as a checking table. A similar approach (with complete paths) was used in (Lu and Chen 2003a). In Popitam, as we are interested in tags, the search is performed from each node of the graph and is exhaustive (we do not use any more the ACO algorithm). By this mean, Popitam can extract all tags that are both consistent with the spectrum peak pattern and the candidate peptide (given a minimum tag length).

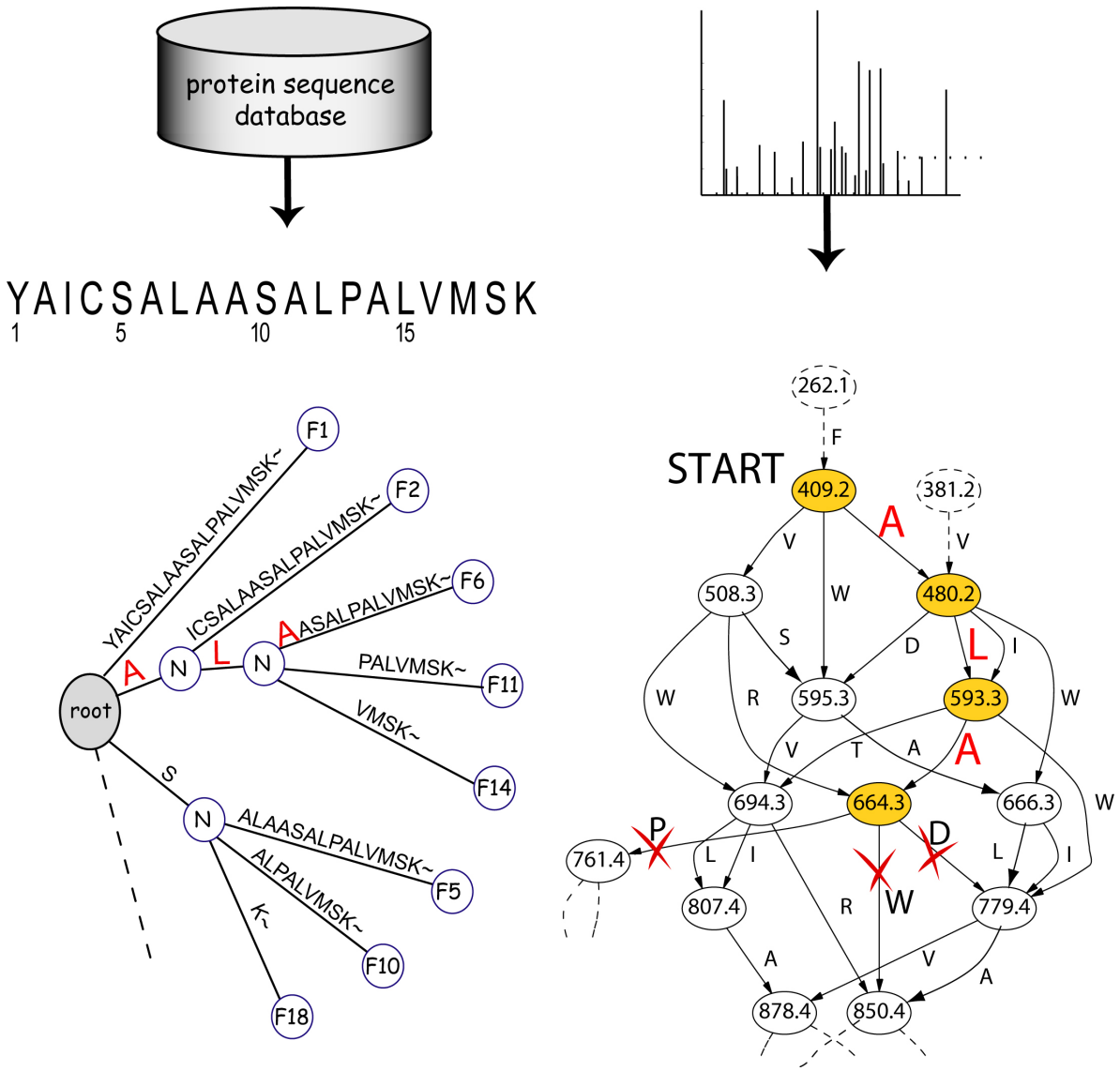



*Figure VI-10: Tag extraction*

*Tag extraction is performed by simultaneously parsing the suffix tree (left) and the graph (right). The parsing is carried out from each node of the graph. At the beginning, the first node is used as a start. In this example, the start node is node 409.2. Exploration proceeds recursively through the graph as long as the parsed path matches a subsequence of the candidate peptide (the suffix tree is used as a checking table). If a tag cannot be anymore elongated, it is stored. Usually, the minimum length for a tag to be stored is set to three nodes. Recursivity allows backtracking to an anterior state in both the graph and the suffix tree and exploring new paths. When all paths have been tested from a given node, a new search is launched from another one, until each node has been used as start for a search.*

```
For each node v_u ∈ G {
   path = {};
   exploreGraph(u, root(T));
}

void exploreGraph(pos(G), pos(T)) {
   For each next a_j from pos(G) {
      For each next a_k from pos(T) {
         if (match(a_j, a_k)) {
            path = path ∪ (pos(G), a_j);
            newPos(G)=moveInGraph();
            newPos(T)=moveInTree();
            exploreGraph(newPos(G), newPos(T));
         }
         else {
            if (size(path) > MIN_TAG_LENGTH) {
               storeTag(path);
            }
         }
      }
   }
}
```

*Algorithm VI-2: Tag extraction*
*The search is launched from each node $v_u$. Two indices, pos(G) and pos(T) store the current positions respectively in the graph and in the tree. The two loops enumerate the successive amino acids in the graph and amino acids in the tree. In the worst case, the iteration number of the suffix tree loop is $\Sigma$ (the number of possible amino acids), and the iteration number of the graph loop is $\Sigma^2$ (all possible two amino acid combinations). In case of match, the current path and both positions are updated, and the function is recursively called. In case of mismatch, the current path is stored if its length is larger than the parameter MIN_TAG_LENGTH.*

Tag extraction is the most time consuming process of Popitam: each node in the graph may start a tag, and the extraction process is performed for each candidate peptides. According to Algorithm VI-2, the "worst case" complexity (for one candidate peptide) is:

$$O\left(m \cdot L \cdot \Sigma^3\right)$$

*where*

*m is the number of nodes in the spectrum graph*
*L is the length of the candidate peptides*
*$\Sigma$ is the number of distinct amino acids*

Fortunately, both the spectrum graph and the suffix tree are not fully connected and thus, the term $\Sigma$ is overestimated. For example, in Section VIII.3.3, the spectrum graphs of the various presented

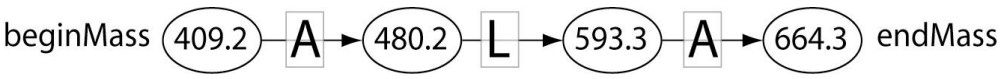
spectra have an average connection rate of 68 edges per node. As we used large fragment errors for these examples, this corresponds to a particularly high connection rate.

The extracted tags have several attributes. First, they represent a sequence, obtained by concatenating the parsed edges. The list of parsed nodes is also stored, as well as the position of the tag in the peptide sequence (this information is given by the suffix tree). The mean PRM of the first node of a tag is called *BeginMass*, and the mean PRM of the last node of a tag is called *EndMass*.

The simultaneous parsing of the spectrum graph and of the suffix tree results in a list of tags. Each tag of the list represents a subsequence of the candidate peptide and is consistent with the spectrum peak pattern. Usually, comparisons of the spectrum with incorrect candidate peptides result in tags of very short length, while comparison of the spectrum with a correct candidate results in longer tags.

## VI.7. Tag cleaning

The tag extraction algorithm does not prevent to extract subtags of tags, i.e. tags that are completely included in others. This problem arises because the recursive parsing is launched independently from every node. For example, the tag




is reported when searching from node 409.2 of the spectrum graph of Figure VI-8, while the tag


is reported when the search is launched from node 480.2. As the second tag does not bring supplementary information, it is discarded. Figure VI-11 shows the list of tags extracted during comparison of the example-spectrum YAIC[cam]SALAASALPALVM[ox]SK with the correct (unmodified) candidate peptide.

| | | | |
|---|---|---|---|
| 0 | 1.0 | YAI | 348.2 |
| 5 | 164.1 | AL | 348.2 |
| 10 | 164.1 | AL | 348.2 |
| 13 | 164.1 | AL | 348.2 |
| 1 | 164.1 | AI | 348.2 |
| 5 | 409.2 | ALA | 664.3 |
| 6 | 480.2 | LA | 664.3 |
| 1 | 409.2 | AI | 593.3 |
| 4 | 508.3 | SALAASALPALV | 1572.9 |
| 5 | 595.3 | ALAASALPALV | 1572.9 |
| 6 | 666.3 | LAASALPALV | 1572.9 |
| 7 | 779.4 | AASALPALV | 1572.9 |
| 8 | 850.4 | ASALPALV | 1572.9 |
| 9 | 921.5 | SALPALV | 1572.9 |
| 10 | 1008.5 | ALPALV | 1572.9 |
| 11 | 1079.5 | LPALV | 1572.9 |
| 12 | 1192.6 | PALV | 1572.9 |
| 13 | 1289.7 | ALV | 1572.9 |
| 14 | 1360.7 | LV | 1572.9 |
| 1 | 595.3 | AI | 779.4 |
| 13 | 595.3 | ALV | 878.4 |
| 14 | 666.3 | LV | 878.4 |

| | | | |
|---|---|---|---|
| 6 | 694.3 | LAA | 949.4 |
| 7 | 807.4 | AA | 949.4 |
| 6 | 761.4 | LA | 945.6 |
| 8 | 874.5 | ASA | 1103.6 |
| 4 | 945.6 | AS | 1103.6 |
| 9 | 945.6 | AS | 1103.6 |
| 1 | 1008.5 | AI | 1192.6 |
| 7 | 1032.6 | AA | 1174.7 |
| 5 | 1103.6 | ALA | 1358.7 |
| 6 | 1174.7 | LA | 1358.7 |
| 1 | 1103.6 | AI | 1287.7 |
| 7 | 1060.6 | AASAL | 1473.8 |
| 8 | 1131.6 | ASAL | 1473.8 |
| 9 | 1202.6 | SAL | 1473.8 |
| 4 | 1202.6 | SAL | 1473.8 |
| 1 | 1289.7 | AI | 1473.8 |
| 5 | 1131.6 | AL | 1315.7 |
| 10 | 1131.6 | AL | 1315.7 |
| 13 | 1131.6 | AL | 1315.7 |
| 1 | 1131.6 | AI | 1315.7 |
| 8 | 1287.7 | AS | 1445.8 |
| 6 | 1605.8 | LA | 1789.9 |
| 17 | 1719.9 | SK | 1935.0 |

*Figure VI-11: List of extracted tags*
*The tags were extracted during comparison of the YAIC[cam]SALAASALPALVM[ox]SK spectrum with candidate peptide YAICSALAASALPALVMSK.*
*Values in the first column are the positions of the tags in the peptide sequence, values in shadowed boxes represent the tag's beginMasses and endMasses. Among the 45 tags, 21 were retained for further processing, the other ones being discarded (light gray).*

The number of tags found by Popitam varies according to the minimal size of the tags (parameter MIN_TAG_LENGTH) and to the mass of the precursor peptide (more tags are extracted for long peptides). In addition, a third parameter, called MODGAPNB, fixes the number of gaps due to modification events in a scenario. Thus, when MODGAPNB is set to 0, Popitam does not make any modification hypotheses and removes from the list all tags with flanking masses not compatible with the candidate peptide sequence (see Figure V-4). This results into an important decrease of the number of tags extracted per candidate peptide (Figure VI-12).

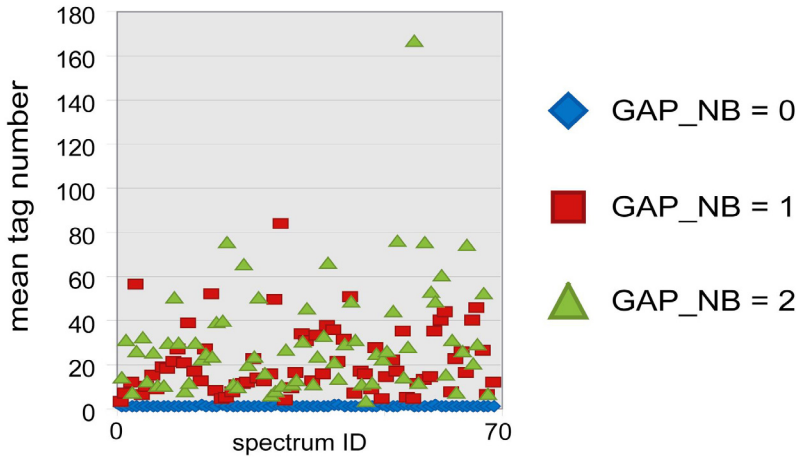



*Figure VI-12: Average number of tags per candidate peptide*
*The plot represents the average number of tags (per candidate peptide) computed for 70 spectra (x-axis). Data were collected from runs performed for Chapter VIII. Minimum tag length was set to 3. When run in mode MODGAPNB=0, Popitam discards all tags with flanking masses not compatible with the candidate peptide sequence. Consequently, the average number of tags is very low (about 1.45 tag per candidate peptide). In mode MODGAPNB = 1 and MODGAPNB = 2, the average number of tags reaches 25.2 per candidate peptide.*

## VI.8. Listing possible "run-and-jump" paths

The next step is to test different possible interpretations of the spectrum, given the current candidate peptide, the list of extracted tags, and compatibility rules between the tags. A possible interpretation is a path of the spectrum graph that runs trough edges or jumps from node to nodes (see Figure VI-18). We call such a path a "run-and-jump" path. Jumps correspond either to a lack of information in the spectrum (in which case they are called *lackGaps*), or to the presence of one or several modifications (*modGaps*). The parsed edges correspond to the tags. Run-and-jump paths are built by determining combinations of tags that are compatible with each other.

If all tags were compatible, this would amount to enumerate all ways of selecting k (unordered) tags among n (n is the number of extracted tags), for k varying from 0 to n, therefore producing

$$\sum_{k=1}^{n} C_n^r = \sum_{k=1}^{n} \frac{n!}{r!(n-r)!}$$

interpretations. Fortunately, by defining tag compatibility rules, most of the combinations can be discarded.

We use four logical rules: the overlap rule, the contiguity rule, the node sharing rule and the precedence rule. The first three rules are based on the fact that there is only one correct interpretation of the spectrum in the graph. The fourth rule is based on the tag positions and flanking masses. Each rule is described below and illustrated with a concrete example. The examples refer to the comparison

of the YAIC[cam]SALAASALPALVM[ox]SK spectrum (Figure VI-8) with the correct unmodified candidate peptide sequence.

a) Logical rule number 1: Overlap rule

Logical rule number one states that two overlapping tags are not compatible (Figure VI-13). Lets consider the spectrum graph of Figure VI-8. In this graph, a certain number of nodes were obtained by assigning correct ionic hypotheses to the set of peaks in the spectrum, all the other ones being false positive nodes. Correct nodes are represented in shadowed ovals and correspond to a correct interpretation that runs-and-jumps through a path from the first node (with PRM 1.0) to the last node (with PRM 1935.0 corresponding to the precursor mass minus OH). In an interpretation built by Popitam, the tags are considered "a priori" to be on the correct path (although there is no indication of where the correct path is in the spectrum). Therefore, since there can be only one correct path in the spectrum and subtags have been discarded, two overlapping tags cannot belong to the same interpretation and are said to be incompatible.

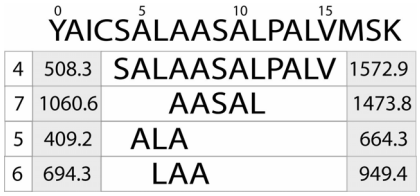



*Figure VI-13: Overlapping tags*

*Only the first tag corresponds to the correct path in the graph. The three other tags (AASAL, ALA, LAA) may come from "random" connections, or from paths due to repeated false interpretations of peak series (see Figure III-5 for a definition of peak series).*

b) Logical rule number 2: Contiguity rule

Logical rule number two states that two contiguous tags are not compatible (Figure VI-14). The reason is that two contiguous tags that would belong to the correct path would not be reported by the extraction process, since the latter always looks for the longest possible tags that match a subsequence of the candidate peptide. Consequently, instead of two contiguous tags, it would have reported only one.

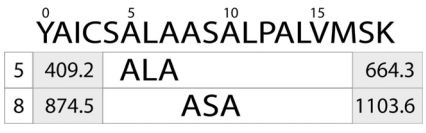



*Figure VI-14: Two contiguous tags.*

*If both tags were on the correct path, the tag extraction process would have reported one tag (ALAASA) instead of two tags*

c) Logical rule number 3: Node-sharing rule

Logical rule number three states that two tags sharing one or more nodes are not compatible (Figure VI-15). The thought process is similar to the first rule. As there can be only one correct path, a node cannot belong to two “correct” tags.

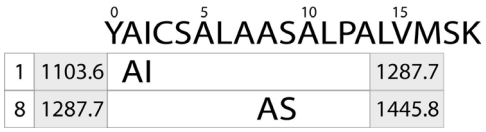



*Figure VI-15: Tags that share nodes*
*Tag AI and tag AS do not fulfill the node-sharing rule because they share the node with PRM 1287.7 (see alsoFigure VI-8).*

c) Precedence rule

The precedence rule states that two tags are considered as incompatible if there is incoherence between their respective positions on the sequence and their *beginMasses* and *endMasses* (Figure VI-16).

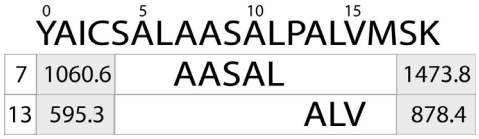
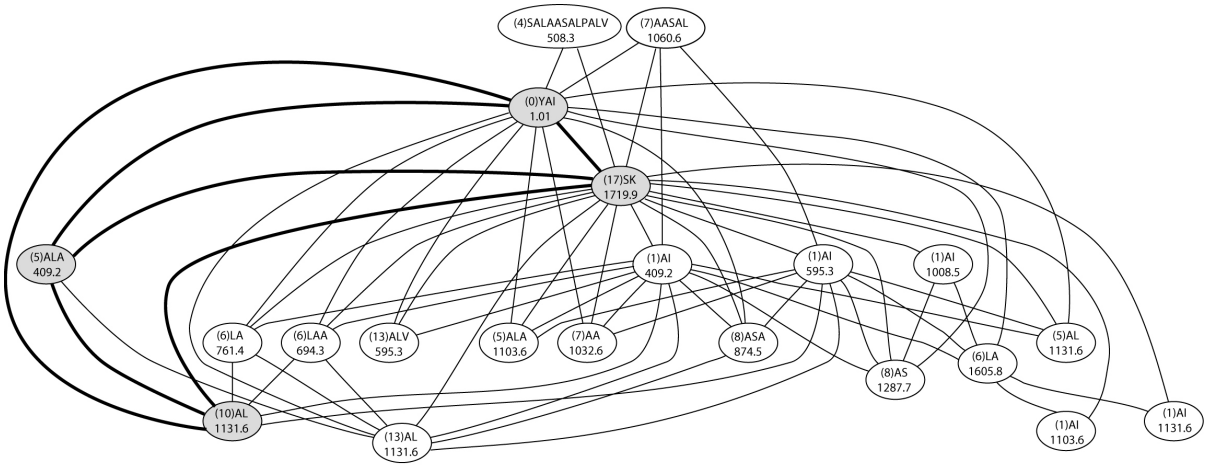



*Figure VI-16: Tags with non logical positions*
*Tag AASAL and tag ALV do not fulfill the precedence rule: ALV is positioned after AASAL on the peptide sequence, but its beginMass is smaller.*

In order to enumerate groups of compatible tags, we build a tag compatibility graph in which each node represents a tag and edges connect tags that fulfill the four logical rules. Then, we look for all cliques in the tag compatibility graph, a clique being a fully connected region of the graph (see Figure VI-17). We start by looking for cliques of size 2. Then we increase the clique size until it is equal to the total number of tags. The search is stopped if no clique is found for a given size s (it is clear that this process guarantees that no longer clique can exist). Although this computation is NP-complete, it is performed in a reasonable time, since the number of tags is kept quite low by using appropriate filtering parameters (MIN_TAG_LENGTH sets the minimal size of the tags, COVBIN sets the number of nodes in the graph and the FRAGMENT_ERRORs set the connection rate of the graph). Furthermore, the compatibility rules make cliques of important sizes unlikely.




0 5 10 15
YAICSALAASALPALVMSK

| | | | |
|---|---|---|---|
| 0 | 1.0 | YAI | 348.2 |
| 5 | 409.2 | ALA | 664.3 |
| 10 | 1131.6 | AL | 1315.7 |
| 17 | 1719.9 | SK | 1935.0 |

*Figure VI-17: A tag compatibility graph and a four-tag clique*
*The compatibility graph was built from the tag list of Figure VI-11. The four-tag clique is shown in bold edges and shadowed nodes, and is reproduced in the table below the graph. The graph contains 13 cliques of size four, 53 cliques of size three. There are as many cliques of size two as edges, and as many cliques of size one as nodes.*

## VI.9. Scenario building

For each candidate peptide, a list of possible arrangements of tags has been constructed using the list of extracted tags and compatibility rules. Each arrangement (or clique) is a possible interpretation of the spectrum (i.e. a run-and-jump path in the spectrum graph), given a candidate peptide. Figure VI-18 shows a clique and the corresponding path in the spectrum graph, highlighting sections that are parsed in the graph, and jumps that connect two parsed sections.

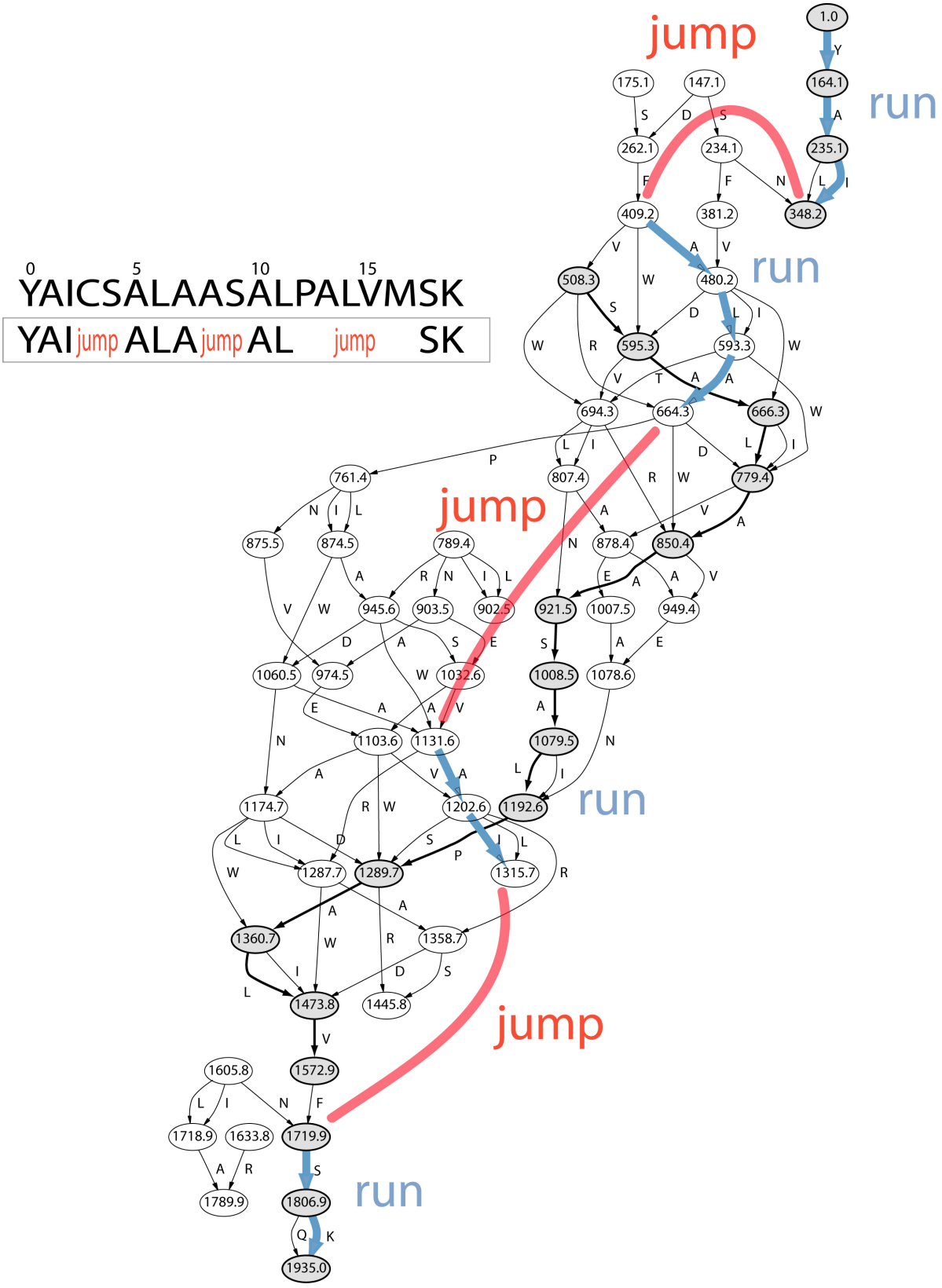



*Figure VI-18: A run-and-jump path*

*A run-and-jump path in a spectrum graph corresponding to a clique of four compatible tags. Jumps may either correspond to modification events, or to missing peaks in the MS/MS spectrum.*

The task of scenario building is to compute a shift value associated with each gap. If the value is very low, then the gap is considered as arising from a lack of information in the spectrum and is denoted as *lackGap*. Otherwise, the gap is considered as arising from the presence of a (or several) modified amino acid(s). In such a case, the gap is denoted as *modGap*, and the reported shift value represents the mass of the modification (or modifications).

Given two tags $T_i$ and $T_j$ delimiting a gap $G_{ij}$. The aim is to compute a shift $\delta(G_{ij})$ associated to $G_{ij}$.
$T_i$ and $T_j$ have several attributes:

a start position bP on the candidate peptide sequence,
an end position eP on the candidate peptide sequence, with $eP(T_i) < bP(T_j)$;
a sequence $S = \{a_1, a_2, ..., a_{|S|}\}$
a list of parsed nodes $nL = \{v_1, v_2, ..., v_{|nL|}\}$, with $meanPRM(v_u) < meanPRM(v_w)$ for all $u < w$;
a begin mass $bM = meanPRM(v_1)$
an end mass $eM = meanPRM(v_L)$, with $eM(T_i) < bM(T_j)$.

Also, each node $v_u$ contains the list of included PRMs, as well as their associated ionic hypotheses and peaks.

A first method to evaluate the mass shift $\delta(G_{ij})$ associated to the gap consists in:

1) computing $\mu_{obs}(G_{ij})$, the observed mass of the gap, using the flanking masses of the two tags

$$\mu_{obs}(G_{ij}) = bM(T_j) - eM(T_i)$$

2) computing $\mu_{exp}(G_{ij})$, the expected mass of the gap using the tag position and the candidate sequence

$$\mu_{exp}(G_{ij}) = \sum_{k=eP(T_i)+1}^{bP(T_j)-1} \mu(a_k)$$

3) and subtracting the latter from the former

$$\delta(G_{ij}) = \mu_{obs}(G_{ij}) - \mu_{exp}(G_{ij})$$

This method has a drawback, because inaccuracy on $\delta(G_{ij})$ relies solely on two nodes. Instead, we would like to share the error on all nodes included in the tags (in other words, we want to reduce the variance by increasing the sample size).

A second method, which is described below, handles this issue. It consists in:

1) listing the observed PRMs contained in tag $T_j$ (Algorithm VI-3)

$PRM\_L_{obs} \leftarrow \{\}$;
For each node $v_u \in T_j$ {
  For each $PRM_m \in v_u$ {
    $PRM\_L_{obs} = PRM\_L_{obs} \cup PRM_m$;
  }
}

*Algorithm VI-3: Building PRM_L$_{obs}$*

2) listing the expected PRMs of tag $T_j$ using the candidate peptide sequence, the tag position and previous mass shifts (Algorithm VI-4)

$PRM\_L_{exp} \leftarrow \{\}$;
For $p=bP(T_j)$ to $eP(T_j)$ {

$$PRM_{exp} = \mu(H)+\sum_{k=1}^{p}\mu(a_k)+\sum_{g=1}^{gapNb}\delta_g ;$$

$$PRM\_L_{exp} = PRM\_L_{exp} \cup PRM_{exp} ;$$

}

*Algorithm VI-4: Building PRM_L$_{exp}$*
*This algorithm lists expected PRMs of a tag $T_j$. Expected PRMs are computed by adding masses of amino acids that compose the sequence of the tag and previous mass shifts. GapNb is the number of gaps in the scenario, and $\delta_g$ is the shift value associated with gap g.*

3) Matching elements from PRM_L$_{obs}$ with elements from PRM_L$_{exp}$;

4) Reporting $\delta(G_{ij})$, the mean of the differences computed by subtracting each element of PRM_L$_{exp}$ from their matched element in PRM_L$_{obs}$

Using this second approach, Popitam reports for each gap a shift value δ. By definition, if δ<FRAGMENT_ERROR2, the shift is imputed to internal error measurements and the gap is reported as missing information in the spectrum and is signified by '-' letters in the scenario (*lackGap*). Otherwise, the gap is reported as a modification event and signified by '*' letters (*modGap*). Figure VI-19 shows examples of scenarios obtained with the example-run.

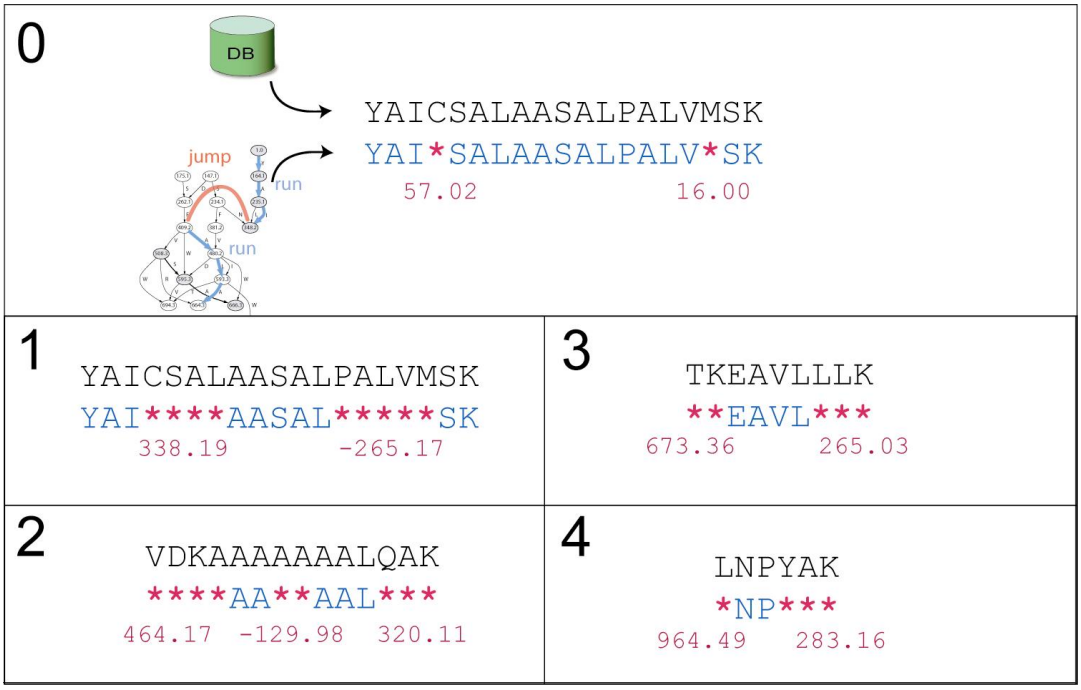



*Figure VI-19: Scenarios*

*The scenarios were obtained during the example-run performed with Popitam. 84 candidate peptides were presented to the spectrum graph, leading to a total of 1463 collected scenarios. Four of them are presented in this figure (the correct peptide is represented twice). For each scenario, the candidate peptide sequence is given in black letters. The run-and-jump paths are indicated below the peptide sequences. Red stars represent gaps associated to modification events (modGaps) that correspond to jumps in the spectrum graph. For each modGap, a shift value is reported. In scenario 0, the correct sequence was presented, and the correct path was followed in the graph. The first shift corresponds to a carbamidomethylation, and the second one corresponds to an oxidation. Scenario 1 shows a different –and false- spectrum interpretation for the same peptide. Scenarios 2, 3 and 4 correspond to interpretation attempts for incorrect peptides. Note that the number of modGaps is not always correlated to the number of modifications. Inversely, one modGap does not systematically correspond to only one modification event.*

As currently implemented, Popitam asks for the number of *modGaps* allowed in a scenario to be entered as argument (0, 1 or 2). During a run, Popitam will only consider scenarios with a corresponding number of *modGaps* (the other ones are discarded).

Moreover, three parameters are used to reduce the number of scenarios to be scored by Popitam. A first one, MIN_ARR_COV, is the minimal coverage of the candidate peptide sequence by simple edges. The reason of this filtering is to avoid evaluating scenarios that are likely to give a poor score. Two other parameters, MAX_ADD_MOD and MAX_LOSS_MOD, are set by the user and correspond to the maximum mass gain and loss allowed for a *modGap* (respectively).

## VI.10. Scenario scoring

The previous section explains how Popitam builds possible interpretation scenarios for each candidate peptide. Figure VI-19 shows five of them, out of a total of 1463. The challenge for Popitam is to succeed in spotting the best scenario (the one that corresponds to the correct peptide and uses the correct path in the spectrum) among all obtained ones. This is achieved by attributing to each scenario a score that measures the quality of the spectrum interpretation given the candidate peptide. The peptide with the highest-scoring scenario is then reported as the identification result.
The scoring function plays an essential role in the efficiency of an MS/MS identification method. In Popitam, the scenario's scoring must be all the more efficient as Popitam authorizes any type of modifications during the search, thus greatly increasing the search space. As we wanted to capture a maximum of information for the scoring procedure, we defined a set of twelve basic subscores. Each of them is based on a particular aspect, such as the coverage of the candidate peptides by the tags, the pertinence of the ionic hypotheses used in the nodes participating to a scenario, and so on.

The subscores have been defined empirically and deserve to be improved in the future. Several methods (Colinge et al. 2003;Elias et al. 2004;Tanner et al. 2005) use log-odd ratios between a null hypothesis (the match is random) and an alternative hypothesis (the match is correct).

### VI.10.1. Scores based on sequence coverage

Five scores are based on the coverage of the candidate sequence by the scenario. Here follows a short description for each of them, and a concrete computation example (Figure VI-20).

lackScore (KS1) corresponds to the number of gaps due to missing information in the spectrum.

modScore (MS1) corresponds to the number of gaps due to modification events.

covScore1 (CS1) is computed by determining $cov^{sE}$, the number of amino acids in the candidate sequence that are covered in the scenario by simple edges. This number is divided by the candidate sequence length.

covScore2 (CS2) is computed by determining $cov^{sE+dE}$, the number of amino acids in the candidate sequence that are covered in the scenario by either simple or double edges. This number is divided by the candidate sequence length.

covScore3 (CS3) is computed by determining $cov^{sE+dE+'-'}$, the number of amino acids in the candidate sequence which are covered in the scenario either by simple and double edges, or by gaps due to missing fragments in the MS/MS spectrum. This number is divided by the candidate sequence length.

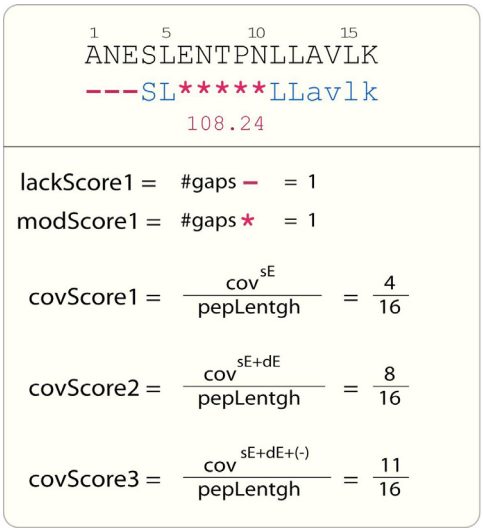



*Figure VI-20: Score computations*
*Examples of computation of the first five scores for a given scenario. "sE" means "simple edges" and corresponds to the number of upper case letters in the scenario, while "dE" means "double edges" and corresponds to the number of lower case letters.*

### VI.10.2. Score based on node pertinence

perScore (PS1) is based on the PRM's confidence scores of nodes parsed by the scenario (Algorithm VI-5). Its computation implies three loops, one for the tags included in the scenario, one for the nodes included in each tag, and one for the PRMs included in each node. The score corresponds to the sum of the PRMs confidence scores $\sigma(PRM_m)$ (the latter are defined in Section VI.5.1).

```
perScore = 0;
For each tag Ti in scenario S {
  For each node vu ∈ Ti {
    For each PRMm ∈ vu {
      perScore = perScore+σ(PRMm)
    }
  }
}
```

*Algorithm VI-5: Computation of perScore*

### VI.10.3. Score based on peak intensity

intScore (IS1) is a score based on the intensity of the peaks included in the parsed nodes (Algorithm VI-6). It represents the percentage of total intensity covered by the scenario. The score is the sum of

the intensities $\iota(s_m)$ obtained from the peaks included in the scenario, divided by the total sum of intensities in the spectrum (this number is pre-calculated once for each spectrum).

```
intScore = 0;
For each tag Ti in scenario S {
   For each node vu ∈ Ti {
      For each peak sm ∈ vu {
         intScore = intScore+ ι(sm)
      }
   }
}
intScore = intScore / Σ_{i=0}^{peakNb} ι(si)
```

*Algorithm VI-6: Computation of intScore*

### VI.10.4. Score based on PRM clustering

Scenarios that include nodes with several PRMs are more likely to be correct than scenarios including "orphans" nodes. Similar PRMs issued from different ionic hypotheses mean that a fragmentation position was represented in the spectrum by several peaks, thus confirming that the peaks corresponded to true fragments.
famScore (FS1) is simply the number of PRMs included in the scenario (Algorithm VI-7).

```
famScore = 0;
For each tag Ti in scenario S {
   For each node vu ∈ Ti {
      For each PRMm ∈ vu {
         famScore ++;
      }
   }
}
```

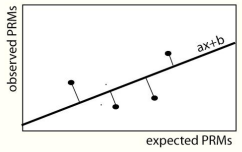
*Algorithm VI-7: Computation of famScore*

### VI.10.5. Score based on errors

errScore (ES1) is based on errors reported between observed and expected PRMs (Algorithm VI-8). Experimental measures reported by mass spectrometers are subject to errors due to calibration and internal error of the device. A good fitting between the observed PRMs included in a scenario and expected ones computed from the candidate sequence plays in favor of correctness. To compute the score, we first build pairs of observed and expected PRMs using Algorithm VI-3 and Algorithm VI-4.

Then we compute a linear regression of these values. The score reported is then based on the deviation of the values from the linear regression.

1. $PRM\_L_{exp} \leftarrow buildPRM\_L_{exp}();$
2. $PRM\_L_{obs} \leftarrow buildPRM\_L_{obs}();$
3. $PAIR\_L \leftarrow buildPairs(PRM\_L_{obs}, PRM\_L_{exp}, delta);$
4. $REG \leftarrow computeRegressionLine(PAIR\_L);$
5. For each $pair_i$ in PAIR_L {
   $errScore \mathrel{+}= dev(pair_i, REG);$
   }
5. $errScore = \sqrt[3]{errScore}$




*Algorithm VI-8: Computation of errScore.*
*$PRM\_L_{exp}$ and $PRM\_L_{obs}$ are built according to Algorithm VI-3 and Algorithm VI-4. The function buildPairs() associates each observed PRM with the closest experimental PRM, given delta, the sum of the previous shift masses.*
*REG represents the linear regression computed from the pairs of expected and observed PRMs. The function dev() returns the distance from the point defined by the pair to the regression line.*

### VI.10.6. Score based on peak redundancy

redScore (RS1) is based on the multiple presence of given peaks in the scenario (Algorithm VI-9). For example, peak 480.24 in the spectrum shown in Figure VI-2 could (given a certain error margin) represent two different fragments: an $a_4$ ion type (of mass 480.23), or an $y_4$ ion type (of mass 480.25). The peak will therefore result in two different PRMs, one representing the prefix mass ending at position 4 of the peptide, and one representing the prefix mass ending at position 15. But such a situation is unlikely to occur. Consequently, if a peak is shared by multiple nodes in the scenario, the final score should be lowered. To compute the redScore, a table (denoted as redTab in Algorithm VI-9) of size peakNb and filled with 0 is prepared. Then the peaks included in the scenario are parsed using the three loops. For each peak index $ind(s_m)$, the corresponding case of redTab is incremented. At the end, the table is parsed to compute the redScore, which is finally divided by the number of different peaks used in the scenario.

```
redScore = 1;
usedPeak = 1;
redTab[]  = {0, 0, ..., 0};
For each tag Ti  in scenario S {
   For each node vu ∈ Ti  {
      For each peak sm ∈ vu {
         redTab[ind(sm)] ++;
      }
   }
}
For k = 0 to peakNb {
   if (redTab[k] !=0) usedPeak++;
   if (redTab[k] > 1) redScore = redScore + redTab[k]^2;
}
redScore = redScore ;
           usedPeak
```

*Algorithm VI-9: Computation of redScore.*

### VI.10.7. Score based on peak series

Peptide fragmentation tends to produce series of peaks of similar ionic types, coming from successive cleavage positions on the peptide. Tags that include peak series are more likely to be correct than tags formed from isolated peaks of different ionic types.
serScore1 (SS1) is the length of the longest series of b-ions observed in the scenario,
serScore2 (SS2) is the length of the longest series of y-ions observed in the scenario.

### VI.10.8. Scenario scoring

Combining all the subscores into an efficient scoring function is a challenging task. We therefore first tested the subscores independently, and then tried to define empirical scoring functions as the one represented below.

$$S = \frac{CS1 + CS2 + CS3}{3} \cdot \frac{1}{ES1 \cdot RS1} \cdot IS1 \cdot PS1 \cdot FS1 \cdot SS1 \cdot SS2$$

*Equation VI-5: Empirical scoring function*

We also used Genetic Programming to explore the space of the possible combinations of subscores using supervised learning. Chapter VII presents in detail the GP methodology. Results obtained with the various scoring functions are presented in Chapter VIII.

ぼくの恋人は、ぼくに哲学がないと言う。
ぼくは気持ちさえあれば、哲学なんて要らないと思う。

Ma petite amie dit que je n'ai pas de philosophie.
Moi, je pense qu'on n'a pas besoin d'en avoir,
si on a du coeur.

My girlfriend says that I have no philosophy.
If one has some heart,
I think philosophy is not needed.

Figure adapted from "Monsieur Hippopotame" by Tanikawa Shuntarô

# CHAPTER VII

# GENETIC PROGRAMMING

This chapter deals with Genetic Programming, a method that belongs to the class of Evolutionary Algorithms. These algorithms are stochastic global optimization methods and are freely inspired from the Darwinian evolution theory. We used Genetic Programming to tailor scenario-scoring functions specifically adapted for Popitam.

# VII. Genetic Programming

## VII.1. Introduction to Evolution theory

Evolution is not a modern discovery. Its inception dates back to a long time. People agree that one of the first proponents of a concept of evolution is Anaximander of Miletus (611-547 B.C.), a pupil of Thales. Anaximander believed that life arose from warm water and earth. According to Anaximander, human beings were not present at the earliest stages because human offspring requires a long period of nursing to survive, and then the first human generation would inevitably have died. For Anaximander, the first forms of live were fish-like creatures, and human beings grew inside them. When they became able to feed and take care of themselves, they left the creatures and came forth onto dry land. Antiquity's thinking was dominated by other theories. Aristotle (384-322 B.C.) proposed a "finalist"[4] vision of nature in which chance had no place. Species were "fixed", each of them representing a level from the simplest forms of life to the most complex ones. Organs were fabricated with a precise aim. Thus, according to Aristotle, roots exist because plants need to take water from the ground and human have teeth because they need to be fed with solid aliments. Like Aristotle, Lucretius (99-55 B.C.) did not believe in the production of new species from previously existing ones. But he anticipated the natural selection theory when he proposed that species were formed by the combination by chance of elements and that once living "monstrous" organisms are now extinct while others survive because of their strength, speed or dexterity.

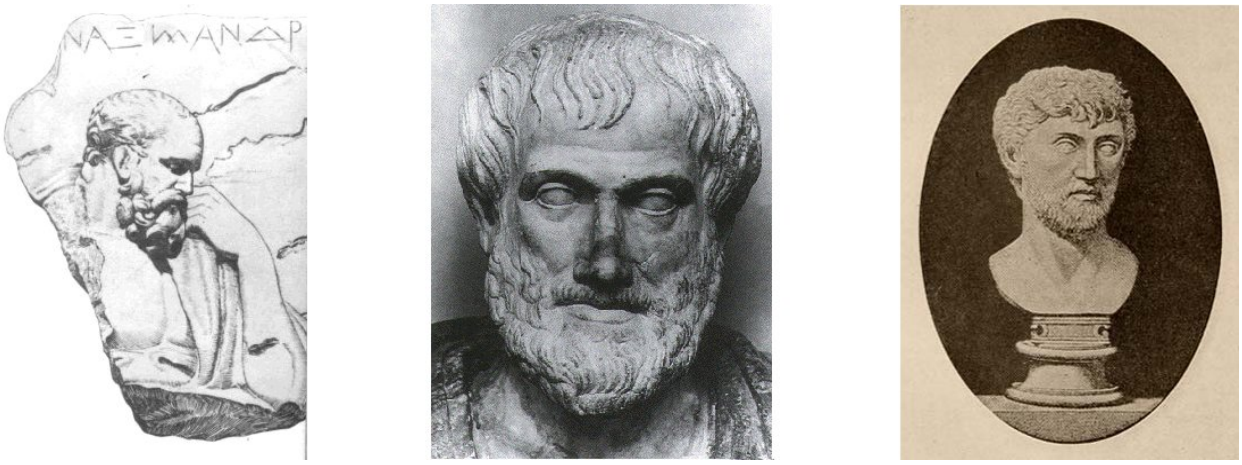


*Figure VII-1: Three Greek philosophers*
*Anaximander (611-547 B.C.) Aristote (384-322 B.C.) and Lucrece (~98-55 B.C.)*

[4] Finalism" is a philosophical term related to a belief in ultimate purpose or design behind everything, including, the evolution of the cosmos and of life

In the 18th century, an astronomer and mathematician, Pierre de Maupertuis (1698-1759), refuted the idea of a purposeful design and suggested that new species arise from successive mistakes occurring randomly during body formation. He wrote in 1750 in his «Essai de cosmologie»:

*"Le hasard, dirait-on, avait produit une multitude innombrable d'individus; un petit nombre se trouvait construit de manière que les parties de l'animal pouvaient satisfaire à ses besoins; dans un autre infiniment plus grand, il n'y avait ni convenance, ni ordre: tous ces derniers ont péri; des animaux sans bouche ne pouvaient pas vivre, d'autres qui manquaient d'organes pour la génération ne pouvaient se perpétuer... les espèces que nous voyons aujourd'hui ne sont que la plus petite partie de ce qu'un destin aveugle avait produit..."*

*"Chance apparently turned out a vast number of individuals; a small proportion of these were organized in such a manner that the animals organs could satisfy their needs. A much greater number showed neither adaptation nor order; These last have all perished -- thus the species that we see today are but a small part of all those that a blind destiny produced."*
*(adapted from http://www.mlahanas.de/Greeks/Evolution.htm)*

Eventually, a new conception of the living world, known as the Transformism, arose. Fossils became recognized as concrete evidence that organisms change over time, and scientists realized that organisms once lived are now extinct, while new species seem to derive from common ancestors. The changes are gradual and slow, because no one ever observed the sudden appearance of new species. But the mechanisms of such changes had yet to be described and demonstrated. Jean-Baptiste Lamarck (1744-1829) drew an evolution theory in which organisms transmit to their descendants characters that are acquired by interacting with the environment.

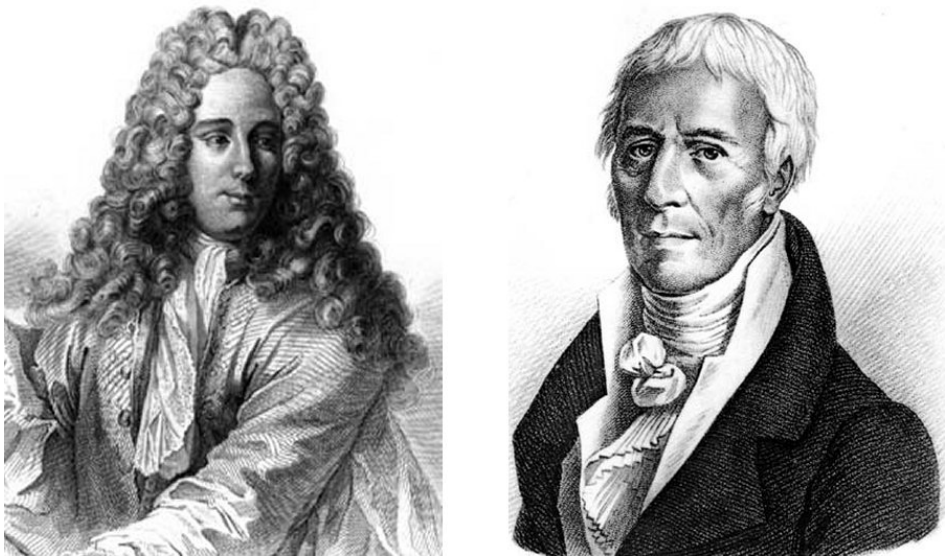


*Figure VII-2: The astronomer Pierre de Maupertuis (1698-1759) and the naturalist Jean-Baptiste Larmarck (1744-1829)*

The acknowledged founder of the modern evolutionism is the naturalist Charles Darwin (1809-1882), who elaborated a theory from observations made during his five-year trip around the world on the ship "Beagle". Darwin wondered about the geographic distribution of species, notably on the Galapagos Islands, and about their similarities and differences. Based on many concrete observations, on proof accumulation and on a strong argumentation, as well as on a great knowledge of contemporaneous works, Darwin formalized a complete theory of evolution, supported by many specific examples. In his book "*On the origin of species by means of natural selection or the preservation of favored races in the struggle for life*", published in 1859, he suggested that the mechanism of evolution is natural selection. Based on the observation that individuals in a population are not all similar and that variations can be inherited, his theory stated that the variations result in differential survival and reproductive success rates, which leads to shifts in the frequency of characters. Darwin's theory of evolution postulated that even very small variations can result in differential reproductive success. By accumulating small changes over a large number of generations, the individuals can no longer interbreed and new species are formed.

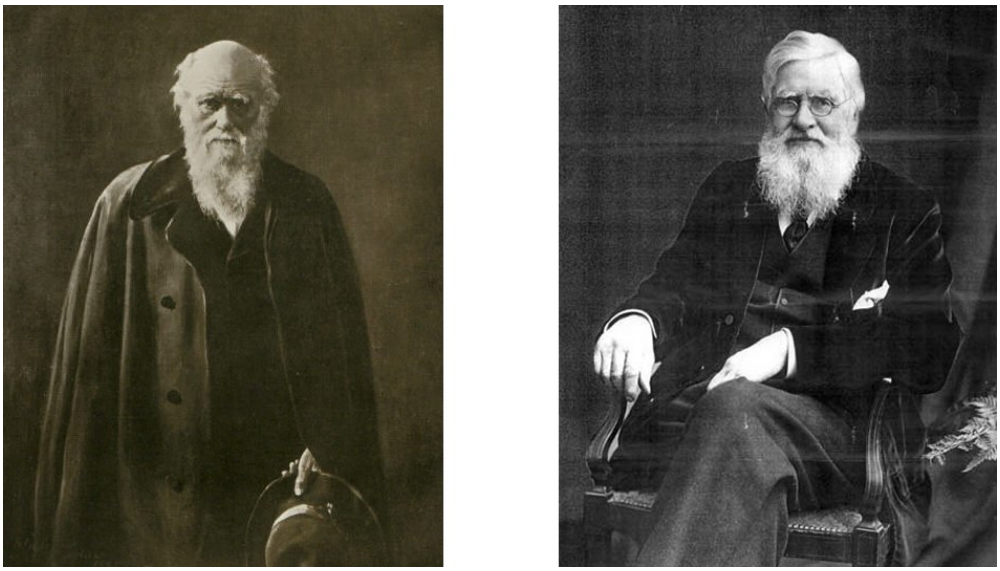


*Figure VII-3: The naturalists Charles Darwin (1809-1882) and Alfred Wallace (1823-1913)*

This chapter is about evolution, natural selection and survival of the fittest. As we will see in the next sections, these principles can be applied to solve difficult optimization problems. Examples of Evolutionary Algorithms are Genetic Algorithms (GA) and Genetic Programming (GP). We were interested in using Genetic Programming to generate optimal functions for scoring Popitam's scenarios.

## VII.2. Genetic Programming

### VII.2.1. Introduction

Optimization problems are characterized by a set $\Omega$ of possible solutions –the search space- and an objective function $f: \Omega \rightarrow \Re$ that associates a value to each possible solution x. The aim is to find the solution(s) $x^* \in \Omega$ that maximizes or minimizes the objective function. When the size of $\Omega$ is reasonable, an exhaustive approach can be used. Such an approach consists in successively enumerating all possible solutions $x \in \Omega$ and keeping the best one(s). But optimization problems often have such a huge search space that they cannot be solved in a reasonable computing time or memory space. In the most difficult cases, where no deterministic approaches can discover the optimal solution in reasonable time, non-deterministic approaches have to be used. These possess the advantage to give good sub-optimal solutions in a rather short time.

Genetic programming, which was introduced by Koza in 1992 (Koza 1992), is a non deterministic approach that is inspired from the evolution of species. Evolution theory is an attractive model because it can be considered as an optimization method that causes organisms to bet better adapted in a changing environment. Key ingredients for an evolution process to take place are: a) overproduction (more offsprings are produced than will ultimately survive and reproduce, generating a struggle for existence); b) inheritance (characteristics are transmitted to the descendants) and c) variation (inheritable features vary from individual to individual). Genetic programming (GP) models these key ingredients and applies them on a population of solutions to a given problem. The solutions are created from a set of available functions (mathematical operators, statements, routines...) and variables. The search space to explore is composed of all possible combinations (with repetition) of the functions and variables. Using evolution principles –differential reproduction, variation and character inheritance -, GP makes the population of solutions evolve. With generations, solutions become more and more specialized for the problem (adaptation to the environment). A typical GP workflow (Figure VII-4) starts with the generation of a population of random solutions. Each solution is evaluated using an *objective function* and receives a *fitness value*. The objective function is dependent of the problem under consideration. It gives a measure on how well (or how badly) a solution succeeds in solving the problem. Solutions that will reproduce are selected according to their fitness, and variation is introduced by applying *genetic operators*. The aim of this procedure is to sample new points of the search space. The genetic operators are generally inspired from genetic events, such as the crossing-over, source of chromosomal exchange during meiosis, or mutational events. The crossing-over allows solutions to exchange parts of themselves. Intuitively, if two solutions are good performer on a problem, then subparts of these solutions are susceptible to form, by random recombinations, solutions even more performing than the parent solutions. A new population (called daughter population) is formed from the modified solutions and replaces the parent population for the next generation. After several generations, the solutions become more and more adapted and the average fitness of the population increases. The fittest solution(s) over all generations is (are) designated as the result of the GP process.

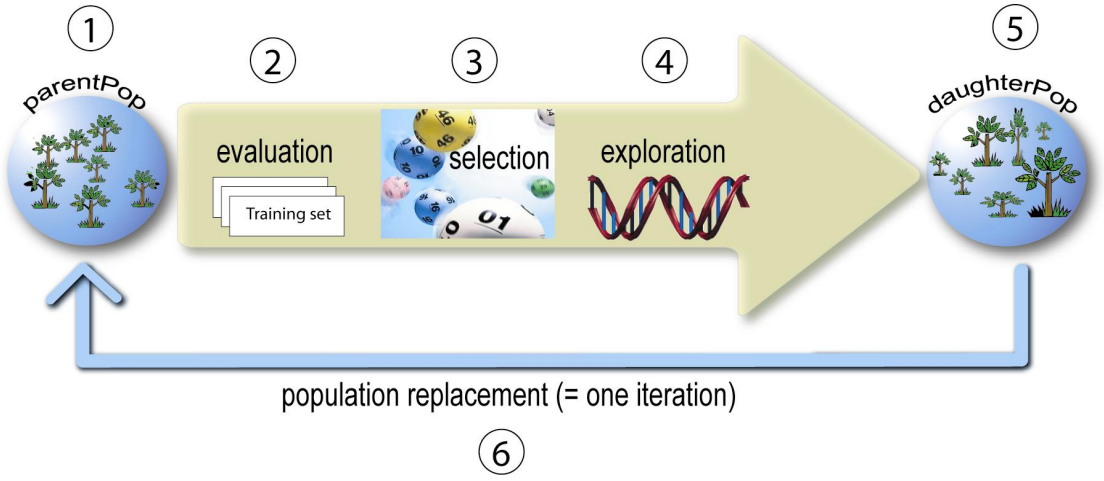



*Figure VII-4: Genetic programming workflow*
*The algorithm starts with an initial population of solutions (randomly built or using a priori knowledge) (1). Each solution is evaluated and assigned a "fitness" value that measures the adequacy of the solution to solve the problem. Typically, each solution is tested in heterogeneous conditions using a set of examples (a learning set) (2). Selection of solutions to reproduce is probabilistic and biased towards solutions with high fitness (3). Genetic operators are applied on the selected solutions and generate variants solutions (4). The latter form the "daughter" population (5). The population generation process is repeated until some criterion is met (e.g. convergence of the system, number of iterations reached or quality of the best solution) (6).*

We used a specific model of GP, called parallel multi-objective GP. The next sections describe the particularities of this model and explain the different steps of the workflow.

### VII.2.2. Parallel GP

A drawback of GP is the considerable amount of computing time that can be required for each generation, as a certain number of them must be completed before the algorithm produces a satisfying solution. However, a parallel algorithm approach can be applied to classical GP to reduce processing time by dividing the population of solutions into several subpopulations and sharing subpopulations between several processors. This technique is known as 'coarse-grained parallel GP' (Cantu 1999). Figure VII-5 charts a GP process performed by a given subpopulation in a parallel GP workflow. We describe in this section the part of the figure that concerns communication between the populations. Section VII.2.3 explains how the solutions are coded, while their evaluation, their selection and the genetic operators are described in Sections VII.2.4, VII.2.5 and VII.2.6, respectively.

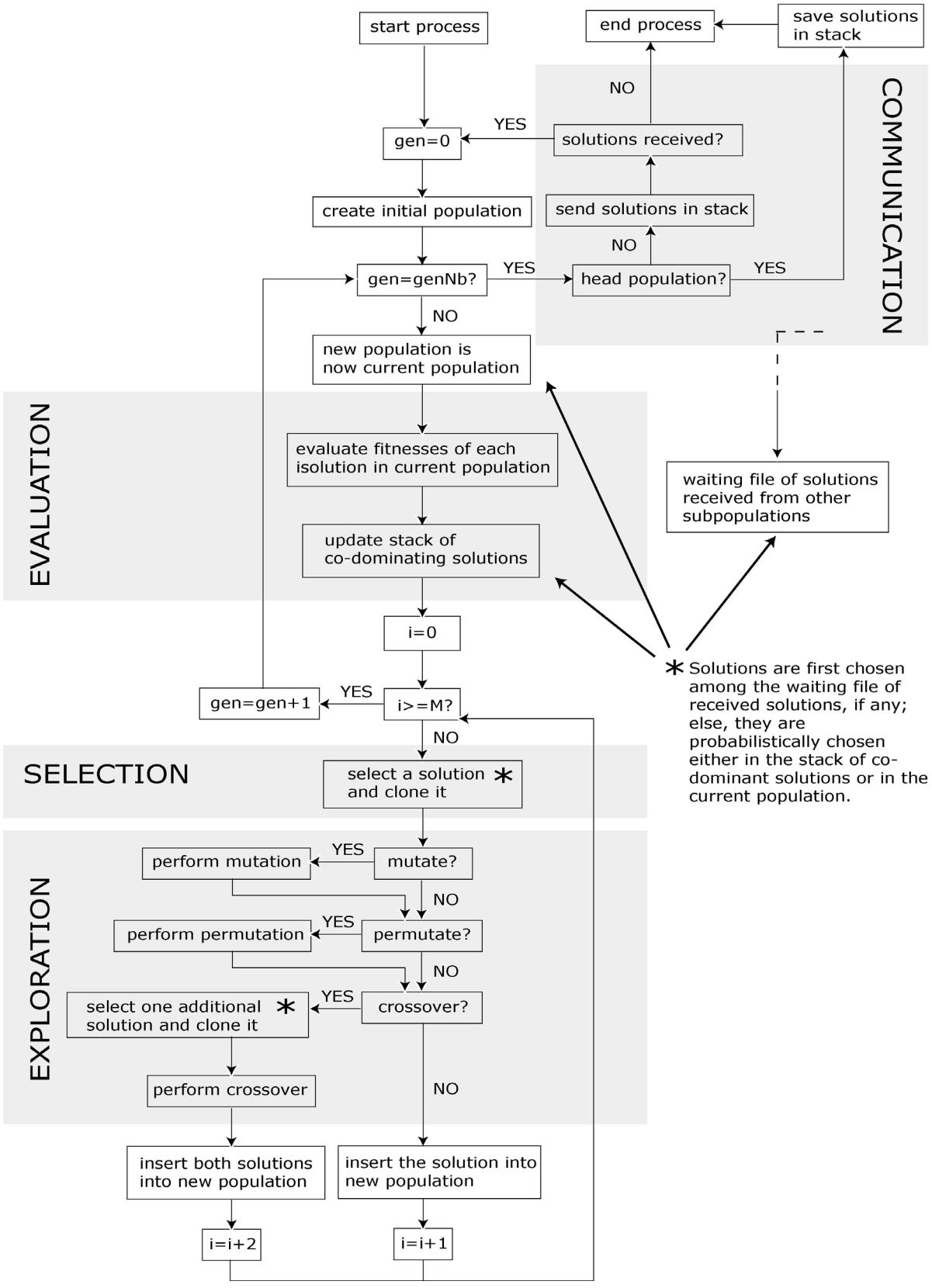



*Figure VII-5: Multi-objective parallel genetic programming (flowchart)*

*This flowchart depicts the actions performed by a subpopulation in a multi-objective parallel Genetic Programming context. GenNb represents the number of generations to produce and M is the population size. Two counters are used: gen (counts the number of generation) and i (counts the solutions).*

The principle of parallel GP is the following: each subpopulation independently runs a classical GP process with its own parameters. Subpopulations communicate with each other by sending and receiving promising solutions. We use an original topology model, called *pyramidal model* (Frey et al. 2003), in which subpopulations are distributed among several processors and organized in several superimposed layers to form a pyramid. Communication flows from the base of the pyramid to the top (the head-subpopulation) (see Figure VII-6).

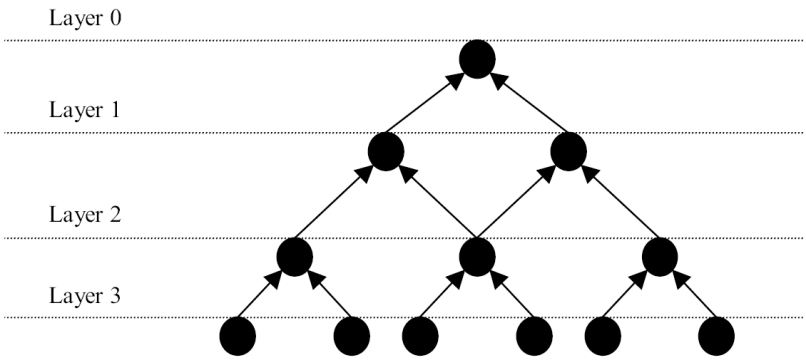



*Figure VII-6: Communication topology of a pyramidal model*
*Figure taken from (Frey et al. 2003)*

Each time a subpopulation has completed a given number of generations, it sends its stack of solutions to higher-level subpopulations (except if the subpopulation is the head-subpopulation), which returns a receipt. After receiving the receipt, the subpopulation starts a new process. If the receipt is not returned, this means that the above populations are not running anymore, and the subpopulation ends its GP process. Correspondingly, the whole GP process is not ended as long as the head-population has not produced its complete number of generations.

The parameters in each subpopulation are adapted according to the depth of their layer. Parameters of low-level subpopulations are set such as to intensively explore the search space of possible solutions. The promising solutions are sent to higher-level subpopulations whose parameters are set such as to exploit this information.

### VII.2.3. Coding of the solutions

In GP, solutions are coded as trees. This hierarchical and evolutional structure is well suited to represent a series of instructions (see Figure VII-7). The trees are composed of nodes, taken from a set of predefined *functions*, and of leaves, taken from a set of predefined *terminals*.

*Functions* are usually of the following types:
a) mathematical operators (+, -, *, /, ^,...)
b) mathematical functions (sin, cos, exp, log,...)

c) boolean operators (and, or, not,...)
d) conditional operators (if then else, case, switch, ...)
e) loop statements (while...do, repeat...until, for...do)
f) any subroutine (any function defined for the problem under consideration)

and *terminals* are of the following types:
a) integers
b) reals
c) vectors
d) any predefined structure

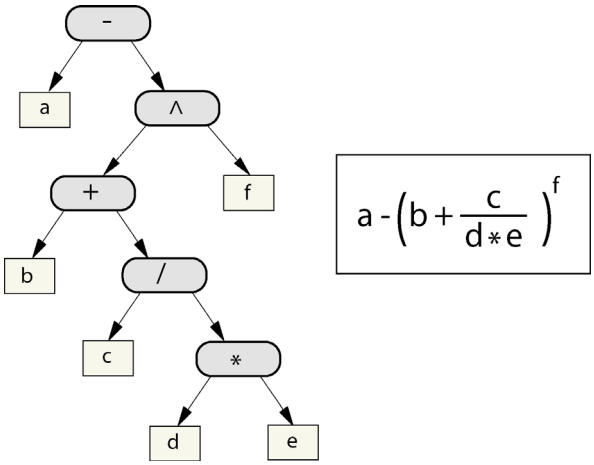



*Figure VII-7: A solution coded as a tree*

### VII.2.4. Evaluation of a solution

To emulate evolution, GP algorithms must determine which solutions best solve the problem under consideration. Evaluation of a solution is performed through the computation of a fitness value by an *objective function.* The higher the fitness of the solution, the higher its chance of being selected and of transmitting information to the next generation (principle of survival of the fittest).

In certain cases, one would like to optimize a solution according to several criteria taken simultaneously. For example, a first objective (to maximize) would be the capacity of the solutions to solve the problem, and a second objective (to minimize) would be the size of the solutions. The aim of such a multi-objective optimization scheme is to avoid the bloating of solutions, which tend to become larger and larger as generations pass. In multi-objective optimization, the solutions receive as many fitness values as the number of criteria to optimize, and each of these fitnesses is measured by a specific objective function.

If a unique measure regrouping these fitnesses cannot be defined, Pareto optimality (Miettinen 1999) can be used to define the best solution. Given $F^*=\{F_1, F_2, ..., F_m\}$ the set of m objective functions to be simultaneously optimized, a solution $x_p$ is said Pareto optimal if there is no other possible solution x for which:

$$\exists\ F_i \in F^* \,|\, F_i(x) > F_i(x_p) \text{ and } \neg\exists\ F_j \in F^* \,|\, F_j(x) < F_j(x_p)$$

In words, a solution $x_p$ is said Pareto optimal if no solution x has one of its fitnesses higher than the corresponding fitness of $x_p$ and if every possible solution x has at least one of its fitnesses smaller than the corresponding fitness of $x_p$.

The Pareto dominance states that a solution $x_1$ dominates a solution $x_2$ if:

$$\forall\ F_i \in F^*,\ F_i(x_2) \leq F_i(x_1) \text{ and } \exists\ F_i \in F^* \,|\, F_i(x_2) < F_i(x_1)$$

In words, a solution $x_1$ dominates a solution $x_2$ if all the fitnesses of $x_2$ are smaller or equal than the corresponding fitnesses of $x_1$ and if at least one fitness of $x_1$ is better than the corresponding fitness of $x_2$ (the solutions have not equal fitnesses).

The sampling of the search space of possible solutions by a GP process leads to the collection of a number of co-dominating solutions forming a *Pareto optimal set*. None of the solutions in the Pareto optimal set dominates another solution of the set. In parallel GP, each subpopulation maintains its own stack of co-dominating solutions. Each time a solution is evaluated during the GP process, the stack is updated according to the two following rules:
a) if the new solution is not dominated by any of the stack solutions, it is integrated in the stack.
b) if the new solution dominates any of the stack solution, the stack solution is deleted.
When a subpopulation completes its number of generations, it sends the solutions contained in the stack to the subpopulations of the above layer and starts a new process.
When the head-population has completed its assigned number of generations, its stack of co-dominating solutions is saved in a text file that represents the result of the whole GP process.

### VII.2.5. Selection

At each iteration, solutions are chosen to form the next generation. In multi-objective parallel GP, there are three sources from which solutions can be chosen: the current population, the stack of co-dominating solutions, and the solutions that have been received from other subpopulations and are stored in a waiting list. First, the solutions in the waiting list are selected until the list is emptied (the maximal number of solutions in a waiting list is set by a parameter). Then, the algorithm picks solutions either from the current population or from the stack of co-dominating solutions according to a probability parameter called *p(elitism)*. If the stack is chosen, the solution is randomly selected among the stack solutions. Otherwise, the solution is selected among the current population using a rank-based procedure and a biased lottery wheel. The procedure is the following: the solutions of the current population are ranked according to the number of other solutions they dominate. Then the solution is randomly chosen using a biased lottery wheel, according to:

$$p(r) = \frac{2 \cdot PS + 2 \cdot (PS - 1) \cdot (r - 1)}{n - 1}$$

*where*

*p(r) is the probability to choose a solution with rank r*
*n is the total number of solutions*
*PS is a parameter (called Selective pressure) that controls the influence of the fitnesses on the selection. If PS is set to 1, the solutions are randomly selected (independently of their fitnesses). A PS between 0 and 1 would favor the selection towards solutions with low fitnesses, while a PS between 1 and 2 favors the selection towards solutions with high fitnesses.*

Once a solution is selected, it is "cloned" (copied) so that the genetic operators modify the clones rather than the parent solution. Consequently, a solution can be chosen several times during the selection process.

### VII.2.6. Genetic operators

Genetic operators are used to generate variability in selected solutions (=clones). The set of possible operators that can be defined is unlimited, but most often, a GP algorithm includes all or part of the following three genetic operators: the *crossing-over* operator (CO), the *mutation* operator (MUT) and the *permutation* operator (PER).
Each operator may or may not modify a solution according to a probability value. For example, if p(MUT) is set to 0.2, a given solution has 1 chance out of 5 to be mutated. Once each genetic operator has had the opportunity to modify the solution, the latter is inserted into the growing daughter population and a new solution is selected from the parent population.

Here follows a short description of each of the three genetic operators (Figure VII-8, Figure VII-9, Figure VII-10). If none of the three genetic operators is applied, the function is inserted untransformed into the daughter population.

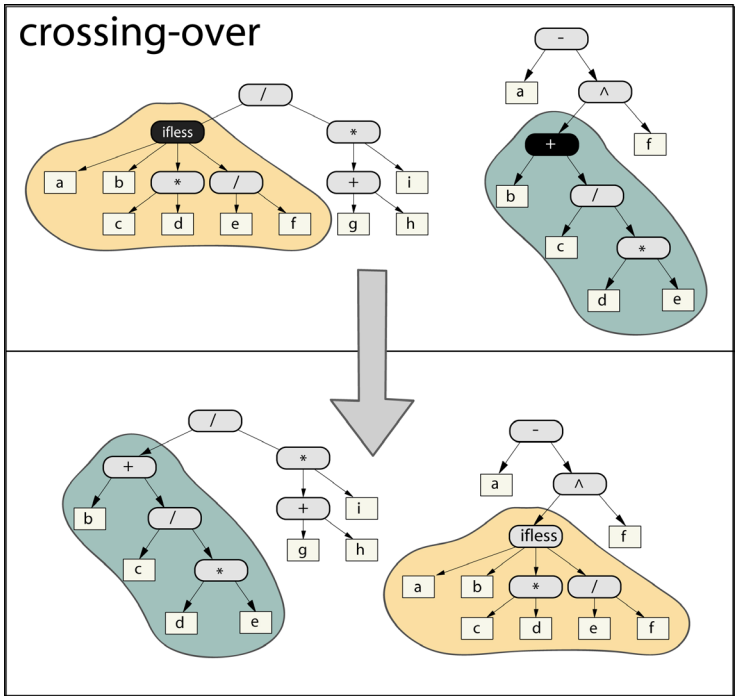



*Figure VII-8: Crossing-over operator*
*The crossing-over operator generates variation by exchanging information between two randomly selected solutions.*
*In each selected solution, a crossover point (either a node or a leaf) is randomly chosen (black nodes). The subtrees defined by the crossover points are exchanged between the two solutions, generating two "new" solutions.*

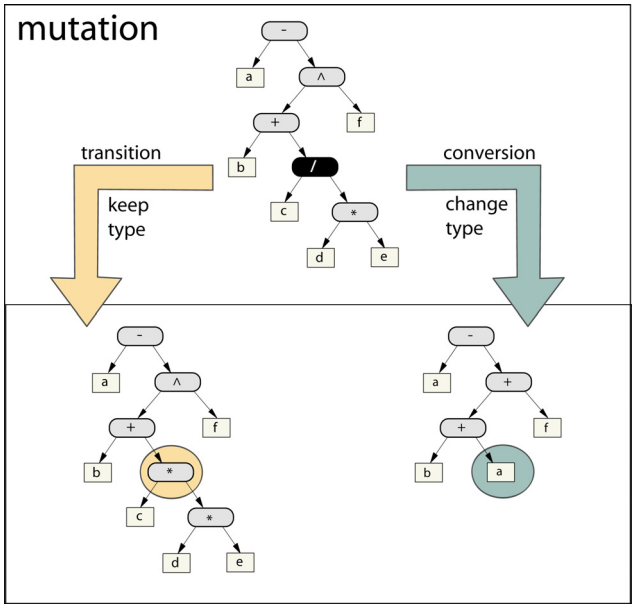



*Figure VII-9: Mutation operator*

*The mutation operator generates variation by modifying a node or a leaf of the tree. First, a mutation point is randomly defined (in black). Then the algorithm chooses between a transition event and a conversion event. The choice is probabilistic (p(transition)=p(conversion)=0.5). In case of transition, the value of the node defined by the point mutation is randomly changed, but the type (node or leaf) is kept. In case of conversion, both type and value of the node are changed, leading to a modification of the tree structure. If the conversion point was a node, the subtree is deleted and replaced by a randomly chosen leaf. If the conversion point was a leaf, the latter is deleted and replaced by a node from which a new subtree is randomly generated.*

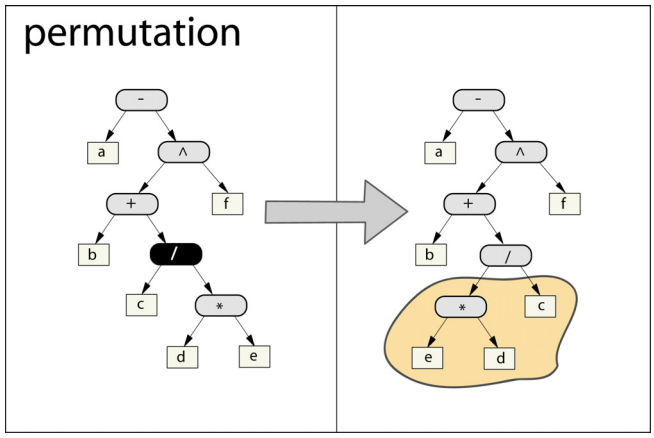



*Figure VII-10: Permutation operator*

*The permutation operator modifies the order of the arguments of a node. It starts by randomly selecting a node as a permutation point (black node). If the node has two arguments, the left-son is permuted with the right-son. If the node has four arguments (IF_LESS), their order is randomly modified among the 4!-1 permutation possibilities. The structure of the subtree defined by the permutation point is then modified accordingly.*

## VII.3. GP application for Popitam

### VII.3.1. Introduction

Genetic Programming appeared to be an attractive way to build functions for scoring scenarios produced by Popitam. The solutions are thus scenario-scoring functions built from a set of 6 different nodes, comprising the mathematical operators addition, subtraction, multiplication division, power, and conditional statement IF_LESS, and from a set of 13 different leaves, represented by the 12 scenario's subscores described in Section VI.10 and a random coefficient. This choice presents the advantage that it is possible to use equally positive or negative values, as well as integers or floats, except for the division operator that does not accept a division by 0. When such a situation arises for a tree, the latter is labeled as "inconsistent" and its associated fitnesses are set as "very bad", thus avoiding the selection of that tree for reproduction. Figure VII-11 presents a possible solution. The search space is formed by all consistent scoring functions that can be built from the set of nodes and leaves (with repetition). The aim of the GP process is to find scoring functions that not only give the highest score to the correct candidate peptide, but also efficiently discriminates the correct peptide from all the other ones.

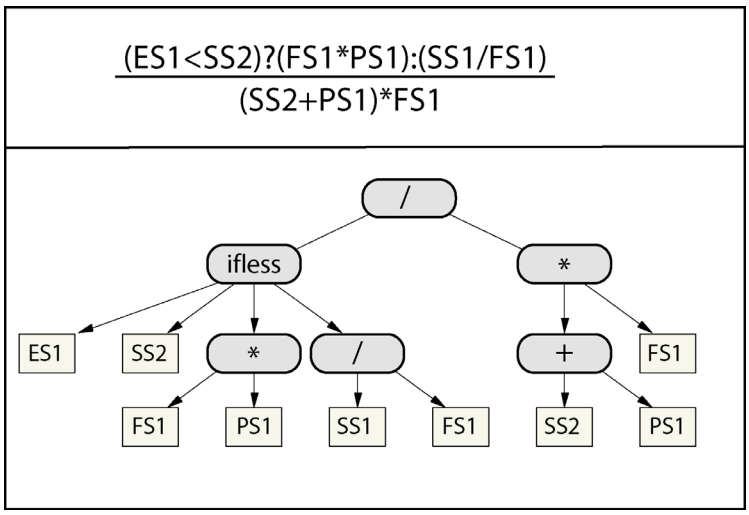



*Figure VII-11: A scenario-scoring function written as a tree structure*
*The conditional operator "ifless" is read as follows: if ES1 is less than SS2, then return FS1*PS1 ; else return SS1/FS1 ; solutions in a population are of various sizes and forms.*

### VII.3.2. Evaluation of solutions

The solutions produced during the GP process are evaluated using a set of "examples" (called *sample set*). The sample set is built from a set of MS/MS spectra with known identifications (called *learning set*) (see Section VII.4). The evaluation procedure, which is depicted in Figure VII-12, is the following: before launching the GP algorithm, Popitam is run on the spectra of the learning set. For each candidate peptide, a list of possible scenarios is built by Popitam and for each of them, the 12 subscores (CS1, CS2, ... SS2) are computed and saved in a text file. One sample file is created per original spectrum. The learning itself is thus executed on the text files composing the sample set, thus

avoiding unnecessary Popitam runs for each solution to be evaluated and for each generation (the subscores are not dependent on the GP process, hence the interest to compute and save them once only). During the GP process, the solutions are evaluated by "reading" the subscores in the files and computing the scenario's scores. The identification score of a candidate peptide is given by its best scoring scenario. Thus, for each sample, a list of scored peptides is obtained. One among the peptides is the correct one (posPep), all the others are negative peptides (negPep). The solution fitnesses express how well the solution was able to give a good rank to the correct peptide (*rankFitness*) and how well it can separate the correct peptide score from the distribution of negative peptide scores (*discFitness*).

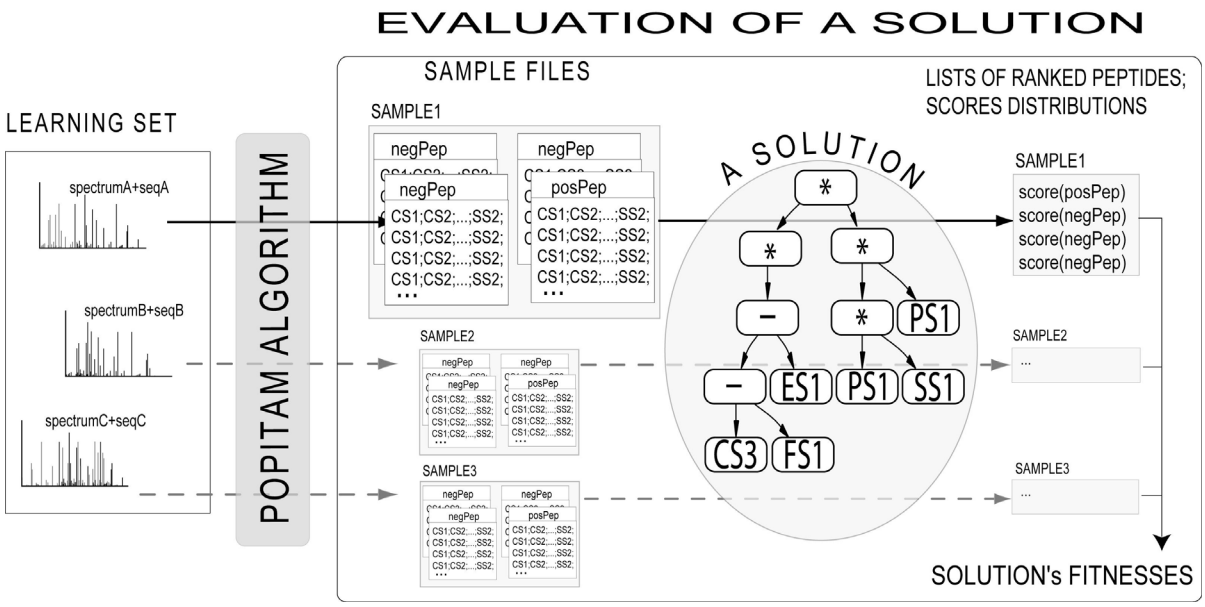



*Figure VII-12: Evaluation of a function*
*Evaluation is based on examples (sample files) prepared with Popitam. Each sample file contains lists of subscores computed by Popitam for candidate peptides (the correct one, denoted as positive peptide, or posPep, and the negative ones). All the other subscores were obtained for negative peptides. The solution is used to compute a global identification score for each positive and negative peptide. The solution's fitnesses measure how well the solution can discriminate the score of the positive peptide with the scores of the negative ones.*

The next Sections describe the objective functions that are used to compute the fitnesses in our GP algorithm.

#### VII.3.2.1. rankFitness

The *rankFitness* measures the capacity of the solution to give the highest score to the correct peptide. It is based on the rank of the correct candidate peptides. For each sample k of the learning set, the candidate peptide scores are sorted by decreasing order and the rank of the positive candidate peptide is reported. The *rankFitness* is computed by adding a contribution from all samples, according to:

$$\text{rankFitness}(S_i) = \sum_{k=0}^{\text{sampleNb}} \frac{1}{\text{rank}(\text{posPeptide})^2}$$

*where*

*$S_i$ is a solution*
*sampleNb is the number of spectra in the learning set, and*
*rank(posPeptide) is the rank of the correct candidate peptide for the sample k.*

This fitness is comprised between 0 and 1; its optimum is 1, which signifies that all spectra of the learning sets were correctly identified (first rank).

#### VII.3.2.2. discFitness

The *discFitness* is related to the capacity of the solution to discriminate the correct scenario from all the other ones. It is based on the value of the positive score relative to the distribution of the negative scores, and corresponds to a p-value. P-values give the probability of obtaining by chance a score greater than or equal to an observed score, in a given distribution. The function we use to compute the p-values presumes that the distribution is gaussian. As we are not guaranteed that the negative distribution is gaussian (actually, as shown in Figure VIII-8, it is not), the p-value is consequently not a "true" probability, but a measure that gives an idea of the deviation between the score of the correct peptide and the negative distribution. *DiscFitness* corresponds to the mean of p-values reported for first ranked positive scores. It is comprised between 0 and 1, and its optimum is 0.

#### VII.3.2.3. sizeFitness

The third fitness, called *sizeFitness*, is equal to the number of nodes in the solutions. It was used as a third objective to optimize to avoid bloating of the solutions.

### VII.3.3. Topology and parameters

The GP topology we used included three subpopulations placed at three levels (see Figure VII-13). We designate the first level subpopulation "head-population", and the two others "low-level populations". As shown in the figure, parameters were chosen to favor the exploration of new

solutions in the low-level populations and the exploitation of solutions in the head-population. Notably, selective pressure (PS), elitism probability (p(elitism)) and crossing-over probability (p(CO)) were increased in the head-population. This means that the head-population will tend to include more good solutions in the next generations, and will focus on combinations of these solutions. Low-level populations will focus less on solutions, but will keep exploring the search space, notably by adding diversity due to a higher probability of mutation (p(MUT)).

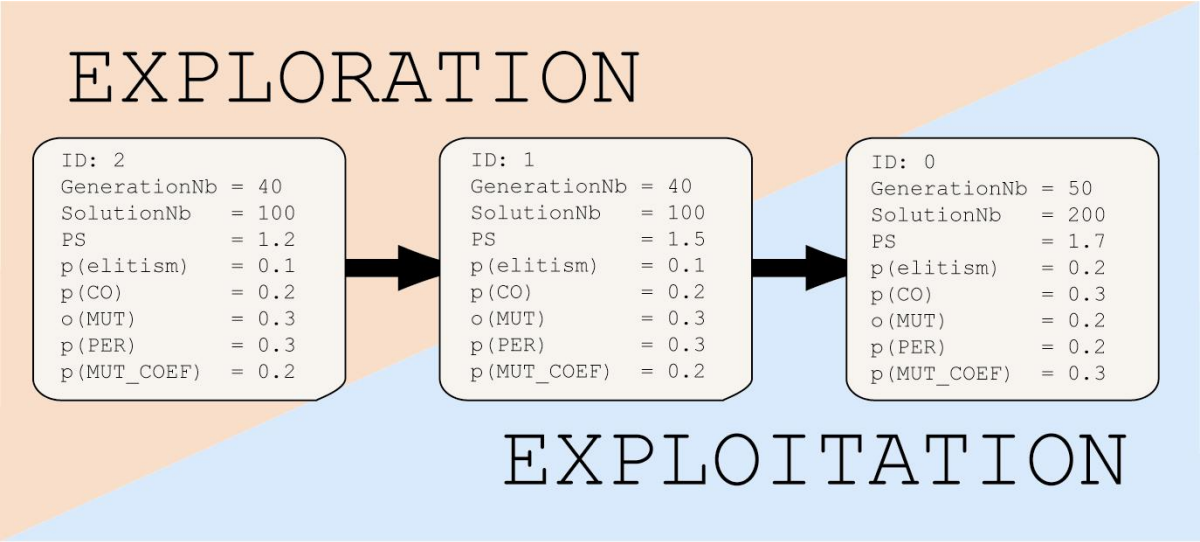



*Figure VII-13: Topology and parameters chosen for the three populations*
*The head-population is the ID 0. Communication flows from low-level populations to high-level populations (black arrows). The parameters are the selective pressure (PS), the probability of elitism (p(elitism)), of crossing-over (p(CO)), of mutation (p(MUT)), and of permutation (p(PER)). A last parameter is p(MUT_COEFF), the probability that a random coefficient is changed (random coefficient are one of the possible leaves).*

As shown in Figure VII-13, the head-population contains more solutions and fills more generations that the other ones. When one of the low-level populations has completed its generations, all solutions contained in the stack of co-dominating solutions are sent to the next level population and a new GP process begins in this low-level subpopulation.

Figure VII-14 shows the convergence of the GP algorithm towards high-fitness solutions for the three subpopulations. Data were collected during runs performed for Chapter VIII. The process starts with a bi-objective optimization (*rankFitness* is maximized and *discFitness* is minimized). During these early generations, the algorithm freely and deeply explores the search space and produces solutions that are more and more adapted and complex. When two thirds of the generations have been completed, the third objective (*sizeFitness*) is taken into account. From this point on, the selection process favors small solutions, even if they have poor *rank*- and *discFitnesses*. As a result, a decrease of the mean *rankFitness* and mean *discFitness* of the population is observed. Nevertheless, as generations progress, new solutions continue to be created, which have a small size and better and better *rank*- and *discFitnesses*. At the end of its GP process, the head-population stack of co-dominating solutions contains a wide range of different kinds of scenario-scoring functions, from the

most complex ones, comprising about one hundred nodes, to the most parsimonious ones, comprising about ten or even only one or two nodes. By this mean, we had the possibility to compare the performance of very complex learned functions with more simple ones (see Section VIII.2.3.3).

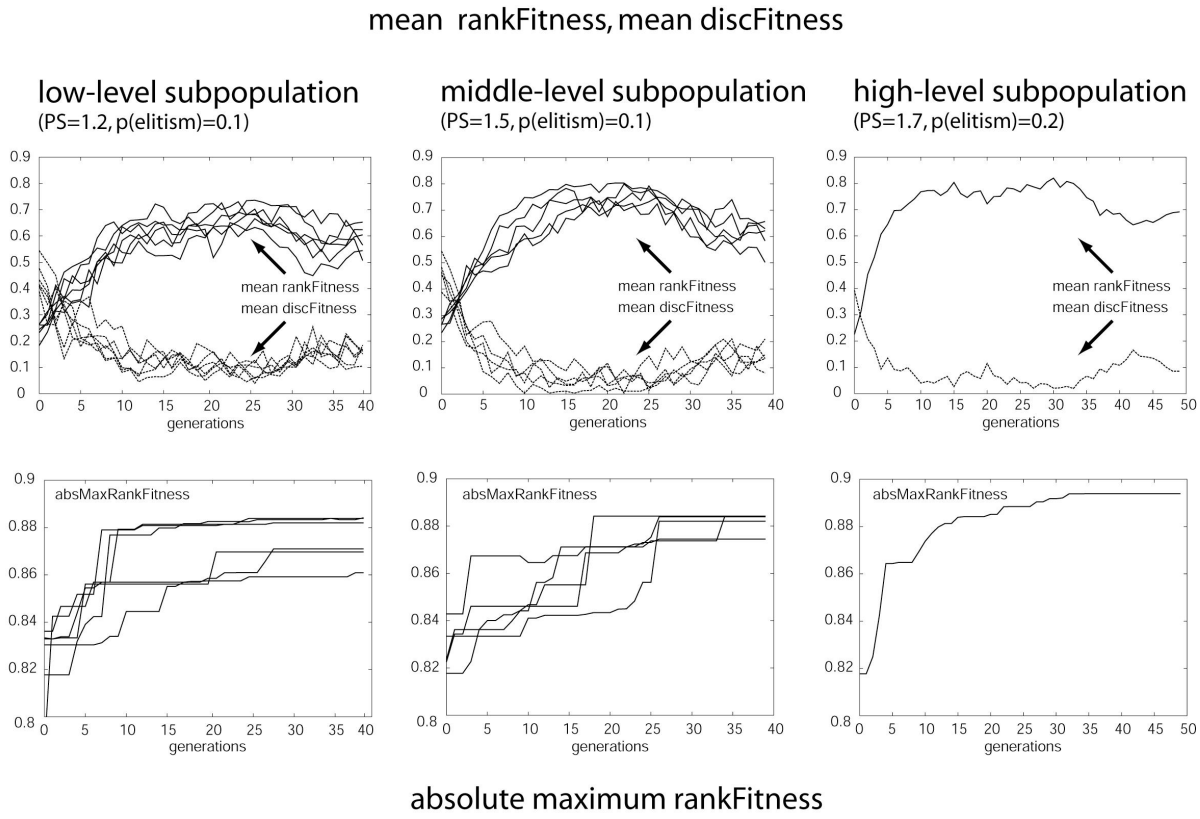



*Figure VII-14: Examples of convergence of the GP algorithm*
*Each column represents a subpopulation. Topology and parameters correspond to Figure VII-13. The graphs in first row show the mean rankFitness (whose optimum is 1) and the mean discFitness (whose optimum is near 0) reported for the population at each generation. As low-level populations perform several runs until the head-population ends its own run, several curves are drawn, each of them corresponding to the values reported for a run. The graphs in the second row represent the maximum rankFitness obtained over all generations. As generations are completed, more and more efficient scoring functions are discovered by the algorithm. At generation 33, a third objective, sizeFitness, is considered during the selection procedure. At this point, the algorithm tries to find smaller trees, resulting in a decrease of mean rankFitness and mean discFitness.*

## VII.4. Learning sets

### VII.4.1. Introduction

The evaluation of solutions in our GP algorithm requires the use of a learning set composed of MS/MS spectra with known identifications. The learning has to be diversified such as to cover as well as possible the space of MS/MS spectra. It follows that the set should contain various spectrum qualities, from various peptides of various amino acid compositions and arrangements. The prerequisite requirement is that the correct identification for each spectrum be known. This means that the set has first to be identified using one or several identification algorithms that provide a high level of confidence and/or be manually validated. This is an important requirement, since including false positive identifications in the set would result in a bias in the GP process, and therefore lead in the learning of less discriminating scoring functions.

### VII.4.2. MS/MS spectrum gathering

To build such a learning set, we used MS/MS data obtained for a proteomic and transcriptomic study (Scherl et al. 2005) performed in the "Laboratoire Central de Chimie Clinique des Hôpitaux Universitaires de Genève" on Staphylococcus aureus strain N315. This pathogen causes frequent and potentially severe infections while rapidly acquiring antibiotic resistance (Naimi et al. 2003).

Scherl et al. applied various proteomic analyses on protein sets extracted from the pathogen. In one of them, a membrane extract, proteins were separated by SDS-PAGE, in-gel digested with trypsin, and analysed with a Q-TOF mass spectrometer. Among the 2088 MS/MS spectra acquired, 751 were correlated with peptide sequences leading to the identification of 269 different proteins. The peptide identifications were performed using Mascot 1.8[5] with the following parameters: precursor mass tolerance was set to 2.0 Da, and fragment error was set to 1.0 Da. A maximum of one missed-cleavage was accepted. Cysteine carbamidomethylation was set as a variable modification and methionine oxidation was set as a fixed modification. The combined Swiss-Prot and trEMBL databases were searched without species restriction. Identifications were validated when the top-scoring peptide corresponded to the correct species (*Staphylococcus aureus*) or to known contaminants, and when the score was above the significance threshold given by Mascot. To minimize the presence of false positives in the learning set, we performed a new identification run on the 751 spectra using Phenyx (November 2004 release[6]). Parameters were set similarly than for Mascot, but the search was limited to the *Firmicutes* taxonomy (+ contaminants). 29 spectra were removed from the learning set because the best-scoring peptide proposed by Phenyx did not correspond to the assignment of Mascot. Since the fragmentation pattern strongly depends on the number of charges carried by the precursor, we decided to focus the learning phase on doubly charged spectra. New scoring functions will have to be learned in the future to optimize Popitam for working with other precursor charge states (as well as with other spectrometer types). We also removed from the learning set spectra with modifications (such spectra could interfere with our modification

[5] http://www.matrixscience.com/search_form_select.html
[6] http://phenyxbeta.vital-it.ch/phenyx/login/login.jsp

simulation methodology, described in Section VII.4.4), as well as contaminants (some spectra corresponded to peptides from the Lysostaphin protein, which is a murolytic enzyme used during sample preparation to degrade the staphylococcal cell). At the end, the learning set was composed of 597 MS/MS identified spectra. Figure VII-15 illustrates the various processing steps for building the learning set.

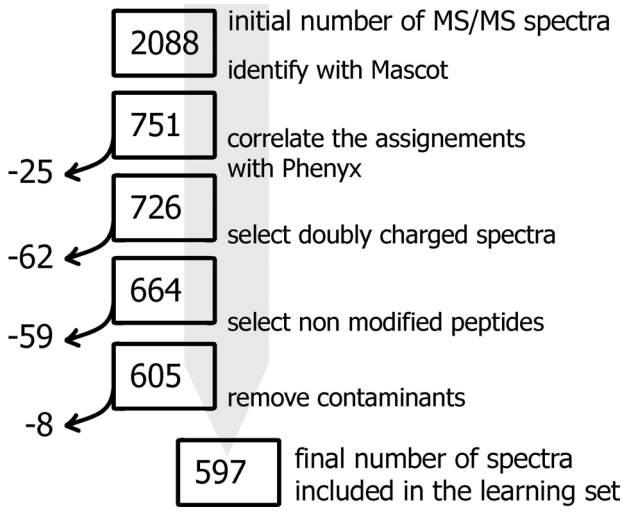



*Figure VII-15: Learning set building*
*A total of 597 among the 2088 initial MS/MS spectra were selected to form the learning set. The 597 selected spectra fill the following requirements: they obtained a significant score with Mascot and their identification was validated by Phenyx; their precursor mass was doubly charged; the assigned peptides were not modified and were present in the Staphylococcus aureus (N315) database.*

The ideal size of a learning set can be subject to discussion. We observed that methods that employ learning sets to optimize parameters generally use sets of several hundreds or even several thousands of spectra. One hundred MS/MS spectra may nevertheless be sufficient, in given conditions, to build efficient scoring functions (Masselot et al. 2003). But the number of spectrum is not the only criterion to meet. Redundancy, which measures the presence of multiple spectra for a given peptide sequence, is a characteristic that should also be considered and minimized. Our learning set contains rather few redundant spectra, as shown in Figure VII-16, since a total of 501 different peptides are represented by the 597 spectra.

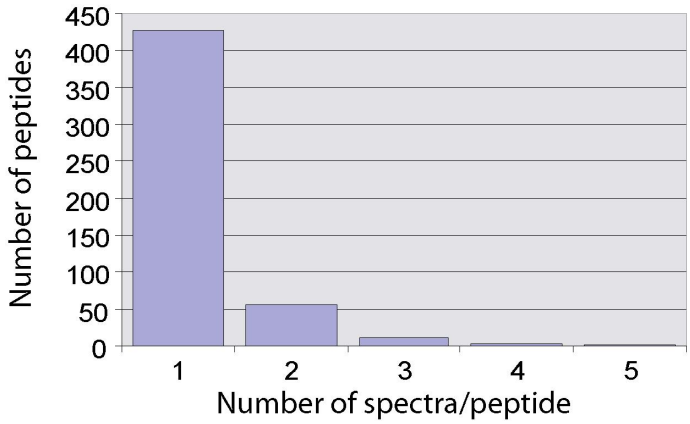



*Figure VII-16: Peptide redundancy*
*This histogram shows the distribution of peptide redundancy in the set of 597 spectra. The total number of distinct peptides represented in the set is 501. 428 spectra come from unique peptides, 56 peptides are represented each by 2 spectra, 12 are represented each by 3 spectra, 4 are represented each by 4 spectra and finally, one peptide is represented by 5 spectra.*

### VII.4.3. Spectrum quality

Spectrum quality is another important criterion that should be taken into account during the building of a learning set. Ideally, a learning set should include spectra of various qualities. The latter can be estimated in different ways. For example, Pevzner et al. (Pevzner et al. 2001) introduced a basic quality measure, denoted here as $p_{by}$, which is computed according to:

$$p_{by}=\left(\frac{m_b+m_y}{l-1}\right)/2$$

*where*

*$m_b$ and $m_y$ are the sequence coverage observed with b-ion type and y-ion type fragments (two peaks with very similar values are counted as one match), and*
*$l$ is the number of amino acids in the identified peptide (in (Pevzner et al. 2001) the denominator is l instead of l-1).*

The quality measure $p_{by}$ varies between 0 (no b- and y-ions are observed for any of the cleavage positions) and 1 (both b- and y-ions are observed for each cleavage position).

Figure VII-17 represents the cumulative graph and the histogram of the quality measure $p_{bys}$ computed for the 597 spectra using the corresponding assigned peptide sequences. The odd pattern observed in the cumulative curve, notably at position 0.5, is due to the presence (in the equation used to compute $p_{by}$) of the discrete variable $l$ that represents the peptide length and is comprised in our learning set between 7 and 25 (see Figure VII-18). This results in a discontinuous curve and in statistically over-represented positions (vertical trails visible in positions 0.5, 0.6, 0.625, and 0.667).

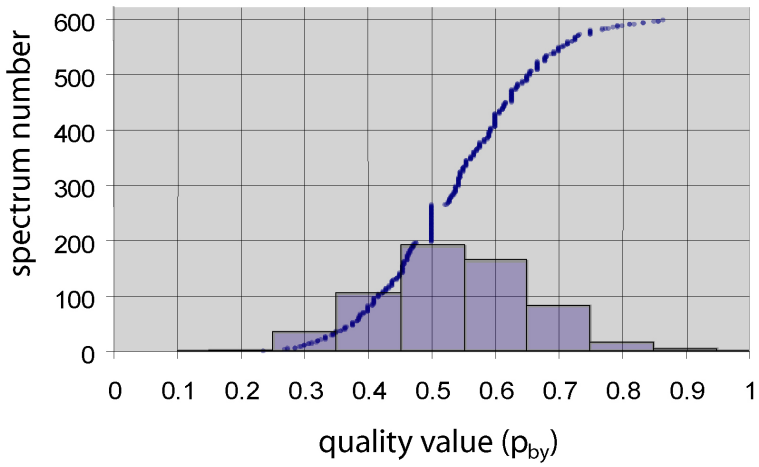



*Figure VII-17: Quality indices*
*Cumulative graph and the corresponding histogram of the quality index $p_{by}$ computed for the 597 spectra of the learning set. About one third of the spectra have a quality measure below 0.5. One third have a quality value between 0.5 and 0.6, and a third have a quality value above 0.6.*

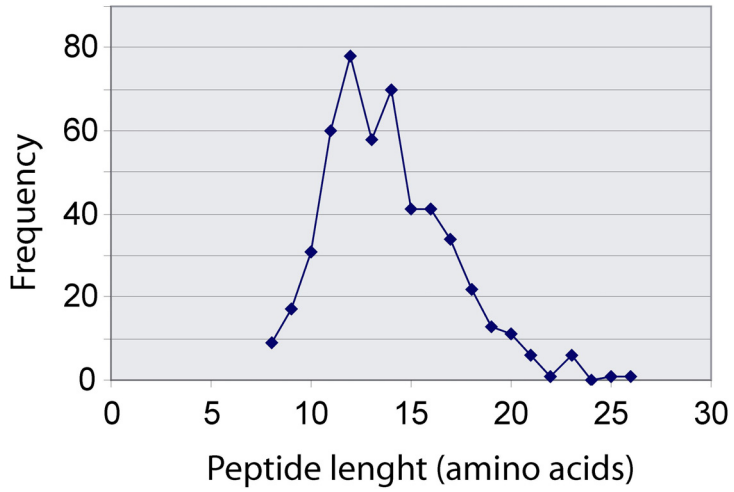



*Figure VII-18: Peptide length*
*This histogram shows the distribution of length of the 501 non-redundant peptides. In this dataset, the mean peptide length falls between 13 and 14 amino acids. The shortest peptide was 8 amino acids and the longest one was 26 amino acids.*

As it is defined, $p_{by}$ possesses the great advantage to give direct knowledge about the sequence coverage by the peaks (a $p_{by}$ of 0.5 means that, when taking into account b and y ions, half of the positions are not represented in the spectrum). But is has two drawbacks: first it does not take into account minor ion types (e.g. a-ion type, ion types with molecule losses, doubly-charged ion types); second, it is not weighted by the presence of contiguous fragmentation positions. Additional terms could therefore be introduced to more completely reflect the quality of spectra. For example, the observed number of a-ion types can be taken into account in the same way than b- and y-ion types.

Moreover, a continuity measure $q_{by}$ taking into account the presence of ion series (continuous fragmentation) can be computed according to:

$$q_{by} = \left( \frac{c_b}{m_b - 1} + \frac{c_y}{m_y - 1} \right) / 2$$

*where*

> *$c_b$ and $c_y$ are the cumulated number of fragmentation positions in peak series (see Figure III-5 for the definition of a peak series) and $m_b$ and $m_y$ are the sequence coverage by b-ion and y-ion fragments.*

The continuity measure $q_{by}$ varies between 0 (each peak match is isolated) and 1 (all b- and y-peak matches are contiguous). Figure VII-19 gives an example of computation of $p_{by}$ and $q_{by}$ for a given spectrum.

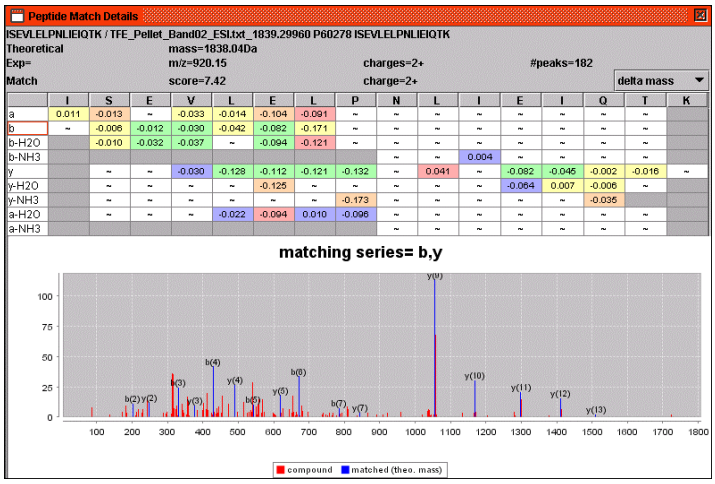



*Figure VII-19: Example of computation of $p_{by}$ and $q_{by}$*
*Example of computation of the quality measure $p_{by}$ and continuity measure $q_{by}$ for a given spectrum. This snapshot of the "Peptide match details" window from Phenyx shows matched peaks between an experimental spectrum and its corresponding identified peptide ISEVLELPNLIEIQTK. Coverage by b-ion types is 6. Coverage by y-ion types is 10. Number of possible fragmentation positions is 16 (peptide length minus 1). The quality measure $p_{by}$ for this spectrum is (6/16+10/16)/2 = 0.5. The continuity measure $q_{by}$ is (5/6 + 7/10)/2 =0.77.*

The observed quality values for the learning set are quite high. This shows the presence of a bias toward good quality spectra, as expected, since we selected only spectra that were confidently identified by both Mascot and Phenyx. Lowering the confidence level would have resulted in introducing false negatives in the set, which was precisely what we wanted to avoid. In addition, such a bias is not really awkward in the context of our study, since Popitam was not designed to specifically identify poor quality spectra, but rather spectra with unexpected modifications resulting in mass shifts that complicate their interpretation. Moreover, a key feature of Popitam is to locate and characterize the modifications. This requires spectra with high-quality fragmentation.

### VII.4.4. Modification simulation

As it was built, the learning set of 597 spectra did not contain spectra of modified peptides. Because Popitam is designed to find unexpected modifications, we need to have learning sets specifically composed of spectra obtained from modified peptides. Moreover, we want to optimize scoring functions according to the run mode of Popitam. In mode MODGAPS=0, the scenarios contains mainly long tags, possibly separated by *lackGaps*. In mode MODGAPS=1, Popitam only considers scenarios with exactly one *modGap* (due to one or several modification events) and zero or more *lackGaps*. In mode MODGAPS=2, Popitam considers only scenarios with exactly two *modGaps* (each of them due to one or several modification events) and zero or more *lackGaps*. Three different learning sets have then to be built. A first one, composed of unmodified spectra, that will serve to learn a function for runs launched in mode MODGAPS=0. A second one, with spectra carrying one modification, for runs launched in mode MODGAPS=1; and a third one with spectra carrying two separate modifications for runs launched in mode MODGAPS=2.
The current subscores being dependent on the number of *modGaps*, we have to learn a separate scoring function for each considered number of *modGaps*. Nevertheless, it is certain that learning a unique function for scoring the scenarios would be more convenient. This issue could be tackled in future developments.

Collecting several hundreds of spectra with one, two or three post-translational modifications or mutations, moreover confidently identified, is quite a difficult task.
A first method to avoid this issue was to use spectra with expected chemical modifications and to run Popitam without specifying what kind of modifications are present. The 726 confidently identified doubly charged spectra comprised 59 spectra issued from peptides with carbamidomethylated cysteines or oxidated methionines. Unfortunately, this option was not satisfying, because the number of such spectra in our set was too low to ensure efficient learning.
A second possibility was to simulate the presence of modifications on the very spectra. A simple procedure is described by Algorithm VII-1. $S=\{s_1, s_2, ..., s_{|S|}\}$ is the spectrum to modify, and $P=\{a_1, a_2, ..., a_l\}$ its assigned peptide sequence. The aim is to model a modification on amino acid m with a delta value $\delta$ (m and $\delta$ are randomly chosen), given $\Delta$ a set of available ionic hypotheses. The algorithm computes the expected peak mass $\mu(s_{exp})$ for a cleavage position p and an ionic hypothesis $\eta_k$ (the fragment must include the modified amino acid). Then it searches in the spectrum for a similar peak mass $\mu(s_i)$ and shifts the latter according to the delta value $\delta$ and the charge of the current ionic hypothesis.

For k=0 to $|\Delta|$ {
  if $(t(\eta_k) = 'N')$ {
    For p = m to l-1 {
      $\mu(s_{exp}) = \left[\mu(H) + \sum_{n=0}^{p} \mu(a_n) + (c(\eta_k)-1) + o(\eta_k)\right] / c(\eta_k)$ ;
      For i=0 to |S| {
        if $(\text{match}(\mu(s_i), \mu(s_{exp})))$ shift $(\mu(s_i), \delta/c(\eta_k))$;
      }
    }
  }
  if $(t(\eta_k) = 'C')$ {
    For p = 1 to m{
      $\mu(s_{exp}) = \left[\mu(OH) + \sum_{n=p}^{l} \mu(a_n) + (c(\eta_k)-1) + o(\eta_k)\right] / c(\eta_k)$ ;
      For i=0 to |S| {
        if $(\text{match}(\mu(s_i), \mu(s_{exp})))$ shift $(\mu(s_i), \delta/c(\eta_k))$;
      }
    }
  }
}

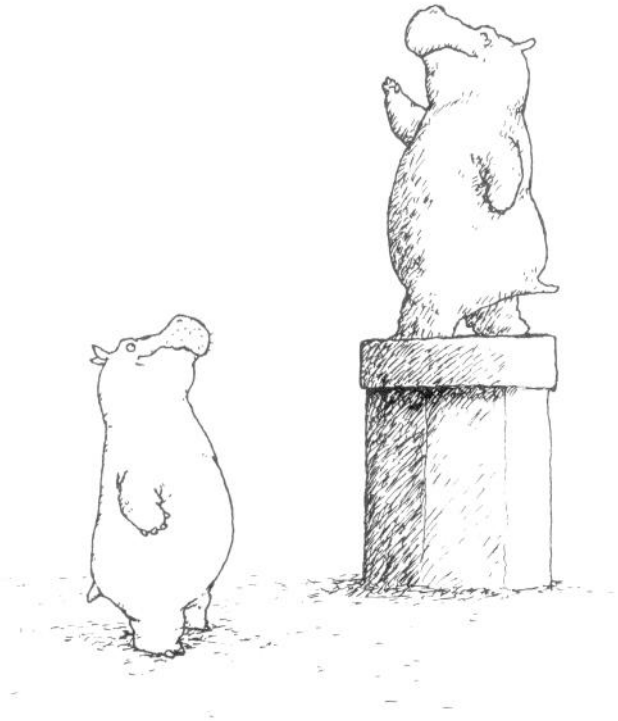
*Algorithm VII-1: Simulating a modification by acting on the spectrum*

Despite its attractiveness, this solution presented an annoying drawback: peak selection is dependent on two adjustable parameters, a mass tolerance error and a set of possible ionic hypotheses. An incomplete set Δ would lead to missing shifted peaks. On the other hand, setting too large an error threshold could lead to random matches and then incorrectly shifted peaks. For these reasons, we finally chose yet another solution: the spectra remained unchanged, but their assigned peptide sequences were mutated together with all occurrences of the sequences in the database. In order to maintain the same cleavage positions, amino acid K, R, and P were not allowed to mutate nor to be chosen as amino acid replacements. In addition, when the number of modifications was equal to 2, the mutation positions were chosen so that they were separated from each other by at least three amino acids.

Using this method, we built three databases from our initial set of spectra (Table VII-1). The first set, LS_597_MOD0.pop comprises all the original spectra and assigned sequences. For the second and third ones (558_MOD1.pop and 471_MOD2.pop), we discarded spectra with overlapping peptide sequences, since their presence was disturbing our database modification algorithm. In the third one, spectra with too short amino acid sequences were also discarded. In addition, spectra for which no scenario was proposed for the correct candidate peptide were discarded. Such a situation can happen because of the scenario coverage filter (see Section VI.9). Each set was then used independently to learn and test scoring functions for evaluating scenarios, given a number of *modGaps*. The methodology and results are presented in Chapter VIII.

| Data sets | Database |
|---|---|
| 597_MOD0.pop | FIRM_MOD0.fas (Swiss-Prot + trEMBL Firmicutes) |
| 558_MOD1.pop | STAPH_N315_MOD1.fas (Swiss-Prot + trEMBL Staphylococcus aureus N315) |
| 471_MOD2.pop | STAPH_N315_MOD2.fas (Swiss-Prot + trEMBL Staphylococcus aureus N315) |

*Table VII-1: The data sets and corresponding databases.*



J'aimerais que l'on érige une statue de moi.
Non pas pour mes résultats,
seulement pour moi.

It would be wonderful to have a statue of me.
Not for my results,
just for myself.

Figure adapted from "Monsieur Hippopotame" by Tanikawa Shuntarô

# CHAPTER VIII

# RESULTS AND DISCUSSION

This chapter presents various results obtained in the context of scoring functions tailoring with Genetic Programming and peptide identification and characterization with Popitam. The potential of our method is shown and discussed using specific examples.

# VIII. Results and discussion

## VIII.1. Introduction

This section presents various results we obtained in the context of scoring function tailoring with Genetic Programming and peptide identification and characterization with Popitam. The first part of the chapter concerns the scoring functions. Using a procedure that includes ten learning runs and cross-validations, we learned and tested 60 scoring functions aimed at scoring scenarios with 0, 1 or 2 *modGaps*. We show that Genetic Programming is robust and that the learned functions are efficient when used with unseen data. We compare the efficiency of the learned functions with the empirical function shown in Section VI.10.8 and with a basic one. The second part of the chapter presents results obtained with Popitam on a different set of MS/MS data containing real modified peptides. We notably show representative identification results of peptides with PTMs, as well as peptides that underwent non-conform cleavage or transpeptidation

## VIII.2. Scoring functions tailoring

### VIII.2.1. Procedure

Three independent experiments, named EXP_MOD0, EXP_MOD1, EXP_MOD2 and aimed at discovering efficient functions for scoring scenarios with 0, 1 or 2 *modGaps* have been performed. Each of the experiments comprises four steps (charted in Figure VIII-2): a) random sampling, b) sample file building, c) function learning and d) validation. In the first step, a learning set and a testing set are built by randomly sampling spectra from the data sets presented in Table VII-1. The sets are not overlapping; therefore, the functions are tested on unseen data. In the second step, the spectra of the learning sets are inputted to Popitam in order to create the "training examples" that will be used by the Genetic Programming algorithm. The third step is the GP learning process. Topology and parameters are the ones presented in Section VII.3.3. Section VIII.2.2 in the present chapter discusses the convergence of the GP towards solutions with high-fitnesses in all three experiments. Finally, the fourth step (validation) is thoroughly covered by Section VIII.2.3.
To assess the robustness of the GP approach, we repeated the procedure "random sampling-learning-testing" ten times (runs 0 to 9).

As shown in Figure VIII-1, Popitam intervenes twice: a first time during the creation of the sample files for the GP algorithm, and a second time during the validation phase. Popitam's parameters are kept unchanged across the two runs, but they are differently tuned in each experiment (see Table VIII-1). For example, EXP_MOD0 uses a tight precursor mass filter. This filter is relaxed in the two other experiments, which is a necessary condition for open-modification search..

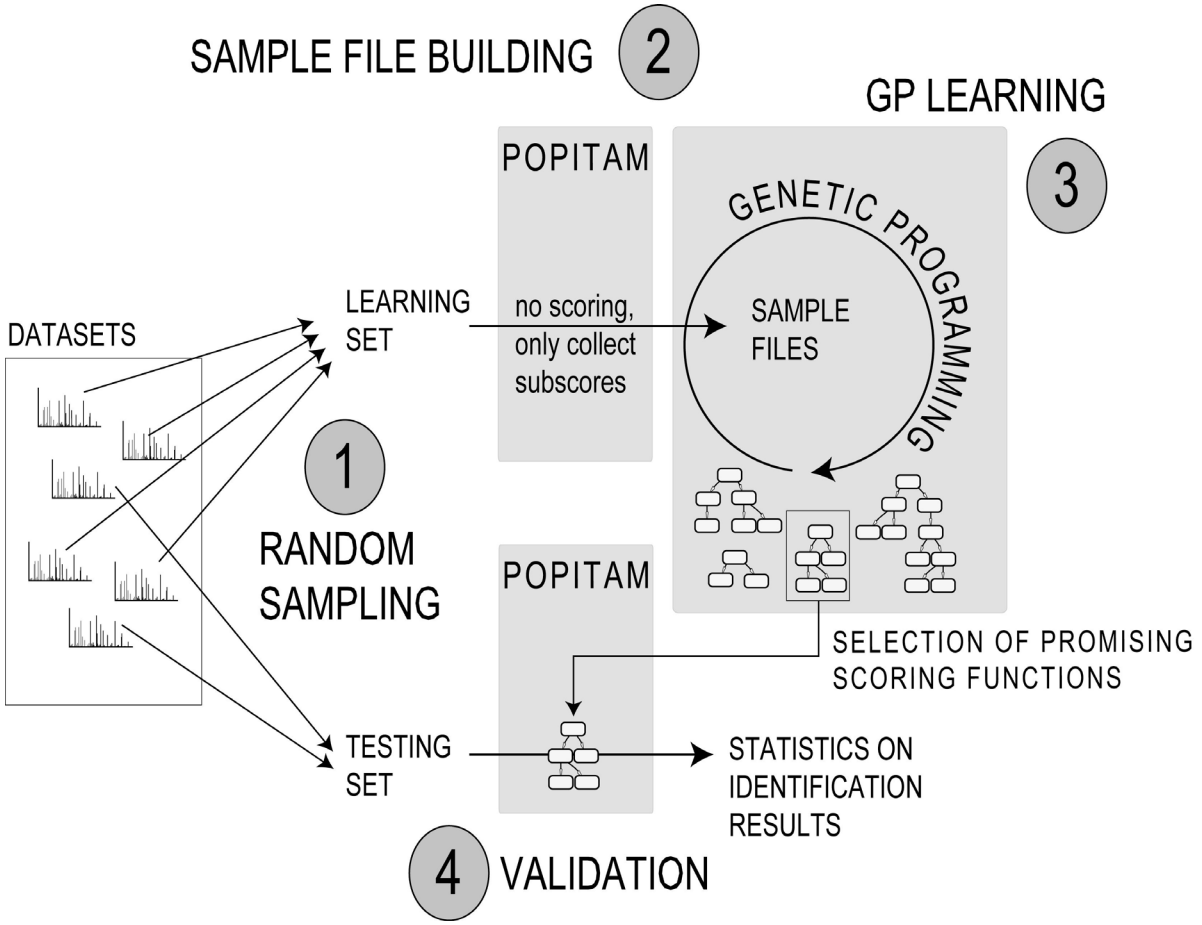



*Figure VIII-1: The sampling-learning-validation procedure.*

On the other hand, the database is smaller in EXP_MOD1 and in EXP_MOD2. Notably, an AC filter is used in EXP_MOD2. The filter is composed of a list of 160 authorized protein accession codes (ACs). The ACs correspond to the proteins identified by at least one spectrum in the data sets of Table VII-1. By this mean, the number of proteins to be effectively digested is reduced from 2'600 to 160. The tuning is necessary to control the size of the sample files that are used as "training examples" by the GP process (Figure VII-12 explains how the sample files are obtained). If the AC filter is removed in EXP_MOD2, the total size of the sample files is about 3Gb and the time required to complete one GP generation is too important (several days). Also, the COVBIN and FRAGMENT_ERRORs parameters are differently tuned in the various experiments. COVBIN is increased when looking for scenarios with two *modGaps*, leading to bigger spectrum graphs and more complete scenarios. On the other hand, the use of smaller fragment errors results in a decrease of the graph connection rate, and consequently in the number of possible paths to explore, thus counteracting the expansion of the search space caused by the allowed jumps in the spectrum.

**EXP_MOD0_runs0..9:**

| dataset: 597_MOD1.pop | DBs and FILTERS | Popitam's PARAMETERS |
|---|---|---|
| LS: 450 spectra,<br>representing 132 Mb of sample files | Firmicutes_MOD0<br>(non modified peptides)<br>~175'000 proteins<br>PREC_MASS: **+/-2.5 Da**<br>AC_FILTER: **NO** | MOD_GAPS = 0<br>COVBIN:6<br>FRAGMENT_ERROR1:0.6 Da<br>FRAGMENT_ERROR2:1.2 Da<br>MAX_ADD_MOD: 0 Da<br>MAX_LOSS_MOD: 0 Da<br>MIN_COV_ARR: 0.3 |
| TS: 147 spectra | | |

**EXP_MOD1_runs0..9:**

| dataset: 558_MOD1.pop | DBs and FILTERS | Popitam's parameters |
|---|---|---|
| LS: 450 spectra,<br>representing 126 Mb of sample files | Staph_N315_MOD1<br>(peptides modified once)<br>~2'600 proteins<br>PREC_MASS: **+/-150 Da**<br>AC_FILTER: **NO** | MOD_GAPS = 1<br>COVBIN:6<br>FRAGMENT_ERROR1:0.2 Da<br>FRAGMENT_ERROR2:0.4Da<br>MAX_ADD_MOD: 150 Da<br>MAX_LOSS_MOD: 150 Da<br>MIN_COV_ARR: 0.3 |
| TS: 108 spectra | | |

**EXP_MOD2_runs0..9**

| dataset: 471_MOD2.pop | DBs and FILTERS | Popitam's parameters |
|---|---|---|
| LS: 400 spectra,<br>representing 245 Mb of sample files | Staph_N315_MOD2<br>(peptides modified twice)<br>~2'600 proteins<br>PREC_MASS: **+/- 250 Da**<br>AC_FILTER: **YES (160 ACs)** | MOD_GAPS = 2<br>COVBIN:7<br>FRAGMENT_ERROR1:0.2 Da<br>FRAGMENT_ERROR2:0.3 Da<br>MAX_ADD_MOD: 150 Da<br>MAX_LOSS_MOD: 150 Da<br>MIN_COV_ARR: 0.5 |
| TS: 71 spectra | | |

*Table VIII-1: Conditions and parameters of the three experiments*
*Overview of the different experiment conditions and parameters set for the sample file building and the validation phase.The first column provides indications about the size of the learning and testing sets (respectively LS and TS). The second column lists the databases and the filters used with Popitam. The third column gives the settings of Popitam's parameters.*

## VIII.2.2. Convergence of the GP process

Statistics collected during the generations of the GP process give information about the evolution of the populations of scoring functions. The mean *rank- and discFitnesses* show the convergence of the exploration towards efficient scoring functions, and the mean *sizeFitness* is a measure of the function's diversity. Figure VIII-2 shows the algorithm convergence towards high *rankFitness* values and low *discFitness* values for the three experiments. *RankFitness* optimum is 1 (all spectra of the learning sets are correctly identified) while *discFitness*'s optimum is close to 0 (all correctly identified spectra have a very small p-value). Although convergence is quite rapid, solutions with better *rankFitness* are found every few generations (vertical arrows). After two-thirds of generations, the introduction of a third objective to optimize (the *sizeFitness*) disrupts the convergence. From this point on, the algorithm is expected to produce smaller functions with more or less comparable

performance rather than keeping exploring the search space. In the EXP_MOD2 experiment, the GP algorithm struggles in discovering efficient scoring functions. In this mode, more scenarios are created for each candidate peptide, because two jumps are allowed for each path in the spectrum graph. Consequently, it is more difficult to build scoring functions able to efficiently separate the correct scenarios from all the incorrect ones.

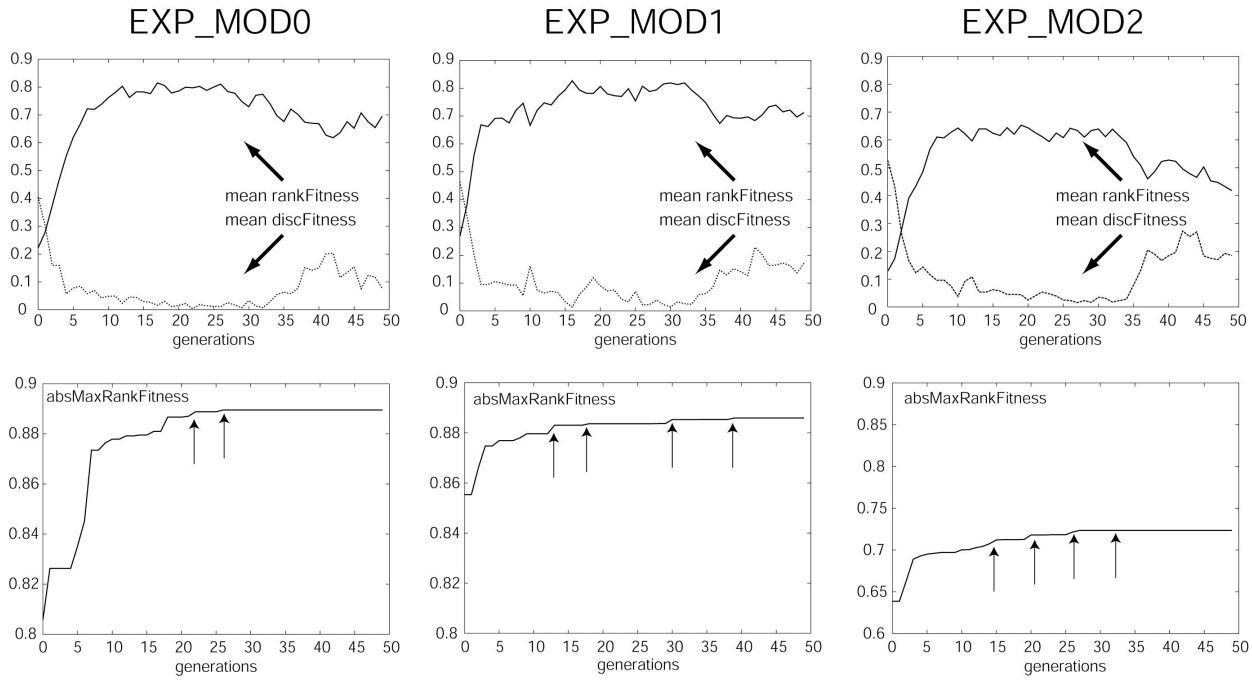



*Figure VIII-2: Evolution of a GP process (head-subpopulation).*
*The plots show the evolution of the head-subpopulation for the three experiments. The first row shows the evolution of the mean rankFitness and discFitness computed from the population of functions for each generation. The second row shows the "best-so-far" rankFitness at each generation.*

## VIII.2.3. Testing phase

### VIII.2.3.1. Selecting promising solutions

At the end of a GP process, a list of co-dominating functions is reported (see Table VIII-2). Each of the co-dominating functions is characterized by a *rankFitness*, a *discFitness* and a *sizeFitness*. The problem is to choose the most promising ones from the set of proposed functions.

| run ID | EXP_MOD0 | EXP_MOD1 | EXP_MOD2 |
|---|---|---|---|
| run0 | 16 | 34 | 24 |
| run1 | 30 | 17 | 18 |
| run2 | 21 | 26 | 21 |
| run3 | 33 | 40 | 43 |
| run4 | 34 | 33 | 23 |
| run5 | 29 | 15 | 28 |
| run6 | 30 | 29 | 38 |
| run7 | 38 | 29 | 30 |
| run8 | 28 | 38 | 27 |
| run9 | 32 | 42 | 16 |

*Table VIII-2: Co-dominating functions*
*Number of co-dominating scoring functions reported by the head-population for each experiment and each run.*

Figure VIII-3 highlights a correlation between the *sizeFitness* and the other two fitnesses: as the number of nodes of a scoring function increases, the *rankFitness* tends to increase and the *discFitness* tends to decrease. In other words, complex functions give better identification rates and show higher discrimination capacities than more parsimonious ones (on the learning sets). The most evident explanation for this is that complex functions catch a greater amount of information and can therefore better respond to the variability of the presented cases.

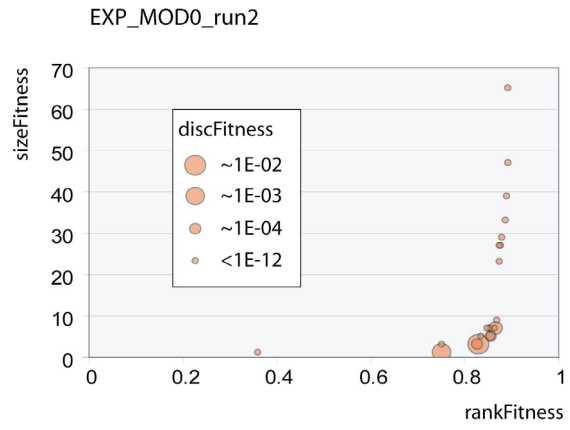



*Figure VIII-3: Fitness variations in co-dominant functions*
*The plot shows the fitnesses of the 21 co-dominant functions reported by EXP_MOD0 run2. Each disc in the graph represents a scoring function. The discFitness is represented by the disc diameters.*

Nevertheless, the *disc-* and *rankFitness* decrease very slowly compared to the *sizeFitness*. Therefore, functions of very different sizes may have more or less similar performance. We decided to select for each run and each experiment two functions, a complex one and a parsimonious one (see Figure VIII-4) and to perform the different validation tests on both categories of functions. For each run, we selected as complex function the function with the best *rankFitness* among all co-dominating

functions, and as parsimonious function the function with the best *rankFitness* among the set of co-dominating function of 15 nodes or less. The choice of the 15-node size was motivated by the fact that it appeared to be a good compromise between small size and good *rank- and discFitnesses*. Using this method, we collected 10 different complex functions and 10 different parsimonious functions for each experiment.

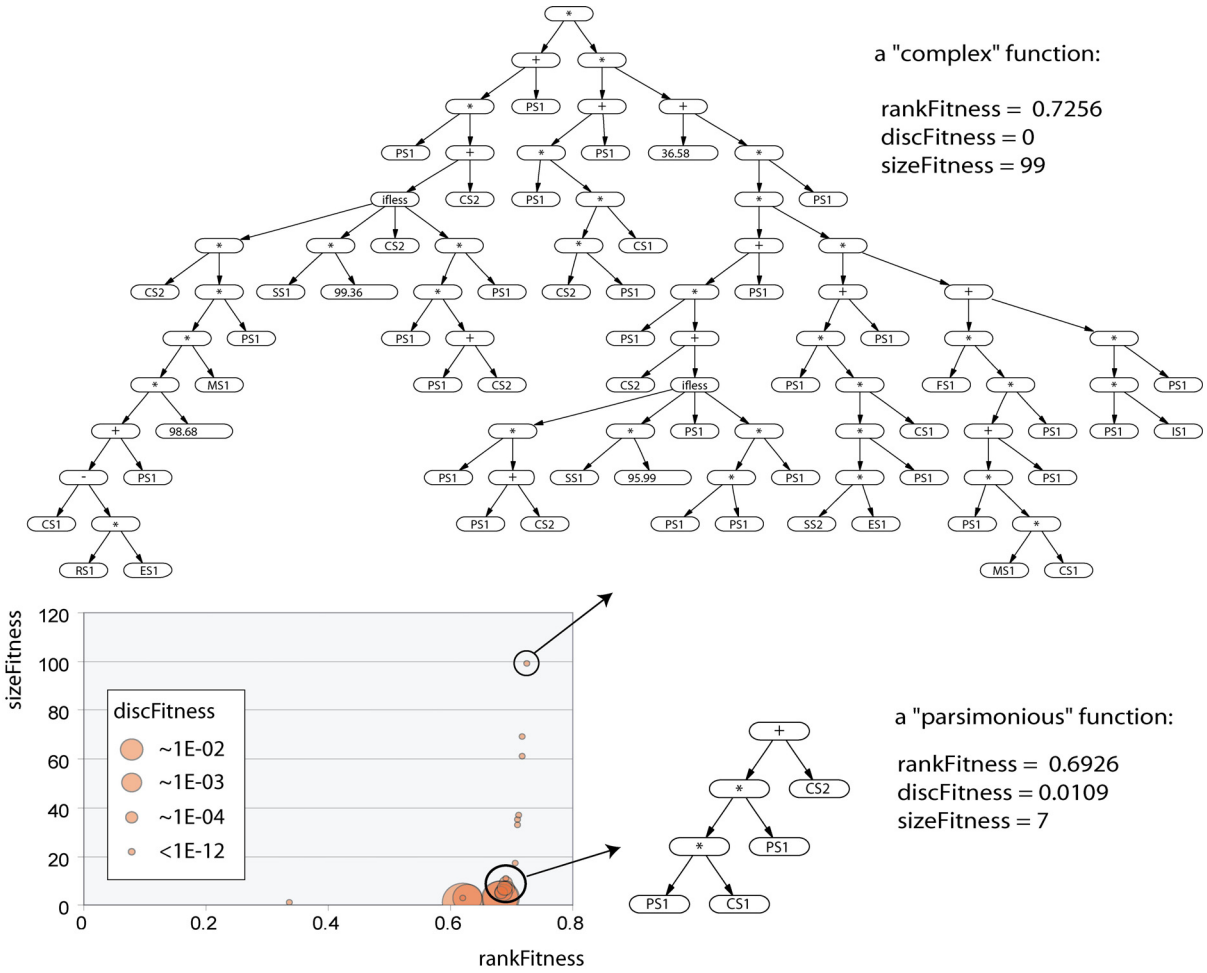



*Figure VIII-4: A complex and a parsimonious function*
*The functions come from experiment EXP_MOD2 (run3). Both functions are co-dominants (the complex one is better in rankFitness and discFitness, but the parsimonious one is better in sizeFitness).*

To be tested, the various functions are fed to Popitam, which is run with the corresponding testing sets and databases. During or after the identification, various statistics are collected, like the number of correct candidate peptides with first rank (rank-based performance) or the score distributions for correct candidate sequences (positive peptides) and incorrect candidate sequences (negative peptides). Based on these statistics, several topics are discussed: Section VIII.2.3.2 tackles the eventuality of overfitting by comparing the rank-based performances of the functions in the learning sets and testing sets; Section VIII.2.3.3 compares the different categories of functions in terms of rank-based performance and in terms of score distributions; and Section VIII.2.3.4 focuses on specific learned

functions and analyzes their individual performance in terms of true positive and false positive rates (ROC curves).

### VIII.2.3.2. Overfitted functions

A critical issue with supervised learning methods is that the learned functions adjust to very specific features of the learning set. Such a situation is denoted as overfitting (Banzhaf et al. 1998). This may happen when the learning was performed for too many generations, when the learning set is too small or when there are too many parameters (in our case, subscores). Overfitted functions fit perfectly to the characteristics of the learning set, but are not capable of generalizing to unseen situations.
As highlighted in Figure VIII-3, as a result of the introduction of the third fitness (*sizeFitness*), the GP process reports scoring functions of very different sizes with similar performances. This observation led us to suppose that the complex functions might be overfitted. This issue is well-known in machine-learning methods such as decision trees, where the trees are regularly pruned to avoid overfitting. In order to confirm or invalidate the overfitting hypothesis, we compared the rank-based performance of the two classes of functions (*COMP* and *PARS*) in the learning sets and in the testing sets. Figure VIII-5 describes the procedure used to construct the boxplots represented in Figure VIII-6.

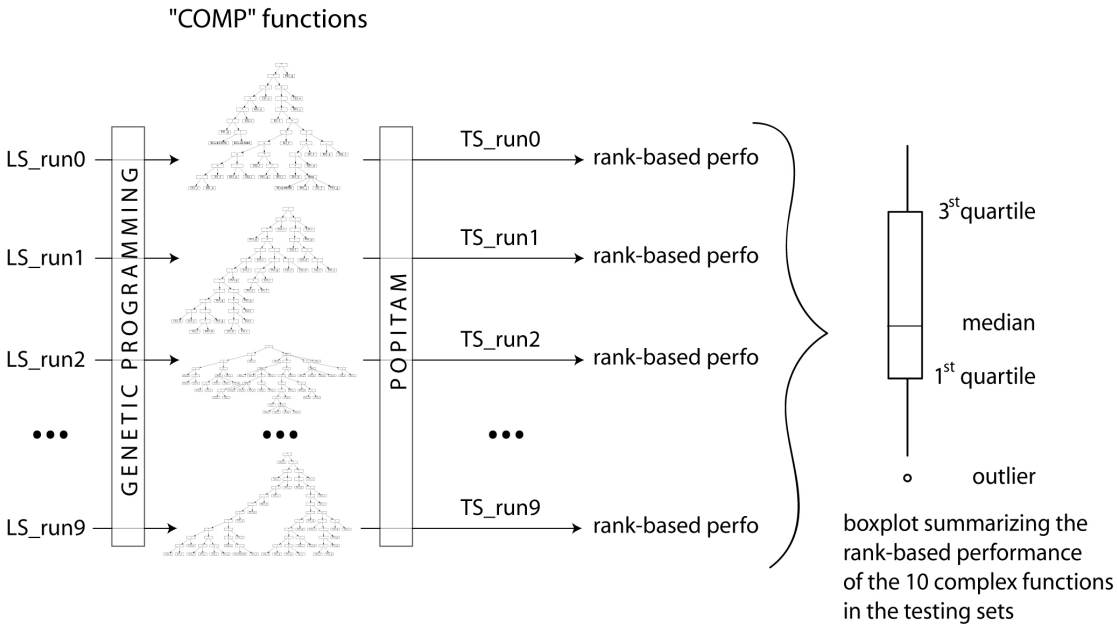



*Figure VIII-5: Procedure used to construct a boxplot*
*The boxplot synthesizes the performance (here in the testing sets) of the complex functions reported for the 10 runs. The line in the box is the median performance value (the upper and lower quartiles are the upper and lower edges). The point represents a suspected outlier. The ends of the vertical lines indicate the minimum and maximum data values (unless outliers are present). Various boxplots are obtained by changing the category of functions or replacing the testing sets (TS) by the learning sets (LS).*

We detected a clear example of overfitting for only one scoring function of the EXP_MOD2 experiment (see Figure VIII-6). This function is actually the complex function shown in Figure VIII-4. It should be noted that it is the one with the greatest number of nodes among the set of co-dominating functions reported by the GP algorithm for the third run.

Figure VIII-6 also shows that: a) each of the ten runs leads to functions of similar performance on the learning sets. Correspondingly, the GP approach can be considered as robust; b) the values are more spread out in the testing sets than in the learning sets; and c) the learned functions are slightly less performing on unseen data. Notably, the decrease in performance between the learning sets and the testing sets (computed as the difference between the median values of the 10 runs, and expressed as percentage) can be assessed to 3.1% (complex functions) and 1.8% (parsimonious functions) for the EXP_MOD1, and 10.4% (complex functions) and 8.9% (parsimonious functions) for the EXP_MOD2. No decrease in performance is observed for the EXP_MOD0.

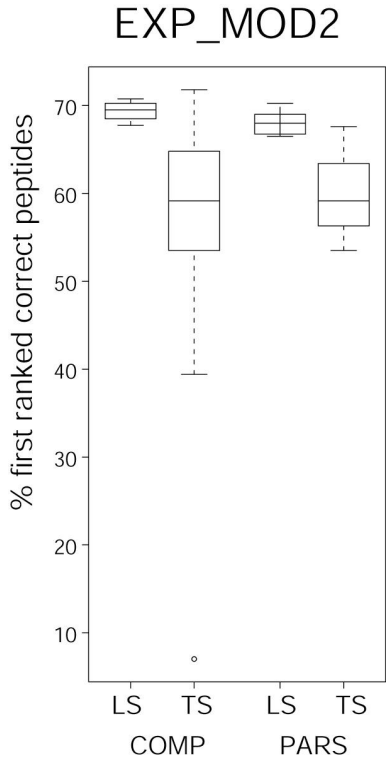



*Figure VIII-6: Example of overfitting in EXP_MOD2*
*The boxplot shows the performance (in percentage of first ranked correct peptides) in the learning sets (LS) and the testing sets (TS) of the 10 complex (COMP) functions and 10 parsimonious (PARS) functions selected from experiment EXP_MOD2. One of the complex functions obtained a very bad performance in its test set (only 7% of correct identifications) while having a high performance in its learning set (around 69% of first ranked peptides), suggesting a strong overfitting for this function.*

### VIII.2.3.3. Performance of learned functions compared to empiric functions

In order to validate the Genetic Programming approach, we compared the learned functions with the empirical scoring function presented in Section VI.10.8 and with a basic function composed of a single subscore (CS2). Figure VIII-7 shows that the learned functions obtain better results in the test sets than the empirical and basic ones (except for the *COMP* outsider function of EXP_MOD2). The complex and parsimonious functions obtain similar results in terms of rank-based performance, although the complex functions have tendency to give more spread out values than the parsimonious functions.

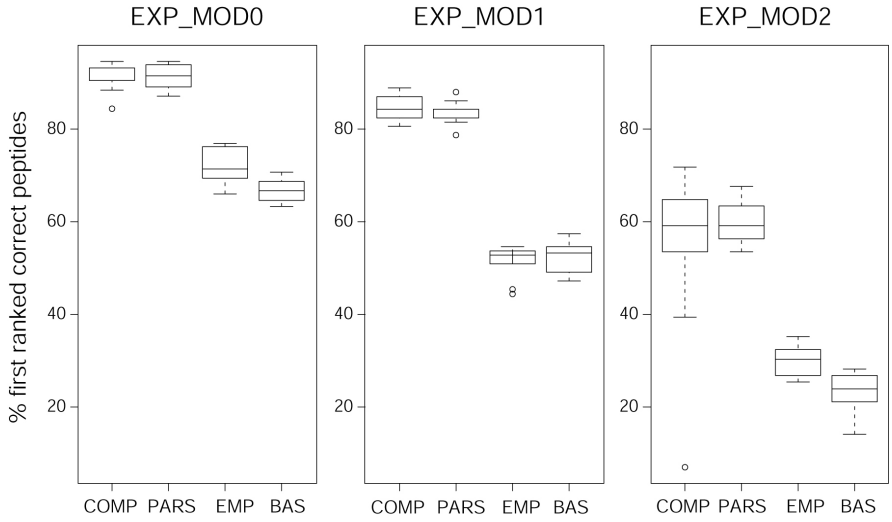



*Figure VIII-7: Comparison of scoring functions*
*In this figure, the rank-based performance of the four types of scoring functions (in the test sets) is compared. For each experiment, the boxplots are built by running Popitam on the ten test sets with the different functions (for the empirical and basic functions, the same function was used for each test set). The four categories correspond to the 10 complex (COMP) functions, the 10 parsimonious (PARS) functions, the empirical (EMP) function presented in Section VI.10.8 and a basic (BAS) function corresponding to a single subscore (CS2).*

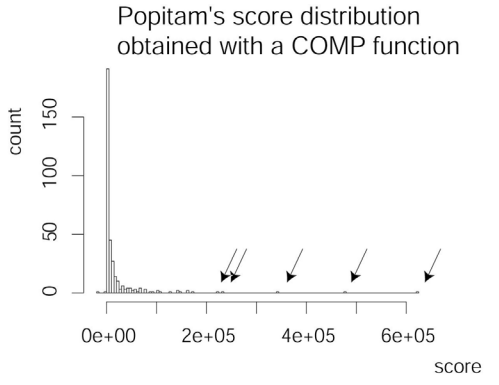
Table VIII-3 shows the gain reached by the complex and parsimonious functions over the empirical and basic ones for the three experiments.

| | **EXP_MOD0** | | **EXP_MOD1** | | **EXP_MOD2** | |
|---|---|---|---|---|---|---|
| % gain | EMP | BAS | EMP | BAS | EMP | BAS |
| COMP | +19.10 | +23.80 | +31.50 | +31.05 | +28.85 | +35.25 |
| PARS | +20.10 | +24.80 | +31.50 | +31.05 | +28.85 | +35.25 |

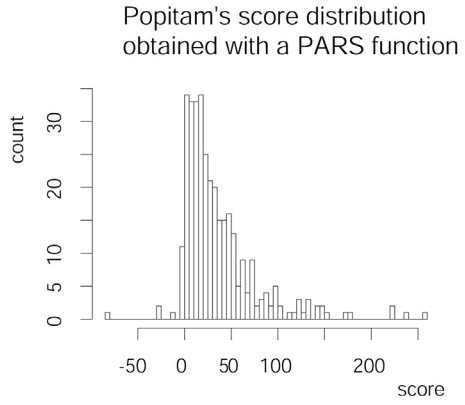
*Table VIII-3: Gain of the complex and parsimonious functions*
*Gain in terms of rank-based performance of the complex and parsimonious functions over the empirical and basic ones (computed as the difference between the median values of the 10 runs, and expressed as percentage).*

As shown in Table VIII-3 and Figure VIII-7, the complex and parsimonious functions are equivalent in terms of rank-based performance. Actually, the difference between the complex and parsimonious functions lies more in the score distributions they produce than in the percentage of correct first ranked peptides. As shown in Figure VIII-8, the complex functions usually give more outlier scores than the parsimonious ones. This complicates the interpretation of the results, because several high-ranked peptides may receive very low p-values (<1e-12). Such p-values are also observed for parsimonious functions when the number of analyzed candidate peptides is important.

In Popitam, finding out whether a match is true or random is difficult. The p-value estimates the probability to obtain a given score by chance. But due to the score distributions and to the function used to approximate it (the function assumes a gaussian distribution), the computed p-values may be of little help (see Figure VIII-17). For these reasons, Popitam also displays a score based on the ratio of adjacent scores (similar to the Sequest's delta score). We observed that Popitam's deltaScore, and more particularly the distribution of high-ranked peptides, are often efficient at distinguishing a correct match from a random one. Furthermore, the proposed scenario and the reported mass shifts associated with *modGaps* may help in confirming a result. At the moment, the decision remains subjective and requires manual interpretation. More work on this issue needs to be done, and notably, the function to approximate the p-value has to be improved.




| # | score (PVAL ; dS) | dtbSeq / scenarios |
|---|---|---|
| 1.| 92e05 | ALLNNMVFGVFQGYVK |
| | (<1.00e-12 ; 0.06) | alLNNMV**VFqg*** |
| 2.| 6e05 | DVNEQFSERPIHVK |
| | (<1.00e-12 ; 0.80) | *VN*QF----ihVK |
| 3.| 4.79e05 | YFNDVPNAVAHVKVK |
| | (<1.00e-12 ; 0.72) | **ND**naVahvkVK |
| 4.| 3.43e05 | MYSVISKILNFILVK |
| | (8.81e-08 ; 0.67) | ----IS**lnFI*VK- |




| # | score (PVAL ; dS) | dtbSeq / scenarios |
|---|---|---|
| 1.| 2448.92 | ALLNNMVFGVFQGYVK |
| | (<1.00e-12 ; 0.10) | alLNNMV*GV*QGYVK |
| 2.| 256.68 | YFNDVPNAVAHVKVK |
| | (1.80e-06 ; 0.93) | **ND**naVahvkVK |
| 3.| 239.91 | DVNEQFSERPIHVK |
| | (2.54e-05 ; 0.92) | *VN*QF----ihVK |
| 4.| 221.53 | MYSVISKILNFILVK |
| | (3.46e-04 ; 0.99) | -----IS**lnFIlv* |

*Figure VIII-8: Score distributions*

*Typical distribution of scores obtained for negative peptides with a complex (left) and a parsimonious (right) function. The distributions are composed of the scores of negative peptides collected for a given spectrum. Popitam's output for the 4 first ranked candidate peptides is shown below the plots. In both cases, the first ranked candidate peptide is the correct one. The complex function produces more extreme scores (arrows) than the parsimonious one. Correspondingly, in the left output, second and third ranked negative candidate peptides also receive a p-value (PVAL) above 1e-12. Fortunately, the deltaScore (dS) computed from the ratio between the first score and the second one suggests that the identification is correct.*

### VIII.2.3.4. Examples of learned functions

The benefits of methods like Genetic Programming for scoring function optimization are shown in Figure VIII-7 and Table VIII-3. Another interest is that the learned function can provide information about the importance of the different subscores. To discuss the results obtained with GP, we selected for each of the three experiments one complex function and one parsimonious function that showed good performance on their testing sets. The functions are shown below (Figure VIII-9, Figure VIII-10, Figure VIII-11):

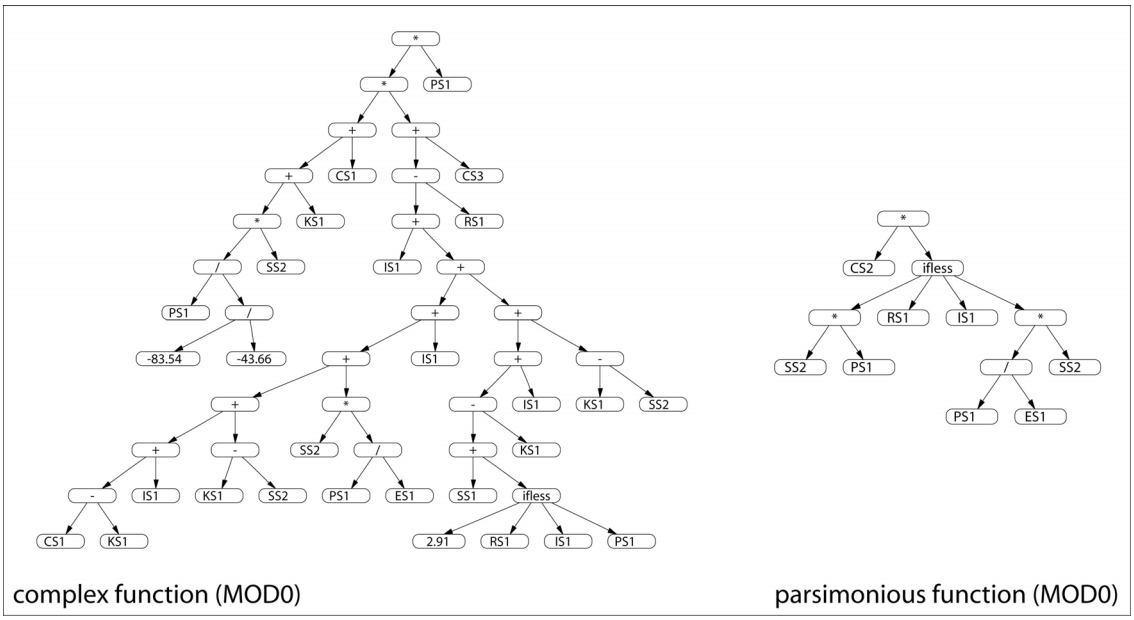



*Figure VIII-9: Two learned functions for scoring scenarios without modGaps*

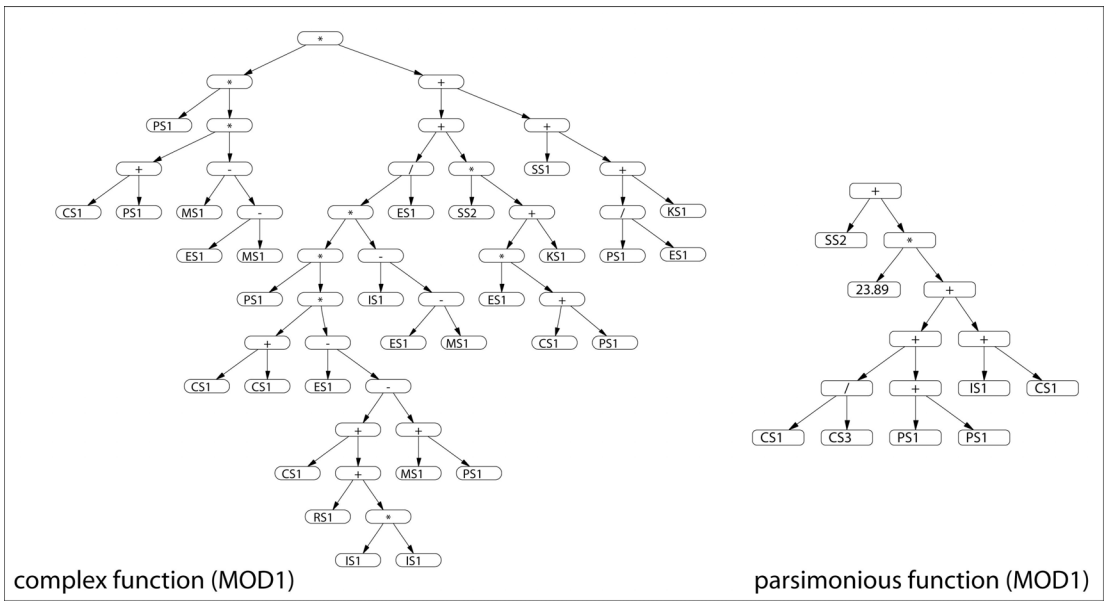



*Figure VIII-10: Ttwo learned functions for scoring scenarios with 1 modGap*

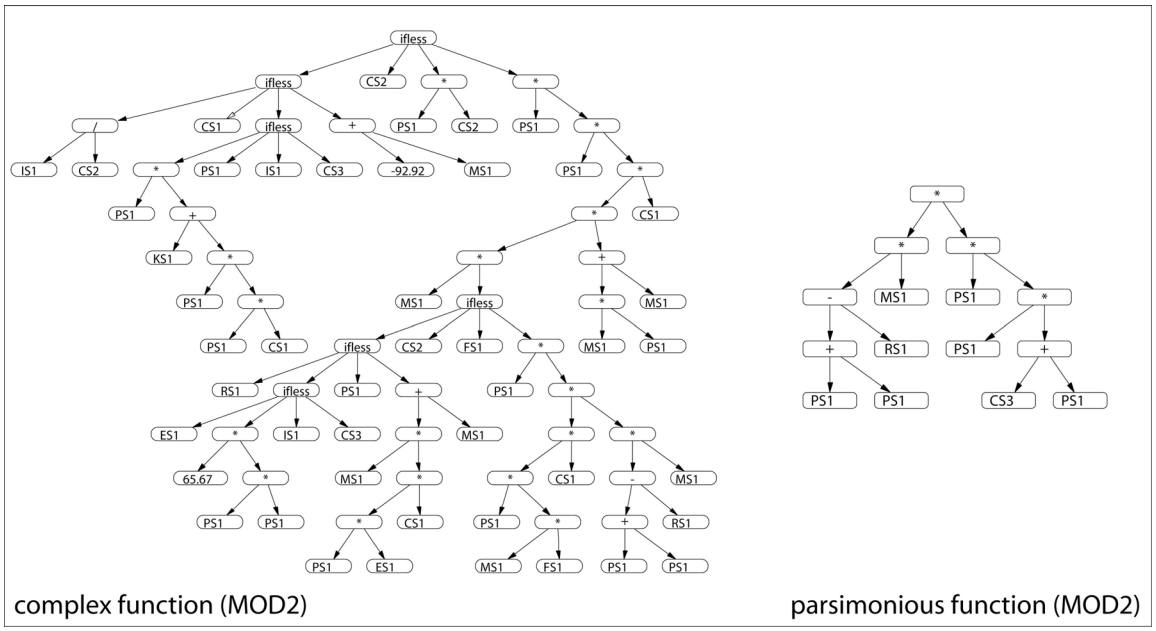



*Figure VIII-11: Two learned functions for scoring scenarios with 2 modGaps*

The COMP functions are too complex to be easily interpretable. But it is possible to make certain general observations. For example, it is evident that each of them uses a large part of the available subscores, and that none of the subscores is systematically discarded. We expect the error score (ES1) and the redundancy score (RS1) to be used "negatively" (either as denominator or in subtraction), because their optimal values are small. This rule is not always observed, notably in the complex function. Consequently, the performance of these functions could probably be manually improved. In the parsimonious functions, the rules are better respected; PS1 is often chosen and is always used "positively"; ES1 and RS1 are used "negatively", as expected. In addition, one of the coverage scores intervenes in each parsimonious function.

We evaluated the performance of the functions charted in Figure VIII-9, Figure VIII-10 and Figure VIII-11 using the receiver operating characteristic (ROC) curve (Baker 2003) method. ROC plots show, for varying score cut-offs, the true positive fraction (TPF) and the false positive fraction (FPF) of a scoring scheme. As cut-off, the deltaScore was used instead of the usual p-value, because of the reasons explained in Section VIII.2.3.3. Figure VIII-12 gives some details about the methodology. The true positive (TP) and false positive (FP) fractions were computed from true and random matches collected by running Popitam with the different functions on their corresponding testing sets, as described in (Colinge et al. 2003b).

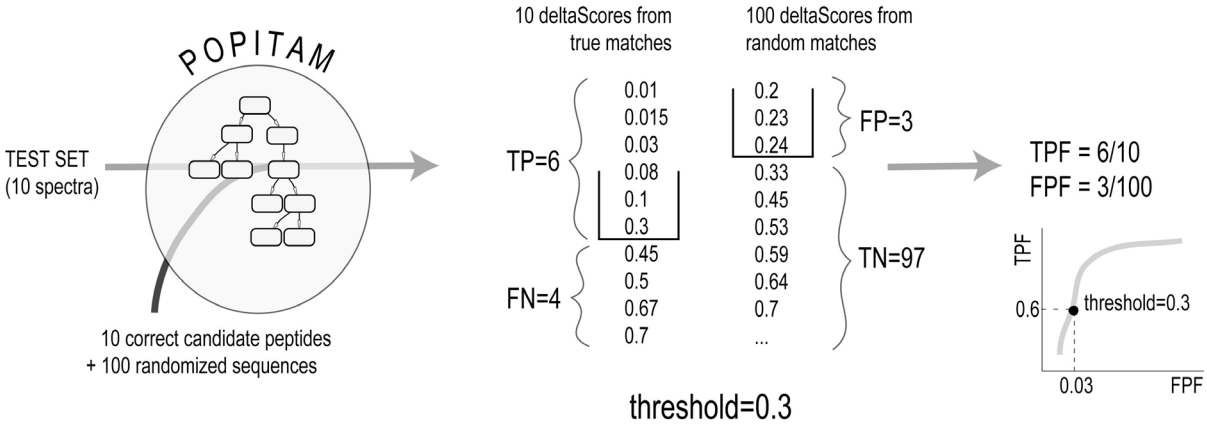



*Figure VIII-12: Example of ROC curve computation*
*Example (based on invented data) of the computation of a discrete point in a ROC curve. True and random deltaScores are collected for a given scoring function and sorted by increasing order. Then, for various threshold values, the number of true positive (TP), false negative (FN), false positive (FP) and true negative (TN) matches are counted and the corresponding TP and FP fractions are reported in the plot.*

True matches are obtained with the correct peptide sequence, and random matches are obtained with candidate peptides with a shuffled sequence. For a given deltaScore threshold, the TP fraction and FP fraction are computed according to:

$$TPF = TP/(TP+FN)$$
$$FPF = FP/(FP+TN)$$

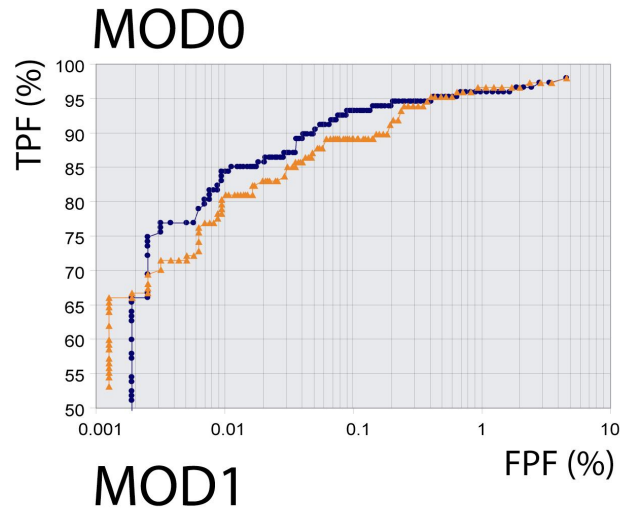
*where*

*TP is the number of true matches that are accepted by the threshold value*
*FN is the number of true matches that are rejected by the threshold value*
*FP is the number of random matches that are accepted by the threshold value*
*TN is the number of random matches that are rejected by the threshold value*

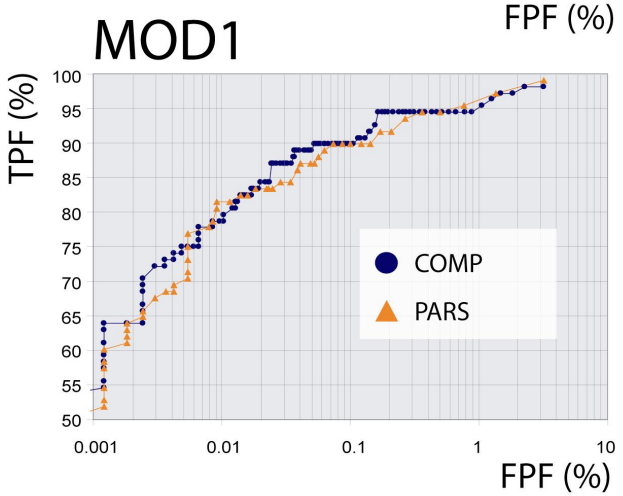
The obtained ROC curves are shown in Figure VIII-13. The TP fraction corresponds to the sensitivity of the scoring, while the FP fraction is the 1 minus the specificity. The shift of the curve towards the right when going through the three graphs expresses the growing difficulty for Popitam to identify peptides as the number of modification sites increases. The three tables show the false positive fraction and the true positive fraction (in percentage) reported for deltaScore thresholds varying from 0.1 to 0.9 with steps of 0.1.

In addition to measuring the performance of a scoring scheme, ROC curves can be used to select thresholds for accepting or rejecting candidate peptides for various TPF and FPF levels in automated identification experiments. In our case, the table can be used to give an idea about the TPF and FPF expected for a given deltaScore and scoring function. Notably, they show that the TPF and FPF associated with a given deltaScore obtained with a given scoring function can be very different from

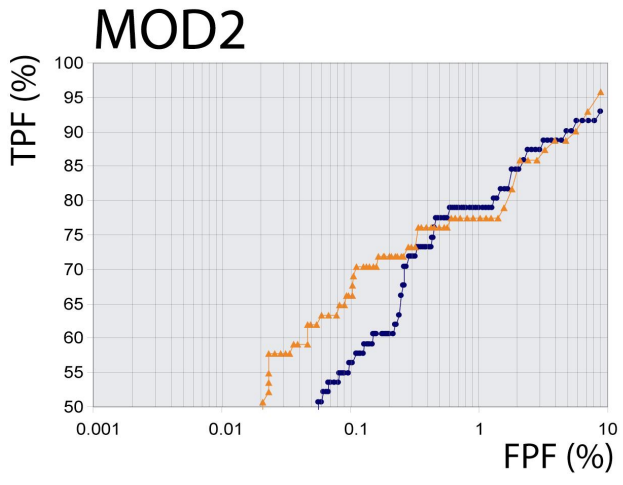
the TPF and FPF observed for the same deltaScore threshold but another scoring function. But for the moment, the decision to accept or reject a scored candidate peptide should not be taken only based on the deltaScore thresholds. The p-value, and overall the scenario and shift values proposed are of the greatest importance to assess the validity of a peptide match by Popitam.




| dS | COMPLEX | | PARSIMONIOUS | |
|---|---|---|---|---|
| | FPF(%) | TPF(%) | FPF(%) | TPF(%) |
| 0.1 | 0.0019 | 51.02 | 0.00 | 14.97 |
| 0.2 | 0.0025 | 73.46 | 0.00 | 40.14 |
| 0.3 | 0.005 | 76.87 | 0.00 | 56.46 |
| 0.4 | 0.0095 | 82.99 | 0.00 | 68.03 |
| 0.5 | 0.018 | 85.71 | 0.01 | 75.51 |
| 0.6 | 0.038 | 89.12 | 0.01 | 80.95 |
| 0.7 | 0.079 | 92.52 | 0.04 | 85.71 |
| 0.8 | 0.17 | 93.88 | 0.10 | 89.12 |
| 0.9 | 0.48 | 95.24 | 0.32 | 93.88 |




| dS | COMPLEX | | PARSIMONIOUS | |
|---|---|---|---|---|
| | FPF(%) | TPF(%) | FPF(%) | TPF(%) |
| 0.1 | 0.0006 | 20.37 | 0.0006 | 0.00 |
| 0.2 | 0.0006 | 46.30 | 0.0006 | 0.00 |
| 0.3 | 0.0012 | 57.41 | 0.0006 | 0.00 |
| 0.4 | 0.0024 | 69.44 | 0.0006 | 0.93 |
| 0.5 | 0.0066 | 76.85 | 0.0006 | 5.56 |
| 0.6 | 0.015 | 82.41 | 0.0006 | 27.78 |
| 0.7 | 0.032 | 87.04 | 0.0012 | 51.85 |
| 0.8 | 0.068 | 89.81 | 0.0036 | 68.52 |
| 0.9 | 0.21 | 94.44 | 0.028 | 84.26 |




| dS | COMPLEX | | PARSIMONIOUS | |
|---|---|---|---|---|
| | FPF(%) | TPF(%) | FPF(%) | TPF(%) |
| 0.1 | 0.01 | 18.31 | 0.0025 | 0.00 |
| 0.2 | 0.01 | 36.62 | 0.0025 | 9.86 |
| 0.3 | 0.03 | 47.89 | 0.0025 | 23.94 |
| 0.4 | 0.06 | 52.11 | 0.0077 | 35.21 |
| 0.5 | 0.09 | 54.93 | 0.018 | 46.48 |
| 0.6 | 0.18 | 60.56 | 0.025 | 57.75 |
| 0.7 | 0.32 | 71.83 | 0.067 | 63.38 |
| 0.8 | 0.61 | 78.87 | 0.16 | 71.83 |
| 0.9 | 1.68 | 81.69 | 0.56 | 76.06 |

*Figure VIII-13: ROC curves for the complex and parsimonious functions*
*The x axis scale is logarithmic. Beside each plot, a table shows the false positive fraction (FPF) and true positive fraction (TPF) observed for deltaScore thresholds varying from 0.1 to 0.9 (some of the table values are outside the plots).*

## VIII.3. Peptide identification and characterization

### VIII.3.1. Introduction

This section gives examples of Popitam's potentiality with real data and databases. In Section VIII.3.2, Popitam's operating mode is illustrated on a spectrum with two carbamidomethylations coming from a study on human nucleolar proteins (Scherl et al. 2002). In Section VIII.3.3, we show how Popitam identified spectra with PTMs, unexpected cleavage and transpeptidation in MS/MS data from a study on transpeptidation (Schaefer et al. 2005). All identification runs were performed on a bi-processor AMD Athlon MP 2000+ (1.4 GHz).

### VIII.3.2. Identification and characterization of a peptide with two modifications

As a first example, we illustrate how Popitam can deal with two modifications located on different parts of a same peptide. For this aim, we used the spectrum of the peptide VFNC[cam]ISYSPLC[cam]K (Scherl et al. 2002). No hint about the place, number or type of modifications was given to Popitam, except that we were looking for any modifications in a range of –100 to 200 Daltons. The parameters were set as follows:

| | |
|---|---|
| PM RANGE: -100 to 200 | COVBIN: 9 |
| FRAGMENT_ERROR1: 0.4 Da | MIN_TAG_LENGTH: 3 |
| FRAGMENT_ERROR2: 0.6 Da | MAX_ADD_MOD: 200 |
| MIN_COV_ARR: 0.3 | MAX_LOSS_MOD: 100 |

We performed two independent runs: the first one with MODGAPS set to 1, and the second one with MODGAPS set to 2. The spectrum and its spectrum graph (with the highest-scoring run-and-jump path found by Popitam) are charted in Figure VIII-14. Popitam's outputs are summarized in Table VIII-4.

When run with MODGAPS set to 1, the scores are quite similar for all high-ranked candidate peptides and the associated deltaScores are consequently all bad. In such a situation, it can be deduced that none of the peptides fits the spectrum better than any other, given the number of *modGaps* to include in the scenarios. The p-values are quite good, but they are not relevant (see Section VIII.2.3.3).
When Popitam searches for scenarios with two *modGaps*, the deltaScore of the second candidate stands out because it is significantly smaller (and thus better) than the others. Actually, both first and second candidate peptides have similar scores and actually correspond to the same peptide (except that the second one has an additional amino acid due to a missed-cleavage).
Concerning the characterization of the peptide, the best-scoring scenario (for *modGaps*=2) indicates that a mass of 57.05 Da should be added to the first cysteine and a mass of 57.04 Da should be added either to the second cysteine, the terminal lysine or the C-terminus group. As we know that carbamidomethylation was used to break disulfide bridges during sample preparation, and that such a modification results in a mass shift of 57.02, the evident conclusion is that both cysteines are carbamidomethylated. To best fit the data, the second ranked peptide (VFNCISYSPLCKR) should have its first and second cysteine carbamidomethylated (+ 57.05 ; +57.04) and lose its last amino acid (-156.10) (this represents three modification events). The -99.06 shift of the second *modGap* is

explained by the second carbamidomethylation (57.04) minus the molecular weight of arginine (156.10). Although this scenario is theoretically possible, preference is given to the less intricate one (parsimonious principle).

Presently, Popitam does not propose automated explanations for *modGaps*, which have to be done manually. But we expect to add an interpretation module in future developments.

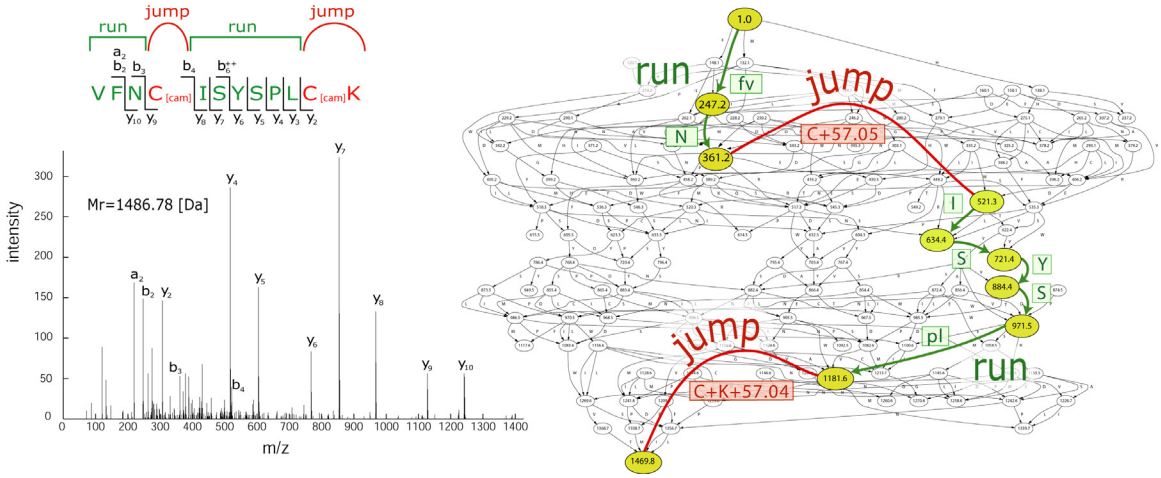



*Figure VIII-14: A spectrum and its spectrum graph*

*Spectrum of peptide VFNC[cam]ISYSPLC[cam]K and the run-and-jump path corresponding to the best scenario found by Popitam (to facilitate the representation of the spectrum graph, only simple edges are drawn except for edges belonging to the run-and-jump path).*

```
Scoring functions: parsimonious (Figure VIII-10 and Figure VIII-11)
Database:          Swiss-Prot
Taxonomy:          Homo sapiens
Spectrum:          744.39 (2+) (Q-TOF)
Graph:             128 nodes, 301 + 3768 edges
```

**MODGAPS: 1**

AC filter: NO
Peptides (digested): 2299426
Peptides (candidates): 121362
Scenarios (evaluated): 10995
Running time: 493 (8mn)

| rank | score (dS;PVAL) | shifts | Scenario |
|---|---|---|---|
| 1 | 130.16<br>(1.00; <1e-12) | 178.08 | NEVLHISRGER<br>***LhiSR--- |
| 2 | 129.60<br>(0.84; <1e-12) | 133.08 | INPLTGEIELKK<br>*NP-----ELkk |
| 3 | 108.72<br>(0.90; <1e-12) | 12.06 | MDQPEAPCSSTGPR<br>MD******SS---- |

**MODGAPS: 2**

AC filter: NO
Peptides (digested): 2299426
Peptides (candidates): 121362
Scenarios (evaluated): 724897
Running time: 848s (14mn)

| rank | score (dS ;PVAL) | shifts | Scenario |
|---|---|---|---|
| **1** | **3.24e05**<br>**(0.87; <1e-12)** | **57.05**<br>**57.04** | **VFNCISYSPLCK**<br>**vfN*ISYSpl**** |
| **2** | **2.81e05**<br>**(0.44; <1e-12)** | **57.05**<br>**-99.06** | **VFNCISYSPLCKR**<br>**vfN*ISYSpl***** |
| 3 | 1.23e05<br>(0.84; <1e-12) | -26.95<br>5.07 | MDGGRTCSCSSFCR<br>MDggRtc**SS*** |

*Table VIII-4: Popitam's output for spectrum of Figure VIII-14*

*The left output was obtained with MODGAPS set to 1 and the right output was obtained with MODGAPS set to 2). The score column contains Popitam's score and the associated deltaScore (dS) and p-value (PVAL) (the p-value was computed from the negative score). The shifts column gives the mass of the modGap(s).*

### VIII.3.3. Concrete examples of identification and characterization

We illustrate here the potentiality of our method on a set of 1248 MS/MS spectra from a study on transpeptidation (Schaefer et al. 2005). This study focuses on the murine Alpha crystallin A chain protein (P24622). The protein was extracted from mice eye lens, isolated by 2-D gel electrophoresis and analyzed with an ion trap mass spectrometer. Its function is not well known, although it is annotated in Swiss-Prot as possibly contributing to the transparency and refractive index of the lens. It can be found in two forms due to alternative splicing. The 196 amino acid variant represents a minor form while the 173 variant (represented below) is the major form.

```
         10          20          30          40          50          60
          |           |           |           |           |           |
MDVTIQHPWF KRALGPFYPS RLFDQFFGEG LFEYDLLPFL SSTISPYYRQ SLFRTVLDSG

         70          80          90         100         110         120
          |           |           |           |           |           |
ISEVRSDRDK FVIFLDVKHF SPEDLTVKVL EDFVEIHGKH NERQDDHGYI SREFHRRYRL

        130         140         150         160         170
          |           |           |           |           |
PSNVDQSALS CSLSADGMLT FSGPKVQSGL DAGHSERAIP VSREEKPSSA PSS
```

As the spectra were obtained with an ion trap mass spectrometer, Popitam was in a tricky situation. First, no ion probability file was available for ion traps; then the scoring functions were not specifically adapted since they were trained with Q-TOF data; and finally, both fragment errors had to be significantly increased due to the lower accuracy of these spectrometers compared to Q-TOFs.
Popitam's parameters were set as follows:

| | |
|---|---|
| MODGAPS: 1<br>PM RANGE: -100 to 400<br>FRAGMENT_ERROR1: 0.8 Da<br>FRAGMENT_ERROR2: 1.5 Da<br>MIN_COV_ARR: 0.3 | COVBIN: 9<br>MIN_TAG_LENGTH: 3<br>MAX_ADD_MOD: 400<br>MAX_LOSS_MOD: 100 |

Analyzing the complete set of spectra with Popitam was difficult because of the important processing time needed per spectrum. Fortunately, all the acquired spectra corresponded to a single protein. We therefore performed a first run with Popitam on a subset of 839 doubly and triply charged spectra using the correct AC (P24622) as a filter. With this filter, only peptides from the correct candidate protein were presented to Popitam. The run took 77 minutes. The aim was to quickly spot spectra that were likely to produce interesting results for illustration purpose. Using these preliminary results, we spotted high-scoring scenarios (with one *modGap*) and isolated the corresponding spectra. Two independent identification runs were then performed on the spectra against the mouse database (Swiss-Prot). In the first run, an AC filter of 33 proteins (including the correct one) was used while the second run was performed without AC filter. In each output, the number of candidate peptides presented to Popitam is given. When the AC filter is applied, their average number is 810. When no AC filter is applied, it is 169'437. Popitam's scenarios were finally manually interpreted and validated with Phenyx. Amino acid replacements were simulated in Phenyx by adding user-defined modifications to the pool of available ones. For example, the exchange of a Q (128.05 Da) by a P (97.12 Da) was simulated by creating a +30.93 modification occurring on P amino acids.

### VIII.3.3.1. Identification of N-acetylated peptides

This first example shows the identification and characterization of two peptides with "unexpected" modifications. The spectra are represented in Figure VIII-15. Both of them correspond to the peptide sequence MDVTIQHPWFK. For spectrum A, Popitam reports a one amino acid *modGap* of 42.50 (Table VIII-5). The interpretation of this shift is obvious, since the protein is actually annotated in Swiss-Prot as being N-acetylated (see Figure II-8), which corresponds to a modification weight of 42.01 Da.

For spectrum B, Popitam announces a shift of 58.60 (Table VIII-6). Such a value can be attributed to two modification events: a N-acetylation plus a methionine oxidation. This case illustrates that one *modGap* may correspond to several modification events (output B). Popitam's results were confirmed by Phenyx (Phenyx parameters were set as follows: taxonomy=Eukaryote Swiss-Prot and TrEMBL; scoring=esquire 3000+-1.0 Da; variable modifications=N-terminal acetylation and methionine oxidation).

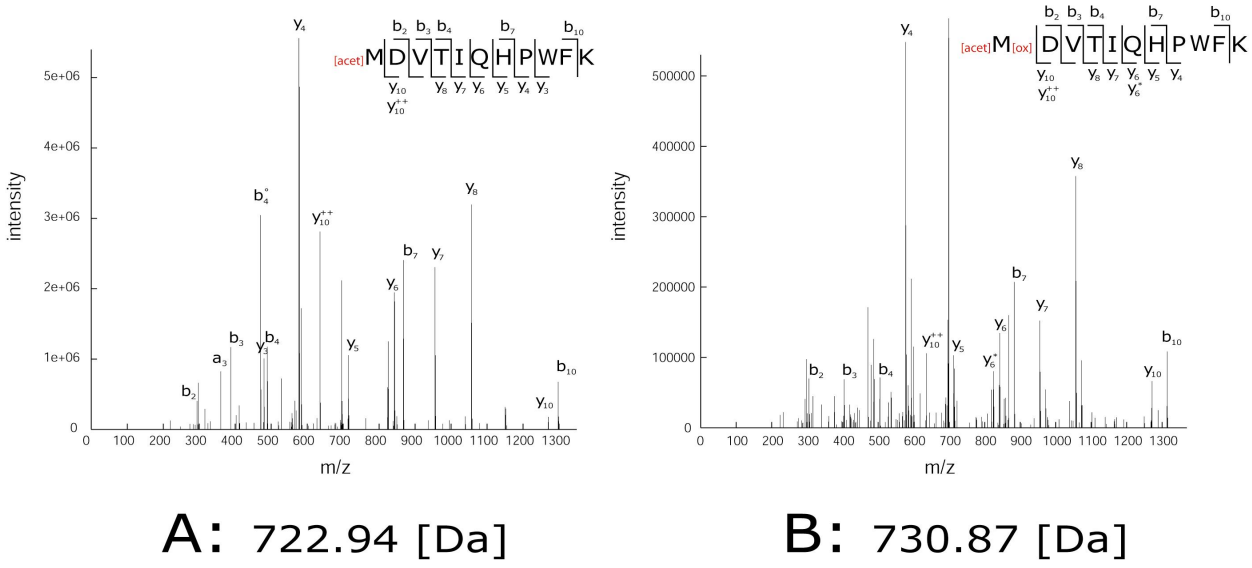



*Figure VIII-15: Two spectra of N-acetylated peptides*

**A**

```
Scoring functions: complex (Figure VIII-10)
Database:          Swiss-Prot
Taxonomy:          Mus musculus
Spectrum:          722.94 (2+) (ION TRAP)
Graph:             128 nodes, 658 + 8818 edges
```

```
AC filter: YES (P24622 + 32 ACs)
Peptides (digested):   7178
Peptides (candidates):  761
Scenarios (evaluated):  921
Running time:            31 sec
```

| rank | score (deltaS; PVAL) | shifts | Scenario |
|---|---|---|---|
| **1** | **507.79 (0.35; <1.00e-12)** | **42.50** | **MDVTIQHPWFK *DVTIqhPwfK** |
| 2 | 179.03 (0.34; 1.09e-02) | -52.40 | EVLVTSRSSGTFSK ***VTS-------- |
| 3 | 61.35 (0.82; 1.00e00) | 98.20 | WGVFSGRTPPSR ---FSGR***** |

```
AC filter: NO
Peptides (digested):   1401817
Peptides (candidates):  158955
Scenarios (evaluated):  198130
Running time:                81 min
```

| rank | score (deltaS; PVAL) | shifts | Scenario |
|---|---|---|---|
| **1** | **507.79 (0.62; <1e-12)** | **42.50** | **MDVTIQHPWFK *DVTIqhPwfK** |
| 2 | 311.28 (0.91; <1e-12) | 259.92 | KTSGLVSLHSR *TSGL------ |
| 3 | 282.95 (0.95; <1e-12) | 259.92 | MDAATLTYDTLR ----TLtyDT** |

*Table VIII-5: Popitam's outputs for spectrum A of Figure VIII-15*
*In the left output, a tight AC filter was set. This results in less candidate peptides, and thus in smaller computing time and fewer high-scoring random matches (random matches are candidate peptides that receive high scores by chance). The best-scoring scenarios map the modification either on the N-terminus group or the initial methionine. The 42.50 shift corresponds to an N-acetylation.*

**B**

```
Scoring functions: complex (Figure VIII-10)
Database:          Swiss-Prot
Taxonomy:          Mus musculus
Spectrum:          730.87 (2+) (ION TRAP)
Graph:             128 nodes, 650 + 9049 edges
```

```
MODGAPS:     1
AC filter: YES (P24622 + 32 ACs)
Peptides (digested):   7178
Peptides (candidates):  761
Scenarios (evaluated):  987
Running time:            60 sec
```

| rank | score (deltaS; PVAL) | shifts | Scenario |
|---|---|---|---|
| **1** | **821.27 (0.09;<1.00e-12)** | **58.60** | **MDVTIQHPWFK *DVTIQHPwfK** |
| 2 | 74.31 (0.95; 2.96e-11) | 208.96 | TLGPANLPLAQR ****anLPlaqr |
| 3 | 70.29 (0.97; 9.64e-10) | 376.76 | SLWPQIKGR **WPQ---- |

```
MODGAPS:    1
AC filter: NO
Peptides (digested):   1401817
Peptides (candidates):  156975
Scenarios (evaluated):  207924
Running time:               86 min
```

| rank | score (deltaS; PVAL) | shifts | Scenario |
|---|---|---|---|
| **1** | **821.27 (0.34;<1.00e-12)** | **58.60** | **MDVTIQHPWFK *DVTIQHPwfK** |
| 2 | 275.47 (0.92;<1.00e-12) | 288.75 | NSSGILLVALGK *ssGILLVA--- |
| 3 | 252.60 (0.88;<1.00e-12) | 23.05 | SHSSGVGKPLSPER *****VGKPLsper |

*Table VIII-6: Popitam's outputs for spectrum B of Figure VIII-15*
*The +58.60 modGap corresponds to two modification events: an N-acetylation (+42.01) plus a methionine oxidation (+16.01).*

### VIII.3.3.2. Identification of half-cleaved peptides

Usually, trypsin cleaves peptides at C-terminal side of K and R amino acids. Therefore, any peptide (except for the first peptide of the protein) should follow a K or a R on the protein sequence. Similarly, any peptide (except for the last peptide of the protein) should end with a K or a R. When one of these two rules is not met, the peptide is said "half-cleaved". PFF methods can identify half-cleavage peptides if they are parameterized to do so (in this case, all possible half-cleaved peptides are generated and matched to the spectrum). In Popitam, half-cleavage is one of the numerous unexpected events that can occur to a peptide sequence and cause a shift between the peak pattern and the candidate peptide. The mass difference between the half-cleaved peptide of the spectrum and the tryptic peptide of the database is reported as a *modGap* and it is mapped either on the first amino acid(s) or the last amino acid(s) of the database peptide. Of course, if the mass difference is larger than the allowed modification range, the scenario is not scored and identification fails (see Section VI.9). Figure VIII-16 shows a spectrum of the tryptic peptide VLEDFEVEIHGK (spectrum A) and of a half-cleaved peptide LEDFEVEIHGK (spectrum B). Popitam's outputs for spectrum B are summarized in Table VIII-7. In both runs, Popitam suggests a *modGap* of 98.30 Da in the N-terminal or first amino acid of the candidate peptide. This corresponds to the removal of the amino acid valine.

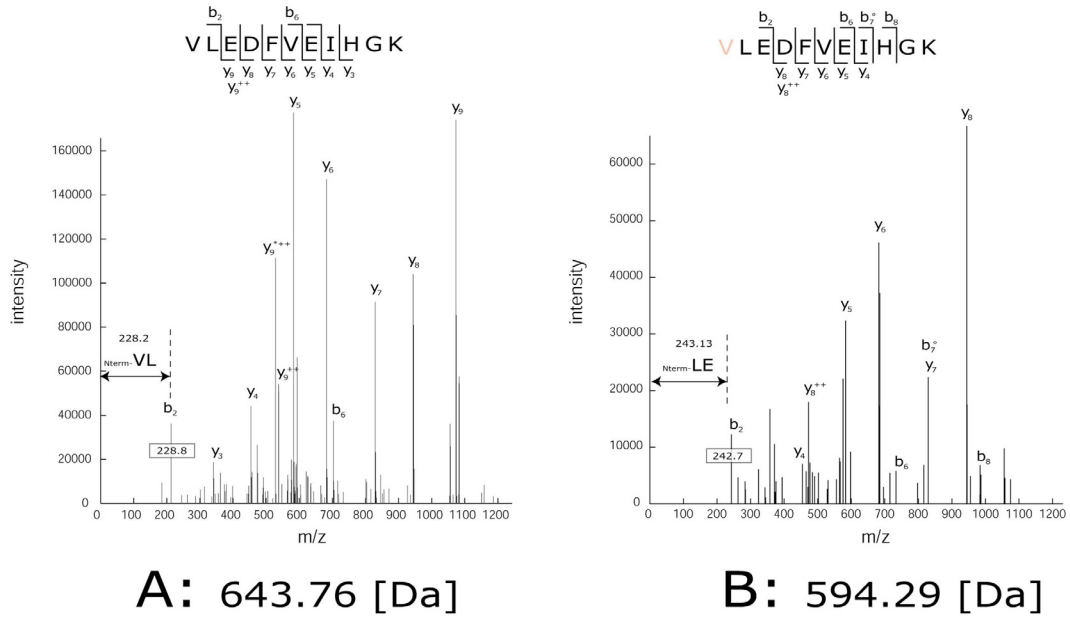



*Figure VIII-16: Annotated spectra of a tryptic peptide VLEDFVEIHGK (A) and of a half-cleaved peptide LEDFVEIHGK (B).The b2 ion in spectrum A (m/z 212.8 Da) corresponds to the fragment NtermVL, while the b2 ion in spectrum B (m/z 242.7 Da) corresponds to the fragment N-termLE.*

| B | Scoring functions: complex (Figure VIII-10)<br>Database: Swiss-Prot<br>Taxonomy: Mus musculus<br>Spectrum: 594.29 (2+) (ION TRAP)<br>Graph: 101 nodes, 609 + 5852 edges |
|---|---|

MODGAPS: 1
AC filter: YES (P24622 + 32 ACs)
Peptides (digested): 7178
Peptides (candidates): 867
Scenarios (evaluated): 1229
Running time: 17 sec

| rank | score (dS;PVAL) | shifts | Scenario |
|---|---|---|---|
| **1** | **234.98 (0.26;<1.00e-12)** | **-98.30** | **VLEDFVEIHGK *LEDFVEIHgk** |
| 2 | 61.01 (0.89; 5.85e-10) | 68.20 | NQAMADALER NQ-----LE* |
| 3 | 54.57 (0.90; 1.81e-07) | 91.20 | DKDDELSFK DKDDEL*** |
| 4 | 49.16 (0.91; 1.16e-05) | -71.70 | EWDLKPMADR **DLKpm--- |

MODGAPS: 1
AC filter: NO
Peptides (digested): 1401817
Peptides (candidates): 186674
Scenarios (evaluated): 257568
Running time: 43 min

| rank | score (dS;PVAL) | shifts | Scenario |
|---|---|---|---|
| **1** | **234.98 (0.89; <1e-12)** | **-98.30** | **VLEDFVEIHGK *LEDFVEIHgk** |
| 2 | 208.51 (0.94; <1e-12) | 211.11 | DEDFVKPK DEDFVkp* |
| 3 | 196.61 (0.96; <1e-12) | 95.11 | NEDFVEIAR NEDFVEI** |
| 4 | 189.56 (0.89; <1e-12) | -32.99 | NEDFVEIARK NEDFVEI*** |

*Table VIII-7: Popitam's outputs for spectrum B of Figure VIII-16*

### VIII.3.3.3. Identification of transpeptidated peptides

Transpeptidation was introduced in Section IV.3.1 as the grafting of a peptide fragment on another one. It is not proven that transpeptidation naturally occurs in a protein lifetime. According to Schaefer et al. (Schaefer et al. 2005), it probably occurs during sample preparation as a side activity of trypsin. Popitam identified several cases of transpeptidation in the data given by Schaefer and colleagues. This section presents three examples. Interestingly, while the first two were reported in Schaefer's publication, the third one was not.

The first example shows the identification of a spectrum corresponding to the transpeptided peptide R+VLEDFVEIHGK. When the peptide VLEDFVEIHGK is compared to the spectrum, Popitam suggests a 156.66 *modGap* mapped on the initial part of the peptide (see Table VIII-8). This shift can be explained by the addition of a R at its N-terminus. For unknown reasons, the scoring function did not give a higher score although the spectrum fragmentation is good. Consequently, the correct candidate peptide bluntly falls to the seventh position when the search is performed against the complete mouse database. Importantly, although the first ranked peptide's scenario seems of good quality, the similar values of the scores suggest that Popitam didn't succeed in identifying the correct peptide in this very case. Two indices play in favor of the seventh peptide: first its scenario is as or even more complete as the preceding ones, with many single edges; second, the reported shift can be explained by a reasonable hypothesis (addition of an arginine as the result of transpeptidation). The same result was reported in Schaefer's publication, and confirmed using Phenyx (user variable modification: +156.19 Da on valine).

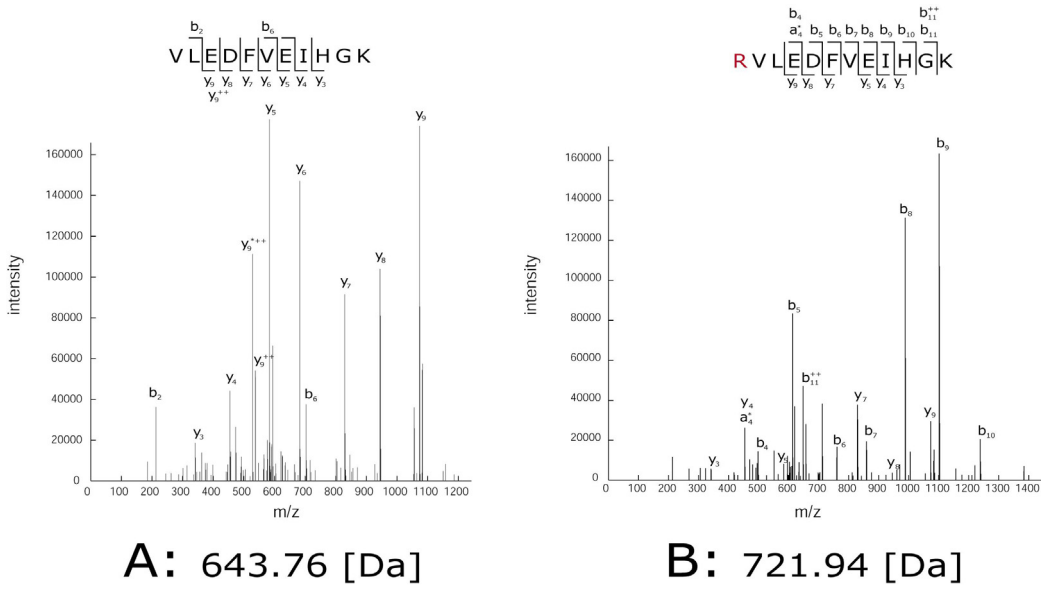



*Figure VIII-17: Annotated spectra of peptide VLEDFVEIHGK (A) and of the transpeptidated peptide RVLEDFVEIHGK (B).*
*The more complete b-ion series in spectrum B suggests the presence of a basic amino acid among the first amino acids of the peptides.*

B

```
Scoring functions: complex (Figure VIII-10)
Database:          Swiss-Prot
Taxonomy:          Mus musculus
Spectrum:          721.94 (2+) (ION TRAP)
Graph:             128 nodes, 663 + 7238 edges
```

```
MODGAPS:    1
AC filter: YES (P24622 + 32 ACs)
Peptides (digested):   7178
Peptides (candidates):  829
Scenarios (evaluated):  666
Running time:            22 sec
```

| rank | score (deltaS) | shifts | Scenario |
|---|---|---|---|
| **1** | **58.80 (0.48; <1e-12)** | **156.66** | **VLEDFVEIHGK** ****EDFveIHGK** |
| 2 | 28.16 (0.51; <1e-12) | 101.29 | KVLLTCHDDAAR ---LtcH***** |
| 3 | 14.36 (0.94; 1.61e-03) | 194.0 | FIEGEVVSALGK *IEGE------- |
| 4 | 13.50 (0.93; 6.49e-03) | 77.83 | LQNLDRAVLPPK -----RA*LppK |
| 5 | 12.51 (0.92; 2.84e-02) | -22.17 | RALIESYQNLTR ---IE******* |
| 6 | 11.48 (0.97; 1.09e-01) | 305.83 | FEVIIKMQK *evIIK--- |
| 7 | 10.78 (0.95; 2.41e-01) | 170.63 | NGLEDGYGEYR **Ledg----- |

```
MODGAPS:    1
AC filter: NO
Peptides (digested):   1401817
Peptides (candidates):  159150
Scenarios (evaluated):  156654
Running time:                64 min
```

| rank | score (deltaS) | shifts | Scenario |
|---|---|---|---|
| 1 | 96.70 (0.87;<1.00e-12) | -98.29 | WRLLYEELYEK WrLlyEELY** |
| 2 | 84.32 (0.84;<1.00e-12) | 24.45 | LNGGLGTSMGCKGPK ----Lgt—GCK*** |
| 3 | 71.23 (0.89;<1.00e-12) | 193.61 | LGGDLGTYVINK ----LgtyvIN* |
| 4 | 63.34 (0.98;<1.00e-12) | 37.60 | GQRDLYSGLNQR ---DlysgLN** |
| 5 | 62.29 (0.97;<1.00e-12) | 95.41 | DGTVQLGDFGIAR ----QlgdfgI** |
| 6 | 60.61 (0.97;<1.00e-12) | -21.49 | FGHLVKCSMVNTK FGHLvkCSM**** |
| **7** | **58.80 (0.93;<1.00e-12)** | **156.66** | **VLEDFVEIHGK** ****EDFveIHGK** |

*Table VIII-8: Popitam's outputs for spectrum B of Figure VIII-17*
*The modGap of 156.66 can be explained by the N-terminal addition of an arginine residue.*

The second example shows a successful and confident identification of the spectrum charted in Figure VIII-18 (B) corresponding to the transpeptided peptide (PR)+~~VQ~~SGLDAGHSER. The scenario proposes a 26.05 [Da] *modGap* mapped on the two first amino acids or N-terminus of the peptide (see Table VIII-9). Such a gap can be explained by the exchange of VQ by PR (or RP). The identification was reported in Schaefer's publication, was consistent with the spectrum interpretation (enhanced b-ion series, indicating the presence of a basic amino acid at the beginning of the peptide) and was confirmed using Phenyx with user defined variable modifications.

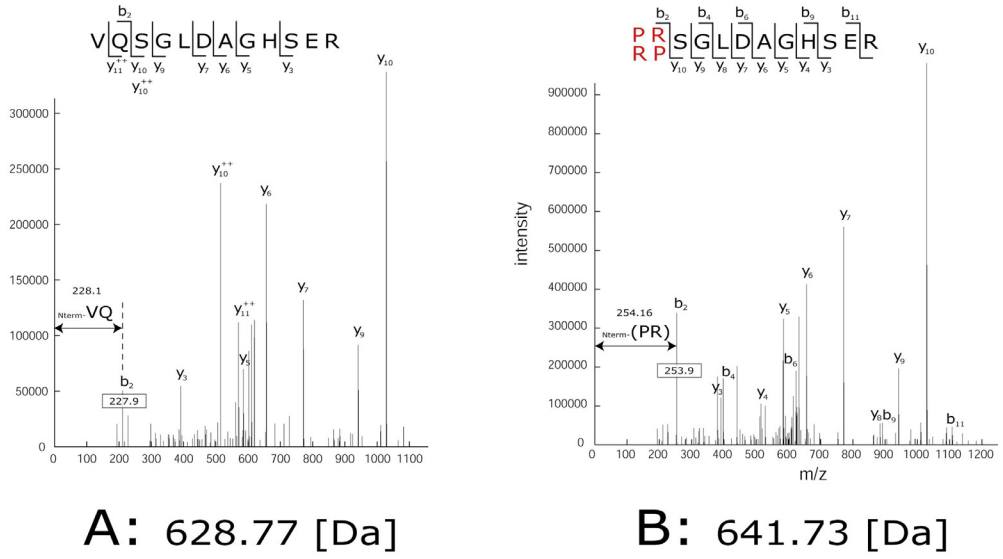



*Figure VIII-18: Annotated spectra of the tryptic peptide VQSGLDAGHSR (A) and of the transpeptidated peptide (PR)SGLDAGHSER (B).*

| | |
|---|---|
| B | Scoring functions: complex (Figure VIII-10)<br>Database: Swiss-Prot<br>Taxonomy: Mus musculus<br>Spectrum: 641.73 (2+) (ION TRAP)<br>Graph: 110 nodes, 574 + 7012 edges |

MODGAPS: 1
AC filter: YES (P24622 + 32 ACs)
Peptides (digested): 7178
Peptides (candidates): 817
Scenarios (evaluated): 686
Running time: 15 sec

| rank | score (deltaS) | shifts | Scenario |
|---|---|---|---|
| **1** | **415.54 (0.14;<1.00e-12)** | **26.54** | **VQSGLDAGHSER<br>**SGLDAGH---** |
| 2 | 56.60 (0.52;<1.00e-12) | 236.76 | LVELGRSSGK<br>LvelgRSS** |
| 3 | 29.28 (0.61; 1.01e-06) | 227.25 | AALEQRGLAK<br>---eqR**** |

MODGAPS: 1
AC filter: NO
Peptides (digested): 1401817
Peptides (candidates): 176641
Scenarios (evaluated): 146667
Running time: 37 min

| rank | score (deltaS) | shifts | Scenario |
|---|---|---|---|
| **1** | **415.54 (0.28;<1.00e-12)** | **26.54** | **VQSGLDAGHSER<br>**SGLDAGH---** |
| 2 | 117.48 (0.89;<1.00e-12) | 166.25 | VRVSADAMLR<br>vrvsADAml* |
| 3 | 104.93 (0.90;<1.00e-12) | 291.67 | HDSGLDSMK<br>--SGLD*** |

*Table VIII-9: Popitam's output for spectrum B of Figure VIII-18*
*The 26.54 Da modGap can be explained by the exchange of amino acids VQ by amino acids (PR).*

In the last example, Popitam matches the spectrum charted in Figure VIII-19 (B) against the peptide VQSGLDAGHSER and reports a 11.92 Da *modGap* mapped somewhere between the N-terminus and the second amino acid (see Table VIII-10). Although Schaefer et al. did not report this transpeptidation event in their publication, we are confident in our result: the deltaScores are low in both outputs; the shift can be explained by a two-amino acid exchange (VQ against HT) (with a 0.94 Da error); spectrum B has enhanced b-ions compared to the unmodified spectrum, which indicates the presence of a basic amino acid at the beginning of the peptide (in our case H); and finally, the match was confirmed using Phenyx with user defined variable modifications.

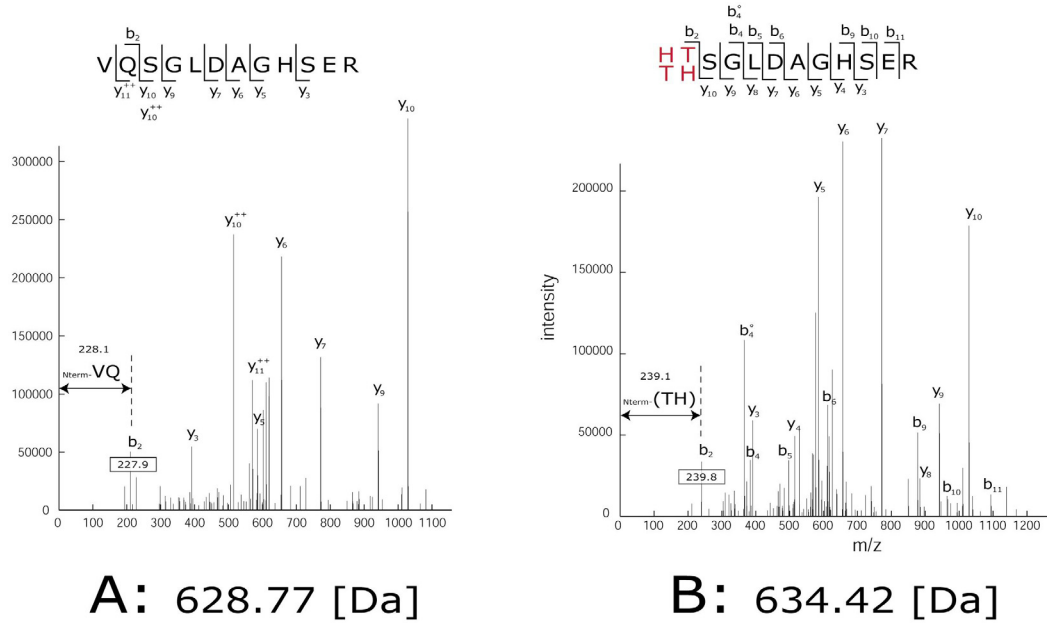



*Figure VIII-19: Annotated spectra of the tryptic peptide VQSGLDAGHSR (A) and of a transpeptidated peptide (HT)SGLDAGHSER (B).*

| B | Scoring functions: complex (Figure VIII-10)<br>Database: Swiss-Prot<br>Taxonomy: Mus musculus<br>Spectrum: 634.42 (2+) (ION TRAP)<br>Graph: 110 nodes, 491 + 7385 edges |
|---|---|

MODGAPS: 1
AC filter: YES (P24622 + 32 ACs)
Peptides (digested): 7178
Peptides (candidates): 829
Scenarios (evaluated): 666
Running time: 13 sec

| rank | score (deltaS) | shifts | Scenario |
|---|---|---|---|
| **1** | **651.93 (0.19;<1.00e-12)** | **11.92** | **VQSGLDAGHSER<br>**SGLDagH---** |
| 2 | 122.24 (0.80;<1.00e-12) | 108.10 | GESDDEKPRK<br>***DDEK--- |
| 3 | 98.18 (0.93;<5.79e-11) | 108.10 | HDGTSNGTAR<br>---TSNgt** |
| 3 | 91.75 (0.53;<3.06e-09) | 113.47 | LVLGDNSPAIR<br>---GDN***** |

MODGAPSS: 1
AC filter: NO
Peptides (digested): 1401817
Peptides (candidates): 178232
Scenarios (evaluated): 140983
Running time: 39 min

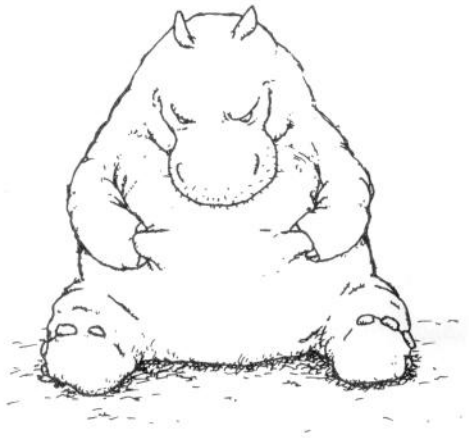
| rank | score (deltaS) | shifts | Scenario |
|---|---|---|---|
| **1** | **651.93 (0.40; <1e-12)** | **11.92** | **VQSGLDAGHSER<br>**SGLDagH---** |
| 2 | 257.75 (0.96; <1e-12) | 197.60 | WINQHLMK<br>*INQH--- |
| 3 | 247.81 (0.98; <1e-12) | 28.26 | VLAALDELGLAR<br>**aaLDE----- |
| 4 | 243.16 (0.97; <1e-12) | 110.81 | VLSSLKHLCR<br>**SS-KH--- |

*Table VIII-10: Popitam's outputs for the spectrumB of Figure VIII-19*
*The modGap of 11.92 can be explain by the exchange of amino acids VQ by amino acids (HT).*

The various results shown in this section highlight potential applications of Popitam. We showed that its use is not limited to post-translationally modified or mutated peptides, but can be extended to other phenomena, such as transpeptidation and non-specific cleavages. The difficult process of scoring has been detailed. In particular, a function aimed at scoring all types of scenario is presently not available. Also we have problems in attributing a significance level to the scores, and thus in automating acceptance or rejection of a match.

It would have been most interesting to compare Popitam's results with other "open-modification search" algorithms. Unfortunately, at present there are many few such algorithms (see Section V.4). Among those, PEDANTA and OpenSea are not publicly available, while GutenTag was unstable on our machines. However, the comparison with GutenTag is of less interest, as this tool is based on efficient filtering rather than a scoring taking into account shifts between peaks.

ぼくはカバであることに満足している、
たったひとつのことを除いては。
ぼくはほんとはもっと痩せたいのだ、せめてあと一トンくらい。



Moi, je suis très heureux d'être un hippopotame.
A part une chose: je voudrais maigrir.
Franchement, une tonne minimum.

I am very happy to be a hippopotamus.
Apart one think: I wish to loose weight.
Frankly, at least one ton.

Figure adapted from "Monsieur Hippopotame" by Tanikawa Shuntarô

# CHAPTER IX

# PERSPECTIVES AND CONCLUSION

# IX. Perspectives and conclusion

## IX.1. Perspectives

Future developments of our method should focus on the following points:

1) code optimization
The development of Popitam took several years of research. During these years, emerging ideas were rapidly implemented, tested and eventually kept, transformed or discarded. As a result, less importance has been put on code optimization. Re-thinking some structures and redesigning some functions will undoubtedly result in a significant gain in computing time.

2) result list management
A future and necessary optimization of Popitam will concern the management of the result list. During the analysis of a candidate peptide, scenarios with any number of *modGap* are produced. Most of them are discarded, because they do not contain the correct *modGap* number according to the parameter MODGAPNB. Consequently, if the user wishes to test scenarios with 0, 1 or 2 *modGaps*, Popitam must be launched three times. The next version of Popitam will score the scenarios independently of the MODGAPNB parameter, and produce several result lists. In a first stage, the result lists will be outputted without further analysis; in a second stage, a procedure to compare the different scores produced by scenarios with 0, 1 or 2 *modGaps* will be implemented.

3) implementing supplementary filters
In Popitam, the comparison between the spectrum and a database sequence comprises three steps: a) tag extraction, b) clique search and c) scenario building. Such an analysis is performed for each candidate peptide. Consequently, when run against large databases, the identification procedure requires too long a computing time, especially if the precursor mass filter is relaxed. In such a case, a sequence-based filter would be an appropriate mean to reduce the number of analyzed peptides.

4) scoring
One of the challenges of "open-modification search" methods is to score peptides of various sizes. The larger the range of modifications to take into account, the more the size of the candidate peptides may differ. At the moment, some of Popitam's subscores favor small peptides (e.g. the coverage scores), while others favor large peptides (e.g. the series scores). A new scoring based on log-odd ratios would probably lead to interesting results.

5) result's interpretation
Popitam proposes p-values based on score distributions of negative or random peptides. Unfortunately, most often, extreme scores contaminate the distribution and too many negative peptides receive p-values near 0. Consequently, the confidence of a peptide assignment remains a subjective interpretation based on the distributions of high-scoring peptides and on the proposed scenarios. We should establish new criteria to separate assignments that are likely to be correct from assignments that are likely to be incorrect.

6) shift interpretation module

Last but not least, a shift interpretation module has to be implemented. At present, Popitam reports a list of shifts and their location on the peptide sequence. In a near future, it should be able to propose, when possible, explanations for the shifts, such as amino acid(s) replacements, known PTMs, chemical modifications due to sample preparation, putative transpeptidations, and so on.

## IX.2. Conclusion

In this thesis, we charted the general context of MS/MS protein identification and expounded some of the issues faced by identification algorithms. We more specifically focused on "open-modification search", which represents a particular challenge for peptide identification and characterization. Most MS/MS identification algorithms require the preselection of a number of possible modifications to be taken into account during the comparison of the spectrum with database sequences. "Open-modification search" algorithms are designed to take into account any type of modification that would allow a better match between the spectrum peak pattern and a candidate peptide. We proposed a new and original method for "open-modification search", called Popitam. By correlating MS/MS spectra with candidate peptides through a spectrum graph structure, Popitam is to our knowledge the first algorithm that combines a *de novo* sequencing and a PFF approach. Other original features of Popitam are: a) the sequence-guided tag extraction performed by the simultaneous parsing of the spectrum graph and of an indexed representation of a candidate sequence, and b) the tag compatibility graph in which cliques represent possible interpretation scenarios of the spectrum.

The large number of scenarios to be tested, notably when searching candidate peptides with more than one modification events, led us to devote a significant part of our work to optimizing scenario-scoring functions using parallel multi-objective Genetic Programming. This approach gave interesting results. Sixty promising functions were selected among a set of more than 800 co-dominating functions reported by the GP process. Using a procedure of repeated sampling and cross-validation, we showed that Genetic Programming is a robust method. Only one of the learned functions was obviously overfitted. We also demonstrated the superiority of the learned functions in terms of rank-based performance over an empiric and a basic function. Nevertheless, this does not mean that it is not possible to design empirical functions that may surpass the learned ones. An interesting feature of Popitam is that it is possible to define one's own scoring functions, as long as they are in an appropriate format and based on the list of Popitam's subscores. By this mean, new scoring functions can be freely tested using the web interface. Defining subscores is currently not possible but could be considered in future developments.

We finally showed that Popitam is a suitable peptide identification and characterization method, not only in case of post-translational modifications or mutations, but also in case of missed- or half-cleavages and transpeptidation. This list can be extended to other events, like errors in databases, badly calibrated data or imprecise precursor masses, and in a general way, in case of any event that modifies the sequence or residue masses of a peptide analyzed by mass spectrometry. Popitam has been implemented in the C++ language and has been made available through a web interface.

Before concluding this work, we would like to stress that Popitam should not be confused with a high-throughput MS/MS identification method. The use of a classic PFF approach or an "open-

modification search" method is a question of strategy. The different methods complement each other rather than being in competition. Popitam in its present form is not adapted to be used on a large number of spectra, neither against large databases. Neither is it adapted to identify spectra with poor peak statistics. Its use should be parsimonious, on specific good quality but recalcitrant spectra, and against targeted protein lists. Popitam could find its place in an expert identification system or as an additional round of a PFF identification tool. Convinced that improvement in MS/MS identification can arise from a judicious use and combination of complementary algorithms involving tactics mixing, task sharing, search space splitting and result compilation, we imagined in (Hernandez et al. 2005) a cyclic identification system in which several identification strategies would be coherently used according to criteria like the number of spectra to process, their quality, the size of the searched database, the analysis type to carry out as well as scores obtained by previous identification attempts. The last figure of this work shows an example of such a system.

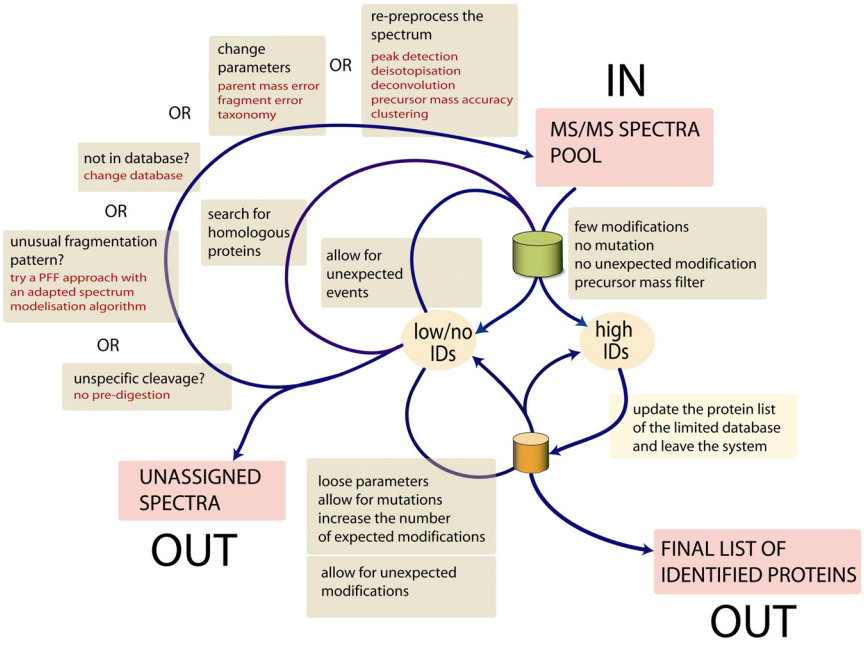



*Figure IX-1: A cyclic system for MS/MS identification*
*Spectra enter the system and remain there until a given criterion is met (typically, a confident identification or on the contrary, a definitively non-identifiable spectrum). The system splits the search space of peptides into two databases, a large one and a small one. Spectra that obtain a high enough identification score to be assigned without doubt to a peptide (high-IDs) are used to feed the limited list of proteins that are potentially present in the sample. Once a spectrum has been classified as "high IDs", it definitively leaves the system and participates to the final list of identifications. The "low/no IDs" remain in the cycle and follow a different path. As they circulate inside the system, the spectra are annotated to keep trace of the applied algorithms, the obtained scores and the used parameters. These annotations help the system to decide what next path a spectrum should follow.*

# X. APPENDICES

Appendix A1: List of abbreviations

| | |
|---|---|
| CE | capillary electrophoresis |
| HPLC | high-performance liquid chromatography |
| ESI | electrospray ionization |
| MALDI | matrix-assisted laser desorption/ionization |
| FAB | fast atom bombardment |
| CID | collision induced dissociation |
| FD | field desorption |
| TQ | triple quadrupole |
| Q-TOF | quadrupole/time-of-flight |
| TOF-TOF | time-of-flight/time-of-flight |
| Q-IT | quadrupole ion trap |
| MS | mass spectrometry |
| MS/MS | tandem mass spectrometry |
| PMF | peptide mass fingerprinting |
| PFF | peptide fragment fingerprinting |
| SPC | shared peak count |
| EM | expectation-maximisation |
| GP | genetic programming |

Appendix A2: MS (PMF) identification algorithms and URLs

| | |
|---|---|
| FragFit | N/A |
| PeptideSearch | http://www.mann.embl-heidelberg.de/GroupPages/PageLink/peptidesearchpage.html |
| Peptident | http://www.expasy.org/tools/peptident.html |
| PepFrag | http://prowl.rockefeller.edu/prowl/pepfragch.html |
| MOWSE | N/A |
| MS-Fit | http://prospector.ucsf.edu/ |
| Mascot | http://www.matrixscience.com |
| ProFound | http://prowl.rockefeller.edu/profound_bin/WebProFound.exe |
| SmartIdent | N/A |
| PepMapper | http://wolf.bms.umist.ac.uk/mapper/ |
| Aldente | http://www.expasy.org/tools/aldente/ |

Appendix A3: MS/MS (PFF) identification algorithms and URLs

**PFF**

| | |
|---|---|
| Phenyx | http://phenyx.vital-it.ch/ |
| Sequest | http://fields.scripps.edu/sequest/index.html |
| Mascot | http://www.matrixscience.com/search_form_select.html |
| PepFrag | http://prowl.rockefeller.edu/prowl/pepfragch.html |
| MS-Tag | http://prospector.ucsf.edu/ucsfhtml4.0/mstagfd.htm |
| ProbID | http://projects.systemsbiology.net/probid/ |
| Sonar | http://65.219.84.5/service/prowl/sonar.html |
| TANDEM | http://www.thegpm.org/TANDEM/ |
| SCOPE | N/A |
| PEP_PROBE | N/A |
| VEMS | http://www.bio.aau.dk/en/biotechnology/vems.htm |
| InsPecT | http://peptide.ucsd.edu/inspect.py |

**MS/MS "open-modification search" algorithms and URLs**

| | |
|---|---|
| GutenTag | http://fields.scripps.edu/GutenTag/index.html |
| PEDANTA | http://peptide.ucsd.edu/ |
| OpenSea | NA |
| **Popitam** | **http://www.expasy.org/tools/popitam/** |

***de novo* sequencing algorithms**

| | |
|---|---|
| SeqMS | http://www.protein.osaka-u.ac.jp/rcsfp/profiling/SeqMS.html |
| Lutefisk | http://www.hairyfatguy.com/Lutefisk |
| Sherenga | N/A |
| PEAKS | http://www.bioinformaticssolutions.com/products/peaksoverview.php |

**sequence similarity search algorithms**

| | |
|---|---|
| PeptideSearch | http://www.narrador.embl-heidelberg.de/GroupPages/PageLink/peptidesearchpage.html |
| PepSea | http://www.unb.br/cbsp/paginiciais/pepseaseqtag.htm |
| MS-Seq | http://prospector.ucsf.edu/ucsfhtml4.0/msseq.htm |
| MS-Pattern | http://prospector.ucsf.edu/ucsfhtml4.0/mspattern.htm |
| Mascot | http://www.matrixscience.com/search form select.html |
| FASTS | N/A |
| MS-Blast | http://dove.embl-heidelberg.de/Blast2/msblast.html |
| OpenSea | N/A |
| CIDentify | http://ftp.virginia.edu/pub/fasta/CIDentify/ |

## Appendix A4: Popitam's command line output

A typical text output of Popitam is given below with explanations:

```
File processed                : MOD1/data_pop_7.pop
Initial number of spectra     : 1
Instrument                    : Q-TOF
COVBIN                        : 6
Edges_type                    : 1,2 aa
Fragment error1               : 0.2
Fragment error2               : 0.4
Minimal tag length            : 3
```

This first header summarizes the parameters used for the search, such as the errors set on the fragments during graph connection.

```
DTB                           : SPROTREMBL, Staphylococcus aureus (strain N315)
AC FILTER?                    : NO
Peptide error                 : -150 to +150 Da
```

This header gives information about different filters applied to the peptides. Candidate peptides are the ones that go through all these filters (species, ACs (if any) and peptide error).

```
MODGAPS                       : 1
MAX_LOSS_MOD                  : 150.00
MAX_ADD_MOD                   : 150.00
MIN_ARR_COV                   : 0.3
```

This header concerns filters on the scenarios. Valuable scenarios that are finally evaluated are the ones that contain exactly 1 *modGap* with a shift comprised between -150 and +150 Da, and that cover at least a third of the candidate peptide.

The next header is about the spectrum being analyzed. The title, number of peaks (before and after preprocessing), parent mass and graph sizes are displayed. Then, some statistics are given, notably the number of candidate peptides and valuable scenarios.

```
SpectrumTitle            : TFE_Pellet_Band01_ESI.txt_1380.926000 Q99XA3 LPIPNTVSDLSPK
initPeakNb               :  93 (102)
ParentMass(M)/Charge     : 1379.93 / 2
NodeNb                   :      80
EdgeNb (simple/double)   :     106 / 1442


Number of protein processed      :      2577
Peptide obtained after digestion :    168500
Number of candidate peptides     :     14610
Number of valuable scenarios     :       377
```

Finally, a table presents the first ranked candidate peptides. The first column gives the ranks of the peptides, the second gives their scores. Finally, the last column displays the peptide sequence and the interpretation scenario. When one or several *modGaps* are present, the associated shifts are also given. The p-values and deltaScores associated with the final score are useful to assign confidence to the result. The p-values can either be computed from the distribution of negative scores obtained during the run, or from the distribution of random scores. In that case, Popitam generates random peptides by randomly shuffling the amino acids of candidate peptides. The deltaScores are obtained

by dividing the final score by the final score of the next ranked peptide. Both p-values and deltaScores are ranked between 0 and 1.

| # | scores | Mr | access | id | dbSeq/scenarios | |
|---|---|---|---|---|---|---|
| 1. | 805.49 | 1407.76 | Q7A873 | Q7A873_STAAN | LPIPNTVDDLSPK | |
| | 0.00e-38 ; 0.08 | | | | lpIpnTV*DLSpk | -27.89 |
| 2. | 69.95 | 1258.63 | Q7A5Z8 | Q7A5Z8_STAAN | TTPNTGERVER | |
| | 8.73e-10 ; 0.92 | | | | **pnTge---- | 121.12 |
| 3. | 64.33 | 1496.89 | Q7A8B4 | Q7A8B4_STAAN | ILGRVLATDIDIAK | |
| | 8.45e-08 ; 0.87 | | | | **grvl---ID--- | -117.13 |

870 DOI 10.1002/pmic.200300402 Proteomics 2003, *3*, 870–878

**Patricia Hernandez**[1]
**Robin Gras**[1]
**Julien Frey**[1]
**Ron D. Appel**[1, 2]

[1]Swiss Institute of Bioinformatics, Geneva, Switzerland
[2]University of Geneva and Geneva University Hospital, Switzerland

# Popitam: Towards new heuristic strategies to improve protein identification from tandem mass spectrometry data

In recent years, proteomics research has gained importance due to increasingly powerful techniques in protein purification, mass spectrometry and identification, and due to the development of extensive protein and DNA databases from various organisms. Nevertheless, current identification methods from spectrometric data have difficulties in handling modifications or mutations in the source peptide. Moreover, they have low performance when run on large databases (such as genomic databases), or with low quality data, for example due to bad calibration or low fragmentation of the source peptide. We present a new algorithm dedicated to automated protein identification from tandem mass spectrometry (MS/MS) data by searching a peptide sequence database. Our identification approach shows promising properties for solving the specific difficulties enumerated above. It consists of matching theoretical peptide sequences issued from a database with a structured representation of the source MS/MS spectrum. The representation is similar to the spectrum graphs commonly used by *de novo* sequencing software. The identification process involves the parsing of the graph in order to emphazise relevant sections for each theoretical sequence, and leads to a list of peptides ranked by a correlation score. The parsing of the graph, which can be a highly combinatorial task, is performed by a bio-inspired algorithm called Ant Colony Optimization algorithm.

**Keywords:** Ant colony optimization / Mutated or modified peptides / Protein automated identification / Spectrum graph / Tandem mass spectrometry PRO 0402

## 1 Introduction

Rapid and automated protein identification from mass spectrometric data is an essential task in proteomics research. Identification tools should be able to handle fast and fully automated protein identification, as well as more targeted identification of specific proteins of clinical interest (and more specifically, of modified or mutated proteins). The most common method used today for MS/MS identification is the Shared Peak Count (SPC), which compares the experimental MS/MS spectrum with theoretical peptidic "ion mass fingerprints" of virtually digested and fragmented proteins. Various SPC algorithms have been developed [1]. They differ mainly by their scoring functions, which can be more or less sophisticated. MS-Tag, PepFrag [2] and MASCOT [3] use a probabilistic scoring function. SCOPE [4] uses both a complex probabilistic model and a dynamic programming method. Another algorithm, SEQUEST [5], uses two filtering levels: an SPC, followed by cross-correlation by means of fast Fourier transformation. A recent approach, sonar [6], reduces the experimental and theoretical MS/MS spectra into two multidimensional peak intensity vectors, with a fixed number of dimensions. The comparison score is a simple dot product between the two vectors. The identification is then validated using a statistical method in order to discriminate false positives from true positives [7]. SPC methods include in the database all modified peptides that they want to consider, which requires prior knowledge of the mass differences associated with the corresponding modifications. Moreover, including all possible modifications or mutations of peptides in the database is unrealistic due to the combinatorial explosion it implies. As a result, SPC methods usually take into account only a few very common modifications occurring on specific amino acids, such as methionine oxidation or cysteine carbamidomethylation. In addition to the combinatorial problem, SPC algorithms have two other limitations. First, they usually consider the peaks independently of each other, thereby wasting important information contained in MS/MS spectra. Second, SPC algorithms need to allow a large error of tolerance when used with badly calibrated spectra.

**Correspondence:** Dr. Patricia Hernandez, Swiss Institute of Bioinformatics, CMU, rue Michel-Servet 1, 1211 Geneva 4 – Switzerland
**E-mail:** patricia.hernandez@isb-sib.ch
**Fax:** +41-22-702-58-58

**Abbreviations: ACO**, Ant Colony Optimization; **SPC**, Shared Peak Count

 0173-0835/03/0606–870 $17.50+.50/0

Interesting identification approaches are the spectral convolution and the spectral alignment, implemented in PEDANTA [8], which are claimed to be very efficient in dealing with modifications and mutations, including unpredictable modifications. Indeed, the spectral alignment has a major advantage over SPC methods, because it uses logical constraints imposed by the spectrum peak composition to limit the number of considered modifications and mutations. But one obvious trade-off of these two approaches is that one must parse the whole peptide database without using the parent mass as a filter. In addition, the combinatorial problem grows with the number of contemplated mass shifts. Accordingly, the number of modifications/mutations considered must be kept rather low in order to allow identifications that are sufficiently discriminating.

A different identification strategy is *de novo* sequencing (by hand or automatically) followed by sequence matching [9]. One or more peptide sequences or tags are directly derived from the MS/MS spectrum independently of any information extracted from a pre-existing protein or DNA database. Lutefisk [10], Sherenga [11], and SeqMS [12] are software designed to infer complete sequences from MS/MS spectra. Chen *et al.* [13] described a method that allows the extraction of complete sequences with one unspecified modification using dynamic programming. Recently, Shlosser and Lehmann [14] described a new strategy that works by extracting small tags from low mass ions. Unfortunately, *de novo* sequencing is a quite complex task, which requires both good quality spectra and manual verification by a MS expert. Accordingly, this approach is not adapted to the huge amounts of data generated by high-throughput settings available nowadays. Another drawback is that the identification is performed in two separate and independent steps. The first step, tag extraction, does not take advantage of any useful information from the database, while the second one, the alignment of the tags with the theoretical sequences, uses only a very reduced view of the original information.

We present a new method for protein identification from MS/MS data, called Popitam, in which we deliberately chose to favor nondeterministic cooperative strategies when confronted with combinatorial problems. The algorithm can be briefly described as follows: after a preprocessing step, in which a given number of peaks in the spectrum are selected according to their intensity, the MS/MS spectrum is transformed into a graph [10, 11] (the graph allows structuring of the information, and then captures the relationships between the peaks), which is compared with theoretical peptides from a database, leading to a ranked list of scored candidate peptides. While identification methods based on tag search typically try to extract tags (or complete sequences) from the graph and then use them to find peptides in the database using alignment software, Popitam uses the database to direct the search and to emphasize relevant sections in the graph from which the peptides can be scored. A first version, denoted as the "Full Path algorithm", has been implemented and tested (a detailed description of the algorithm is given in Section 2). As the name shows, this version works by looking for complete paths in the graph. Although it is already functional, the "Full Path algorithm" is only a rough draft for a new and more efficient version of Popitam, based on extraction of tags from the graph and therefore called "Tag algorithm".

Popitam's "Tag algorithm", which is briefly described in Section 3.2.2, brings three major advantages over the first version. First, it is poorly influenced by a bad calibration, second it is less sensitive to low fragmentation of the source peptide, and finally, it should be able to handle unknown modifications as well as mutations of the source peptide. Of course, by looking for sections in the graph, instead of complete paths, and by allowing modifications and mutations, we will have to face a combinatorial problem. Various strategies are exploited to reduce the search space. The number of theoretical peptides to be analyzed can be reduced by using two filtering levels, and the number of mutation/modifications to consider are greatly reduced, because the graph structure allows use of logical constraints of the spectrum peak composition, and in addition, only modifications that are consistent with a large covering of the theoretical sequences by pertinent tags are analyzed. Since the preliminary step of tag discovering becomes a more combinatorial problem than for the "Full Path algorithm", the use of a heuristic search turns out to be particularly appropriate to efficiently explore the graph.

Ant Colony Optimization (ACO) metaheuristics is a class of several related algorithms built from an original prototype called Ant System, initially proposed by Colorni in 1991 [15]. These algorithms are defined as multi-agent systems inspired from real ant colony behavior. They have been successfully applied to a wide range of difficult combinatorial optimization problems, as vehicle routing, job-shop scheduling or graph coloring [16]. The principle of ACO is to explore, iteratively and simultaneously, different solutions of a given problem by an ant-agent population. The emergent collective behavior is guided by indirect communication between the ants, mediated by environmental modifications (stigmergy). Ants modify their environment by depositing given amounts of pheromone, which are locally accessible and affect the behavior of the other ants. Since ants can find the shortest path con-

necting the colony to a food source, it is possible to exploit the rules governing the foraging process and to use them to find good scoring paths in the graph.

## 2 Materials and methods

### 2.1 General method

In this section, we describe Popitam's "Full Path algorithm". The first step consists of transforming the *m/z* of the source MS/MS peak list into potential b-ion type fragments. For comprehension purposes, we will call such fragments bFragments, and their corresponding masses bMasses. The peak list is then structured into a graph similar to the spectrum graph commonly used for *de novo* sequencing. The identification process implies comparing the graph with theoretical peptides referenced in a database and leads to a similarity score for each peptide sequence. This score is then used to determine the best peptide match or matches.

### 2.2 Transforming *m/z* into bMasses

Let us define $S_{exp} = \{s_1, s_2, \ldots, s_{|Sexp|}\}$, the experimental MS/MS peak list to be identified, and a set of ionic hypotheses $\Delta = \{\eta_1, \eta_2, \ldots, \eta_{|\Delta|}\}$. An ionic hypothesis is a possible interpretation of a peak as a given ionic fragment. Each $\eta_k$ has four attributes, which are presumptions concerning the ionic fragment $s_i$ measured by the spectrometer: an offset value $o_k$, *i.e.* the mass difference between the ionic fragment and the corresponding bFragment, a terminus side $t_k$ (*N*-term or *C*-term), a number of charges $c_k$, and an approximated occurrence probability $p_k$. The probabilities $p_k$ depend among other things on the spectrometer used and can be approximated during a learning phase using a set of identified spectra [11]. The bMasses $\mu_j$ are obtained by attributing to each *m/z* $\mu_i^{m/z}$ of the peak list all ionic hypotheses from $\Delta$ comprising all four attributes described above according to Equation 1.

$$\text{if } (t_k = \text{"}N\text{-term"})\ \mu_j \leftarrow c_k \cdot \mu_i^{m/z} - (c_k - 1) - o_k$$

$$\text{if } (t_k = \text{"}C\text{-term"})\ \mu_j \leftarrow M_{exp} - [c_k \cdot \mu_i^{m/z} - (c_k - 1) - o_k]; \qquad (1)$$

with

$t_k$, $c_k$ and $o_k$ the terminal, charge and offset of the ionic hypothesis $\eta_k$, $M_{exp}$ the experimental charge obtained from the observed parent mass $M_{obs}$ and its charge $c_{obs}$, with $M_{exp} = (M_{obs} - 1) \cdot c_{obs}$. Each peak from $S_{exp}$ leads to $|\Delta|$ potential bFragments, and among these $|\Delta|$ bFragments, at least $|\Delta| - 1$ are false interpretations.

### 2.3 Building the graph

Let us define G = (V,E) as a directed acyclic graph, with $V = \{v_1, v_2, \ldots, v_{|V|}\}$ a set of vertices, and $E = \{e_{ij} \mid i<j<|V|, v_i$ and $v_j \in V\}$ a set of edges. Each vertex $v_i$ is characterized by a bMass $\mu_i$, the corresponding ionic peak *m/z* $\mu_i^{m/z}$, an intensity $\iota_i$, and an ionic hypothesis $\eta_i$. In addition, it receives a score $\sigma_i$, a family $F_i$ and a successor list $succ_i$ representing all its connected vertices. Each edge $e_{ij} \in E$ from $v_i$ to $v_j$ is characterized by a pheromone trail $\tau_{ij}$ and a label $\lambda_{ij}$. We also create an initial vertex corresponding to the empty sequence and a final vertex corresponding to the complete sequence.

#### 2.3.1 Families

For each vertex, a family F is defined. The concept of family is based on the idea that when a bFragment is represented by several ionic peaks in $S_{exp}$, the computed bMasses of these peaks will be almost equal. Since vertices that belong to large families are more likely to be true positives than orphan vertices, we use the size of the family for scoring the vertices (see Section 2.3.2). We chose not to merge the vertices as described in [11], because the merging process does not handle the calibration error on the peaks and depends on the parent mass accuracy, which is often quite low. Accordingly, two bMasses representing the same bFragment and derived by ionic hypotheses of different terminal types can be quite different when compared to bMasses obtained from ionic hypotheses of the same terminal type. Such bMasses therefore cannot be merged because they are too different or, if merged, can produce a new vertex with a substantially less accurate bMass. Since we do not merge the vertices, several paths in the graph are equivalent and represent a same bFragment. The rules for adding a vertex $v_j$ to a family $F_i$ are as follows: first, the two vertex bMasses must be close enough. The threshold must be adapted, depending on whether the two vertices joined in a same family are derived by ionic hypotheses of a same terminal type or of different terminal types. Second, the two vertex bMasses have to be issued from different ionic hypotheses.

#### 2.3.2 Vertex score

Because the vertices are built under some assumptions, we need a value defining the credibility level of each vertex. This value is represented by a score $\sigma_i$ and is computed from the occurrence probability $p_i$ of the ionic hypothesis used to build the vertex, and from its family size according to Equation 2.

$$\sigma_i = p_i^{\frac{1}{|F_i|}} \qquad (2)$$

### 2.3.3 Graph connection

If the bMasses of two associated vertices $v_i$ and $v_j$ differ by the value of one or several amino acids, they can be connected by an edge $e_{ij}$. According to the number *N* of amino acids included in a given edge, the latter can be called a simple edge ($|\lambda_{ij}| = 1$), a double edge ($|\lambda_{ij}| = 2$), and so on. Let $A = \{a_1, a_2, \ldots, a_{|A|}\}$ be the alphabet of the amino acids. A contains all common amino acids, as well as some modified amino acids, such as carboxymethylated cysteine, carbamidomethylated cysteine, or oxidated methionine. $A^c = \left\{a_1^c, a_2^c, \ldots, a_{|A^c|}^c\right\}$ is the set of all combinations of 1 to *N* amino acids among |A|. Because the edge number increases exponentially with the value of *N*, the latter is usually small (typically *N* = 2).

Algorithm 1 (described here in a simplified form) shows the principle of edge creation. The vertex list must be sorted according to the bMass values.

```
For i = 0 to |V|{
  For j = i + 1 to |V|{
    if (ti = tj) ε ← ε1;
    else ε ← ε2;        || ε1 < ε2
    For n = 1 to |A^c|{
      if ( |μ_i^v − μ_j^v − μ^a_n^c| < ε) create Edge (e_ij, a_n^c)
    }
  }
}
```

**Algorithm 1.** Connecting the graph

with: $\mu^{a_n^c}$, the sum of the masses of all amino acids in $a_n^c$; $\mu_i^v$ and $\mu_j^v$, the masses of vertices i and j; and $\varepsilon_1$, $\varepsilon_2$, the two thresholds (with $\varepsilon_1 < \varepsilon_2$).

### 2.4 Identification process

### 2.4.1 The peptide database

Let $D = \{P_1, P_2, \ldots P_{|D|}\}$ be the peptide database used for the identification.

The identification process consists of comparing the peptides of D with the graph G, leading to the computation of an identification score $I(P_c)$ for each peptide $P_c$.

### 2.4.2 Comparison process

The comparison process between the graph G and a peptide $P_c$ requires finding in G the sections that best explain $P_c$. Figure 1 shows an example of the state of the graph during the comparison of two theoretical peptides with an experimental MS/MS spectrum.

The Full Path version of Popitam that we are describing here is meant to find complete sections in the graph (a complete section is a path starting from the first vertex, and corresponding to a whole peptide sequence).

Let $F = \{f_1, f_2, \ldots, f_{|F|}\}$ be the ant population. Each ant $f_k$, walking on the graph at iteration t, builds a path $L^k$. The quality of $L^k$ is represented by the ant's score $S^k$. The concatenation of the edge labels included in $L^k$ represents the sequence parsed in the graph.

Algorithm 2 is an adaptation to our problem of an ACO algorithm. First, the amount of pheromone $\tau_{ij}$ of each edge $e_{ij} \in G$ is initialized (with $\tau_0 = 10^{-6}$), as well as the best complete path found in the graph ($L^+$) and its associated score $S^{L+}$. At the beginning of each iteration ($t_{max}$ is a predefined total number of iterations), the amount of pheromone that will be added at each edge, $\Delta\tau_{ij}$, is initialized at 0. Then, each ant parses the graph, building a solution $L^k$ with a score $S^k$. This score is used for updating the $\Delta\tau_{ij}$ for each $e_{ij} \in L^k$ (Q is a predefined constant value, chosen of a same order of magnitude as that of the optimal score. It has been demonstrated that the value of Q has little influence on the final result [17]). If the path built by the ant obtains a higher score than $S^{L+}$, $L^+$ and $S^{L+}$ are updated.

```
Initiation:
L+ ← ∅; S^L+ ← 0;
For each edge e_ij: τ_ij ← τ_0;

Iterations:
For t = 1 to t_max {
  For each edge e_ij: Δτ_ij ← 0;
  For each ant k do{
    L^k ← parseGraph(P);
    S^k ← scorePath (P, L^k);
    For each edge e_ij ∈ L^k: Δτ_ij ← Δτ_ij + S^k/Q;
  }
  S^L+ = max_k (S^k);
updateGraph(); || update the pheromone values on the
edges with Δ_ij
  }
```

**Algorithm 2.** General framework of the ant algorithm

When all ants have parsed the graph and have added their contribution to the $\Delta\tau_{ij}$, the graph is updated according to Equation 3. The value $\omega \in [0;1[$ is called *evaporation*

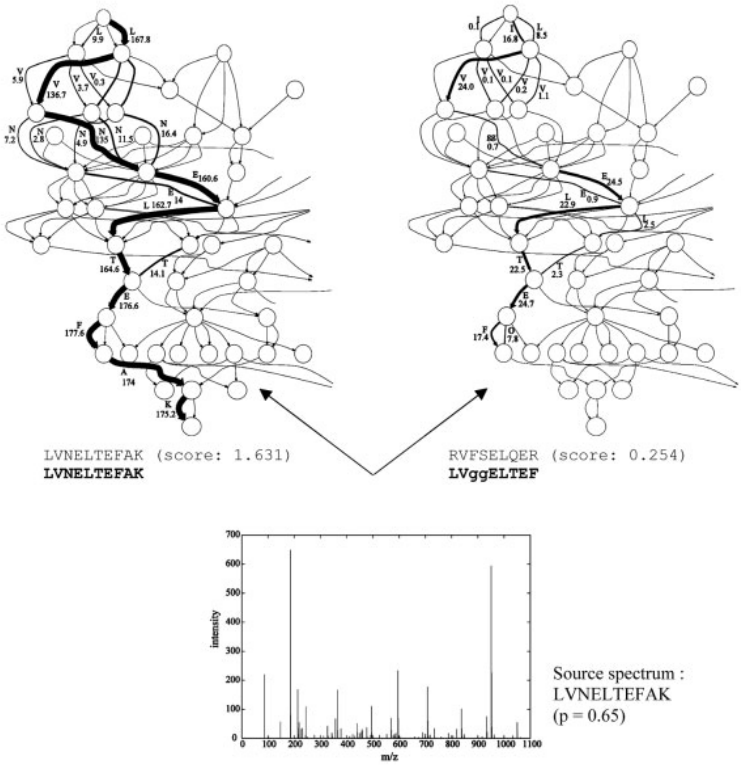



**Figure 1.** Illustration of the graph state after comparison (Full Path algorithm) with two theoretical sequences: On the left, the sequence LVNELTEFAK (which corresponds to the source peptide at the origin of the spectrum), and on the right, another good scoring peptide sequence from the database. Bold letters represent the best scoring path for each sequence. Edges are weighted according to the amount of pheromone deposited by the ants. For clarity purposes, the graph is represented only with single edges (one amino acid edge), except for the parsed path, which includes a double edge.

*rate*. It allows a better exploration of the search space by removing a certain quantity of pheromone from each edge.

$$\tau_{ij} \leftarrow (1 - \omega) \cdot \tau_{ij} + \Delta\tau_{ij} \quad (3)$$

A more detailed description of the *parseGraph* and *score Ant* functions follows.

### 2.4.3 Parsing of the graph

The ant $f_k$ is first placed on the initial vertex $v_0$. It can go forward as long as the current vertex $v_i$ has one or more successors ($succ_i \neq \emptyset$). The transition rule used to go from a vertex $v_i$ to a vertex $v_j$ with $v_j \in succ_i$ depends on several factors. First, an edge may be disabled if the successor vertex has been obtained from a peak already used in the ant's path, thus avoiding using different interpretations of the same peak. The list of peaks used by an ant is stored in a individual memory, called tabooList. If the peak associated with $v_j$ has already been used by an ant k, the function tabooList($v_j$) returns true, and the probability $p^k(e_{ij})$ to take the edge $e_{ij}$ is set to null. Else, the computation of $p^k(e_{ij})$ depends on three pieces of information. The first one is visibility, symbolizing the *a priori* desirability of the move [18] and represented by the score $\sigma_j$ of the successor vertex. The second piece of information corresponds to the knowledge acquired in previous iterations, symbolizing the *a posteriori* desirability of the move, and represented by the amount of pheromone $\tau_{ij}$ laid on the edges. Finally, the third piece of information is the sequence of the current database peptide $P_c$. Indeed, if the label of the next edge $e_{ij}$ matches the next amino acid in the sequence of $P_c$, the transition probability is multiplied by a predefined constant value $\sigma_l$ dependent on the edge label length.

Given $\alpha$ and $\beta$, two adjustable parameters controlling the relative weights of the learning and the visibility respectively, $p^k(e_{ij})$ the probability for ant $f_k$ to take the edge $e_{ij}$ at iteration $t$, $p^k(e_i) = \{p^k(e_{ij}) \forall j \in succ_i\}$ the set of these probabilities for all successors of $v_i$, and $Q(P_c) = \{a_1^P, a_2^P, \ldots, a_l^P\}$ the current peptide sequence, the parsing of the graph follows algorithm 3.

```
i = 1; || position in the graph
n = 1, || position in the peptide sequence
L^k = ∅;
while (succ_i ≠ 0) {
  for each v_j ε succ_i {
    if (tabooList(v_j)) p^k(e_ij) = 0;
    else {
      p^k(e_ij) ← (τ_ij)^α · (σ_j)^β;
      if (match(λ_ij, P_n)) : p^k(e_ij) ← p^k(e_ij) · c_i;
      p^k(e_i) ← p^k(e_i) ∪ p^k(e_ij);
    }
  }
  normalize (p^k(e_i));
  e_ij = chooseEdge (p^k(e_i));
  updateTabooList (v_j);
  n = n + |λ_ij|;
  L^k ← L^k ∪ e_ij;
  i ← j;
}
```

**Algorithm 3.** Parsing the graph with ant k

#### 2.4.4 Ant's score

At each iteration t, one must evaluate the similarity between the current peptide $P_c$ and the different paths used by the ants. Each ant gets a score $S^k$ depending on its path $L^k$. The goal is to include in $S^k$ all possibly relevant information from various sources. We use for the moment four subscores: (1) covS, a coverage measure (for example a Hamming distance) between the peptide sequence in the database and the sequence parsed in the graph by the ant; (2) intS, the mean of the intensities of the peaks included in $L^k$; (3) relS, for relevancy score, the mean of the probability occurrences of the ionic hypotheses included in $L^k$; and (4), regS, a regression score representing the quality of the correlation between the bMasses of the vertices in $L^k$ and the theoretical masses expected from the theoretical sequence. This last subscore measures the global correspondence between the experimental masses $\mu_i^{m/z}$ of the vertices included in the ant's path and the corresponding theoretical masses computed from the current database peptide sequence: a linear regression is calculated from the set of couples of theoretical and experimental masses, and the mean of the deviation between the points and the straight line of the regression represents the regression score [19]. Each subscore is scaled so that it takes a value between 0 and 1. The different subscores are then combined to build the score $S^k$ (see Equation 4).

$$S^k = \frac{covS^{pCov} \cdot relS^{pRel} \cdot intS^{pInt}}{regS^{pReg}} \quad (4)$$

The weights pCov, pRel, pInt and pReg have been learnt using a genetic algorithm [19] and an independent learning set of MS/MS spectra. Other information can be added, such as rules resulting from the expertise of biologists used to studying MS/MS data.

## 3 Results and discussion

### 3.1 Testing set

To test Popitam, we used a set of 721 identified MS/MS spectra obtained from nucleolar proteins purified by SDS-PAGE and 2-DE, digested with trypsin and analyzed with a Q-TOF mass spectrometer [20]. The spectra were first identified using the MASCOT search engine (http://www.matrixscience.com) against SWISS-PROT and TrEMBL databases [21]. When necessary, identification was confirmed by manual *de novo* sequencing using the ProteinInfo search engine from PROWL (http://prowl.rockefeller.edu). In order to have comparable results, we have computed a quality value for our data, as defined in [8] (see Fig. 2).

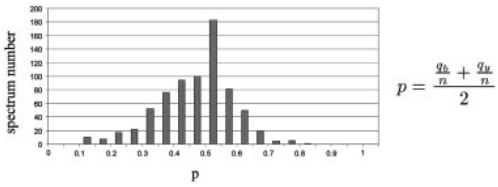



**Figure 2.** Histogram obtained from our learning set of 721 spectra and representing the spectrum number as function of a quality value p. The quality value is the average of the ion-type b and y frequencies (*n* is the sequence length).

### 3.2 Results with Popitam's Full Path algorithm

Popitam was able to correctly identify 641 spectra out of 721 (88.9%). Eighteen spectra were proposed as second rank. The 62 remaining ones were ranked below, or missed. Table 1 shows the percentage of correct identification (first rank) according to the quality value p.

**Table 1.** Percentage of correct identification (Full Path Algorithm) according to the quality value p

| p | Spectrum number | Success | % |
|---|---|---|---|
| [0; 0.1] | 0 | – | – |
| [0.1; 0.2] | 18 | 1 | 5.6 |
| [0.2; 0.3] | 38 | 16 | 42.1 |
| [0.3; 0.4] | 128 | 105 | 82 |
| [0.4; 0.5] | 194 | 181 | 93.3 |
| [0.5; 0.6] | 264 | 259 | 98.1 |
| [0.6; 0.7] | 69 | 69 | 100 |
| [0.7; 0.8] | 9 | 9 | 100 |
| [0.8; 0.9] | 1 | 1 | 100 |
| [0.9; 1] | 0 | 0 | – |
| [0; 1] | 721 | 641 | 88.9 |

Typically, identification software usually fail in identifying peptides in the following cases: (1) the spectrum quality is too bad; this includes bad calibration, low fragmentation, noise, incorrect precursor charge, *etc.*; (2) the source peptide carries one or several unsuspected post-translational modifications or is mutated; (3) the source peptide is not present in the database; (4) there is a problem in the digestion process (nonspecific cleavages). The current implementation of Popitam is also subject to these problems. Considering Popitam's performance, we have to understand why it failed to correctly identify 80 spectra out of 721.

### 3.3 Improvements in progress

#### 3.3.1 Scoring function

A possible explanation is related to the scoring formula, which evaluates the correlation level between the theoretical peptides and the graph. For the moment, it does not include all the rules we intend to use. New scoring formulae discovered by Genetic Programming [22] are currently under investigation. Such formulae are generated by combining mathematical operators and identification subscores (the regScore, perScore, intScore and covScore described in Section 2.4.4.). The solutions are evaluated using a set of identified spectra. One of the pertinent solutions obtained is shown in Fig. 3.

Preliminary results are very encouraging, since we have obtained many more discriminative scores and as a consequence, an increase in Popitam's performance. Table 2 shows the results obtained with the optimized scoring function described in Fig. 3.

Nevertheless, Popitam's Full Path Algorithm has a significant drawback which explains its limited performance on our set of spectra: since it looks for complete paths in the graph, it struggles to identify spectra having a gap of two or more consecutive fragmentation positions. Such gaps force the ants to lose the correct path, and there may be no possibility to reach it again later.

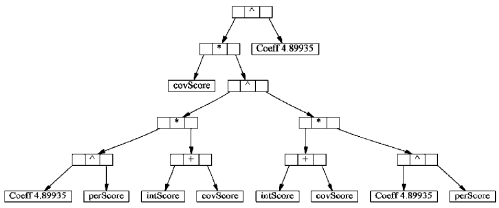



**Figure 3.** Scoring function generated by Genetic Programming. The candidate functions are represented as trees. New solutions are generated by exchanging information between trees, or changing the value of a node or a leaf.

**Table 2.** Percentage of correct identification (Full Path Algorithm, optimized scoring function) according to the quality value p

| p | Spectrum number | Success | % |
|---|---|---|---|
| [0; 0.1] | 0 | – | – |
| [0.1; 0.2] | 18 | 0 | 0 |
| [0.2; 0.3] | 38 | 18 | 47.4 |
| [0.3; 0.4] | 128 | 108 | 84.4 |
| [0.4; 0.5] | 194 | 187 | 96.4 |
| [0.5; 0.6] | 264 | 261 | 98.9 |
| [0.6; 0.7] | 69 | 69 | 100 |
| [0.7; 0.8] | 9 | 9 | 100 |
| [0.8; 0.9] | 1 | 1 | 100 |
| [0.9; 1] | 0 | – | – |
| [0; 1] | 721 | 653 | 90.6 |

#### 3.3.2 Popitam's Tag algorithm

We are now working on a new version of Popitam, called Tag algorithm, which allows extracting weighted sections from the graph, rather than entire paths. This new version is weakly influenced by a bad calibration, since the parsing does not necessarily include the extremities of the graph (the first vertex corresponding to the empty sequence and the last one, corresponding to the complete sequence), which have arbitrary fixed masses. The use of a graph allows comparing mass differences instead of absolute masses. Consequently, Popitam can use a lower threshold than other identification software. This threshold takes place during the graph connection for Popitam and during the matching for SPC algorithms.

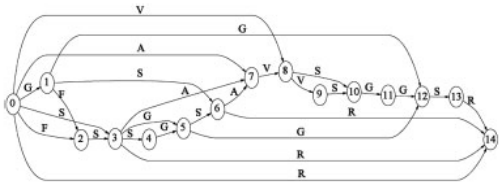


**Figure 4.** Factor oracle of the peptide GFSSGSAVVSG GSR. Numbers in circles are the states of the automaton. All states can be terminal. Starting from state 0, one can build at least all possible subsequences of GFSSGSAVVSGGSR.

It is an important parameter, because it is at the origin of an increase of false positive edges or matches between theoretical and experimental masses.

Popitam's Tag algorithm is currently under development, but it has already produced interesting results. In this algorithm, ants use a factor oracle [23] of the theoretical peptide to be compared. A factor oracle is a deterministic finite state automaton [24] able to represent at least all possible subsequences of a given word. Figure 4 shows the factor oracle of the peptide GFSSGSAVVSGGSR.

Ants start from any vertex in the graph, and from state 0 in the automaton. They parse the graph as long as they can move forward simultaneously in both structures, following only arcs in the automaton that match edges in the graph. This ensures that only paths in the graph that exactly match a subword of the peptide are considered. Since this new version of Popitam is not complete yet, only preliminary results are available. They were obtained using an elementary scoring function (the correlation measure of a tag is simply the number of parsed vertices), and keeping in memory only one tag for each theoretical peptide. Despite these extremely basic conditions, Popitam's Tag Algorithm has already been able to correctly identify a great number of spectra (78.6%) from the training set. In addition, we believe that the new version of Popitam will be able to identify modified and mutated peptides without any prior knowledge of the modifications to be considered. Modifications can be simulated by joining unconnected relevant sections of the graph in order to maximally cover the theoretical peptides. Since we will only test modifications/mutations that connect pertinent tags (according to the theoretical sequences), this approach strongly restrains the search space. In order to avoid parsing the whole peptide database, we plan to perform a first identification step using the parent mass as a filter to extract candidate peptides from the database. A spectrum produced by a modified peptide will not lead to any high score, but highlighted tags can be used as a filter for a second identification step. This time, the identification will be performed independently of the parent mass.

## 4 Concluding remarks

SPC algorithms are reaching their limits in identifying mutated or modified peptides from MS/MS spectra. Nowadays, new strategies in protein identification from MS/MS spectra tend to involve tag extraction from the source spectra. A tag approach brings great benefits, as for example taking into account successions of peaks or focalizing on informative regions of a spectrum. It is also an efficient approach for identifying mutated or modified peptides, by finding tags located before and/or after the modifications. To our knowledge, no current MS/MS identification method based on tag extraction takes advantage of information contained in databases during the extraction process. This information can be of great help, because it allows considerable reduction of the search space, which is rather vast when looking for tags. The search space can be explored efficiently using a nondeterministic heuristic approach as an Ant Colony Optimization algorithm. ACO are particularly convenient, because these algorithms are very flexible and can be easily adapted to the identification problem. The strategies described in the present work take into consideration several important difficulties encountered by the current MS/MS identification tools. A first version of Popitam denoted as Full Path Algorithm includes some of these strategies and leads to promising results. For these reasons, we think that Popitam's Tag Algorithm is facing a great future. Further improvements will be added. First, tags have to be combined in order to more efficiently score theoretical peptides. Second, new scoring functions should be investigated, in particular for genomic sequence matching. A latter step will be developed to correlate several peptide identifications to produce a single accurate protein identification.

*The authors would like to thank Nadine Zangger, Alexander Scherl, Patricia Palagi, Garry Corthals, Guido Bologna and Pierre-Alain Binz for their very useful contributions. This work is funded in part by a grant from Geneva Bioinformatics (GeneBio) SA.*

Received October 15, 2002

## 5 References

[1] Gras, R., Muller, M., *Curr. Opin. Mol. Ther.* 2001, *3*, 526–532.

[2] Qin, J., Fenyo, D., Zhao, Y., Hall, W. W., Chao, D. M., Wilson, C. J., Young, R. A., Chait, B. T., *Anal. Chem.* 1997, *69*, 3995–4001.

[3] Perkins, D. N., Pappin, D. D. J., Creasy, D. M., Cottrell, J. S., *Electrophoresis* 1999, *20*, 3551–3567.

[4] Bafna, V., Edwards, N., *Bioinformatics* 2001, *17*, Suppl. 1, 13–21.

[5] Eng, J. K., McCormack, A. L., Yates, I. J. R., *J. Am. Soc. Mass Spectrom.* 1994, *5*, 976–989.

[6] Field, H. I., Fenyo, D., Beavis, R. C., *Proteomics* 2002, *2*, 36–47.

[7] Eriksson, J., Chait. B. T., Fenyo, D., *Anal. Chem.* 2000, *72*, 999–1005.

[8] Pevzner, P. A., Mulyukov, Z., Dancik, V., Tang, C. L., *Genome Res.* 2001, *11*, 290–299.

[9] Mann, M., Wilm, M., *Anal. Chem.* 1994, *66*, 4390–4399.

[10] Taylor, J. A., Johnson, R. S., *Rapid Commun. Mass Spectrom.* 1997, *11*, 1067–1075.

[11] Dancik, V., Addona, T. A., Clauser, K. R., Vath, J. E., Pevzner, P. A., *J. Comput. Biol.* 1999, *6*, 327–342.

[12] Fernandez-de-Cossio, J., Gonzalez, J., Satomi, Y., Shima, T. *et al.*, *Electrophoresis* 2000, *21*, 1694–1699.

[13] Chen, T., Kao, M. Y., Tepel, M., Rush, J., Church, G. M., *J. Comput. Biol.* 2001, *8*, 325–337.

[14] Schlosser, A., Lehmann, W. D., *Proteomics* 2002, *2*, 524–533.

[15] Colorni, A., Dorigo, M., Maniezzo, V., *Proc. 1991 Europ. Conf. Artificial Life*, Elsevier, Paris 1992, pp. 134–142.

[16] Dorigo, M., Di Caro, G., Gambardella, L. M., *Artif. Life* 1999, *5*, 137–172.

[17] Bonabeau, E., Dorigo, M., Theraulaz, G., *Swarm Intelligence. From Natural to Artificial Systems*, Oxford University Press, New York 1999.

[18] Maniezzo, V., Carbonaro, A., in: Ribiero (Ed.), *Essays and Surveys in Metaheuristics,* Kluwer, New York 2001, pp. 21–44.

[19] Gras, R., Muller, M., Gasteiger, E., Gay, S. *et al., Electrophoresis* 1999, *20*, 3535–3550.

[20] Scherl, A., Couté, Y., Déon, C., Callé, A. *et al., Mol. Biol. Cell* 2002, *13*, 4100–4109.

[21] Bairoch, A., Apweiler, R., *Nucleic Acids Res.* 2000, *28*, 45–48.

[22] Koza, J. R., *Genetic Programming: on the Programming of Computers by Means of Natural Selection*, The MIT Press, Cambridge 1992.

[23] Allauzen, C., Crochemore, M., Raffinot, M., in: Pavelka, J., Tel, G., Bartosek, M. (Eds.), *Factor Oracle: a New Structure for Pattern Matching*, SOFSEM'99 1999, 1725, 291–306.

[24] Hopcroft, J. E., Ullman, J. D., *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley Publishing, Reading 1979.

# XI. REFERENCES


Aebersold R, Mann M (2003) Mass spectrometry-based proteomics. Nature 422:198-207

Aho AV, Corasick MJ (1975) Efficient string matching: An aid to bibliographic search. Comm. ACM 18:333-340

Allauzen, C., Crochemore, M., and Raffinot, M. Factor oracle: a new structure for pattern matching. 1725, 291-306. 1999. SOFSEM'99. J.Pavelka, G.Tel, and M.Bartosek.

Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ (1990) Basic local alignment search tool. J. Mol. Biol. 215:403-410

Apweiler R, Bairoch A, Wu CH (2004a) Protein sequence databases. Curr. Opin. Chem. Biol. 8:76-80

Apweiler R, Bairoch A, Wu CH, Barker WC, Boeckmann B, Ferro S, Gasteiger E, Huang H, Lopez R, Magrane M, Martin MJ, Natale DA, O'Donovan C, Redaschi N, Yeh LS (2004b) UniProt: the Universal Protein knowledgebase. Nucleic Acids Res. 32 Database issue:D115-D119

Bafna V, Edwards N (2001) SCOPE: a probabilistic model for scoring tandem mass spectra against a peptide database. Bioinformatics. 17 Suppl 1:S13-S21

Baker SG (2003) The central role of receiver operating characteristic (ROC) curves in evaluating tests for the early detection of cancer. J. Natl. Cancer Inst. 95:511-515

Bandeira N, Tang H, Bafna V, Pevzner P (2004) Shotgun protein sequencing by tandem mass spectra assembly. Anal. Chem. 76:7221-7233

Banzhaf W, Nordin P, Keller RE, Francone D (1998) Genetic Programming: An Introduction: On the automatic Evolution of Computer Programs and Its Applications.

Bartels C (1990) Fast algorithm for peptide sequencing by mass spectrometry. Biomed. Environ. Mass. Spectrom. 19:363-368

Blom N, Sicheritz-Ponten T, Gupta R, Gammeltoft S, Brunak S (2004) Prediction of post-translational glycosylation and phosphorylation of proteins from the amino acid sequence. Proteomics. 4:1633-1649

Blueggel M, Chamrad D, Meyer HE (2004) Bioinformatics in proteomics. Curr. Pharm. Biotechnol. 5:79-88

Boeckmann B, Bairoch A, Apweiler R, Blatter MC, Estreicher A, Gasteiger E, Martin MJ, Michoud K, O'Donovan C, Phan I, Pilbout S, Schneider M (2003) The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. Nucleic Acids Res. 31:365-370

Cantu, Paz. Topologies, Migrations Rates, and Multi-Population Parallel Genetic Algorithms. 99007. 1999. Illinois Genetic Algorithms Laboratory.
Ref Type: Report

Casey PJ (1995) Protein lipidation in cell signaling. Science 268:221-225

Castegna A, Thongboonkerd V, Klein JB, Lynn B, Markesbery WR, Butterfield DA (2003) Proteomic identification of nitrated proteins in Alzheimer's disease brain. J. Neurochem. 85:1394-1401

Chen T, Kao MY, Tepel M, Rush J, Church GM (2001) A dynamic programming approach to de novo peptide sequencing via tandem mass spectrometry. J. Comput. Biol. 8:325-337

Clauser KR, Baker P, Burlingame AL (1999) Role of accurate mass measurement (+/- 10ppm) in protein identification strategies employing MS or MS/MS and database searching. Anal. Chem. 71(14):2871-2882

Colinge J, Magnin J, Dessingy T, Giron M, Masselot A (2003a) Improved peptide charge state assignment. Proteomics. 3:1434-1440

Colinge J, Masselot A, Giron M, Dessingy T, Magnin J (2003b) OLAV: towards high-throughput tandem mass spectrometry data identification. Proteomics. 3:1454-1463

Craig R, Beavis RC (2004) TANDEM: matching proteins with tandem mass spectra. Bioinformatics. 20:1466-1467

Creasy DM, Cottrell JS (2004) Unimod: Protein modifications for mass spectrometry. Proteomics. 4:1534-1536

Creasy DM, Cottrell JS (2002) Error tolerant searching of uninterpreted tandem mass spectrometry data. Proteomics. 2:1426-1434

Dancik V, Addona TA, Clauser KR, Vath JE, Pevzner PA (1999) De novo peptide sequencing via tandem mass spectrometry. J. Comput. Biol. 6:327-342

Dayhoff MO, Schwartz RM, Orcutt BC (2005) Atlas of Protein Sequence and Structure. National Biomedical Research Foundation, Washington

Dongré AR, Jones JL, Somogyi A, Wysocki WH (1996) Influence of peptide composition, gas-phase basicity, and chemical modification on fragmentation efficiency: evidence for the mobile proton model. J. Am. Chem. Soc. 118:8365-8374

Dorigo M, Di Caro G (1999) The Ant Colony Optimization Meta-Heuristic. In: New Ideas in Optimization.

Elias JE, Gibbons FD, King OD, Roth FP, Gygi SP (2004) Intensity-based protein identification by machine learning from a library of tandem mass spectra. Nat. Biotechnol. 22:214-219

Eng JK, McCormack AL, Yates IIIJ (1994) An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. J. Am. Soc. Mass Spectrom. 5:976-989

Fenn JB, Mann M, Meng CK, Wong SF, Whitehouse CM (1989) Electrospray ionization for mass spectrometry of large biomolecules. Science 246:64-71

Fenyo D, Qin J, Chait BT (1998) Protein identification using mass spectrometric information. Electrophoresis 19:998-1005

Fernandez-de-Cossio J, Gonzalez J, Besada V (1995) A computer program to aid the sequencing of peptides in collision-activated decomposition experiments. Comput. Appl. Biosci. 11:427-434

Field HI, Fenyo D, Beavis RC (2002) RADARS, a bioinformatics solution that automates proteome mass spectral analysis, optimises protein identification, and archives data in a relational database. Proteomics. 2:36-47

Frank A, Tanner S, Bafna V, Pevzner P (2005) Peptide sequence tags for fast database search in mass-spectrometry. J. Proteome. Res. 4:1287-1295

Frey J, Gras R, Hernandez P, Appel RD (2003) A Hierarchical Model of Coarse-Grained Parallel Genetic Programming. 5th international conference on parallel processing and applied mathematics, Poland September 2003

Fu Y, Yang Q, Sun R, Li D, Zeng R, Ling CX, Gao W (2004) Exploiting the kernel trick to correlate fragment ions for peptide identification via tandem mass spectrometry. Bioinformatics. 20:1948-1954

Garavelli JS (2003) The RESID Database of Protein Modifications: 2003 developments. Nucleic Acids Res. 31:499-501

Gentzel M, Kocher T, Ponnusamy S, Wilm M (2003) Preprocessing of tandem mass spectrometric data to support automatic protein identification. Proteomics. 3:1597-1610

George DG, Barker WC, Hunt LT (1986) The protein identification resource (PIR). Nucleic Acids Res. 14:11-15

Giddings JC (1987) Transport, space, entropy, diffusion and flow. Elements underlying separation by electrophoresis, chromatography, field-flow fractionation and related methods. J. Chromatogr. 395:19-32

Gras R, Muller M, Gasteiger E, Gay S, Binz PA, Bienvenut W, Hoogland C, Sanchez JC, Bairoch A, Hochstrasser DF, Appel RD (1999) Improving protein identification from peptide mass fingerprinting through a parameterized multi-level scoring algorithm and an optimized peak detection. Electrophoresis 20:3535-3550

Hamm CW, Wilson WE, Harvan DJ (1986) Peptide sequencing program. Comput. Appl. Biosci. 2:115-118

Henikoff S, Henikoff JG (1992) Amino acid substitution matrices from protein blocks. Proc. Natl. Acad. Sci. U. S. A 89:10915-10919

Henzel WJ, Billeci TM, Stults JT, Wong SC, Grimley C, Watanabe C (1993) Identifying proteins from two-dimensional gels by molecular mass searching of peptide fragments in protein sequence databases. Proc. Natl. Acad. Sci. U. S. A 90:5011-5015

Heredia-Langner A, Cannon WR, Jarman KD, Jarman KH (2004) Sequence optimization as an alternative to de novo analysis of tandem mass spectrometry data. Bioinformatics. 20:2296-2304

Hernandez P, Gras R, Frey J, Appel RD (2003) Popitam: towards new heuristic strategies to improve protein identification from tandem mass spectrometry data. Proteomics. 3:870-878

Hernandez P, Mueller M, Appel RD (2005) Automated protein identification by tandem mass spectrometry: issues and strategies. Mass Spectrom Rev. In press

Hines WM, Falick AM, Burlingame AL, Gibson BW (1991) Pattern-based algorithm for peptide sequencing from tandem high energy collision-induced dissociation mass spectra. J. Am. Soc. Mass Spectrom 3:326-336

Huang L, Jacob RJ, Pegg SC, Baldwin MA, Wang CC, Burlingame AL, Babbitt PC (2001) Functional assignment of the 20 S proteasome from Trypanosoma brucei using mass spectrometry and new bioinformatics approaches. J. Biol. Chem. 276:28327-28339

Ishikawa K, Niwa Y (1986) Computer-aided peptide sequencing by fast atom bombardment mass spectrometry. Biomed. Environ. Mass Spectrom 13:373-380

James P, Quadroni M, Carafoli E, Gonnet G (1993) Protein identification by mass profile fingerprinting. Biochem. Biophys. Res. Commun. 195:58-64

Johnson RS, Biemann K (1989) Computer program (SEQPEP) to aid in the interpretation of high-energy collision tandem mass spectra of peptides. Biomed. Environ. Mass Spectrom 18:945-957

Johnson RS, Martin SA, Biemann K (1988) Collision-induced fragmentation of $(M+H)^+$ ions of peptides. Side chain specific sequence ions. Int. J. Mass Spectrom. and Ion Processes 86:137-154

Jonsson AP (2001) Mass spectrometry for protein and peptide characterisation. Cell Mol. Life Sci. 58:868-884

Kapp EA, Schutz F, Reid GE, Eddes JS, Moritz RL, O'Hair RA, Speed TP, Simpson RJ (2003) Mining a tandem mass spectrometry database to determine the trends and global factors influencing peptide fragmentation. Anal. Chem. 75:6251-6264

Karas M, Hillenkamp F (1988) Laser desorption ionization of proteins with molecular masses exceeding 10,000 daltons. Anal. Chem. 60:2299-2301

Koza JR (1992) Genetic Programming: on the programming of computers by means of natural selection. The MIT Press, Cambridge

Lane CS (2005) Mass spectrometry-based proteomics in the life sciences. Cell Mol. Life Sci. 62:848-869

Lu B, Chen T (2003b) A suboptimal algorithm for de novo peptide sequencing via tandem mass spectrometry. J. Comput. Biol. 10:1-12

Lu B, Chen T (2003a) 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

Ma B, Zhang K, Hendrie C, Liang C, Li M, Doherty-Kirby A, Lajoie G (2003) PEAKS: powerful software for peptide de novo sequencing by tandem mass spectrometry. Rapid Commun. Mass Spectrom. 17:2337-2342

Mackey AJ, Haystead TA, Pearson WR (2002) Getting more from less: algorithms for rapid protein identification with multiple short peptide sequences. Mol. Cell Proteomics. 1:139-147

Mann M, Hojrup P, Roepstorff P (1993) Use of mass spectrometric molecular weight information to identify proteins in sequence databases. Biol. Mass Spectrom 22:338-345

Mann M, Jensen ON (2003) Proteomic analysis of post-translational modifications. Nat. Biotechnol. 21:255-261

Mann M, Wilm M (1994) Error-tolerant identification of peptides in sequence databases by peptide sequence tags. Anal. Chem. 66:4390-4399

Masselot, A., Magnin, J., Giron, M., Dessingy, T., Ferriera, S., and Colinge, J. OLAV: General applicability of model-based MS/MS peptide score functions. Proc. 51st ASMS Conf. on Mass Spectrom. and Allied Topics, Montreal. 2003.

Matthiesen R, Bunkenborg J, Stensballe A, Jensen ON, Welinder KG, Bauw G (2004) Database-independent, database-dependent, and extended interpretation of peptide mass spectra in VEMS V2.0. Proteomics. 4:2583-2593

Miettinen K, (1999) Nonlinear Multiobjective Optimization. Kluwer Academic Publishers, Boston

Naimi TS, LeDell KH, Como-Sabetti K, Borchardt SM, Boxrud DJ, Etienne J, Johnson SK, Vandenesch F, Fridkin S, O'Boyle C, Danila RN, Lynfield R (2003) Comparison of community- and health care-associated methicillin-resistant Staphylococcus aureus infection. JAMA 290:2976-2984

Needleman SB, Wunsch CD (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. J. Mol. Biol. 48:443-453

Pappin DDJ, Hojrup P, Bleasby AJ (1993) Rapid identification of proteins by peptide-mass finger printing. Curr Biol 3:327-332

Pearson WR, Lipman DJ (1988) Improved tools for biological sequence comparison. Proc. Natl. Acad. Sci. U. S. A 85:2444-2448

Perkins DN, Pappin DDJ, Creasy DM, Cottrell JS (1999) Probability-based protein identification by searching sequence databases using mass spectrometry data. Electrophoresis 20:3551-3567

Pevzner PA, Dancik V, Tang CL (2000) Mutation-tolerant protein identification by mass spectrometry. J. Comput. Biol. 7:777-787

Pevzner PA, Mulyukov Z, Dancik V, Tang CL (2001) Efficiency of database search for identification of mutated and modified proteins via mass spectrometry. Genome Res. 11:290-299

Plebani M (2005) Proteomics: the next revolution in laboratory medicine? Clin. Chim. Acta 357:113-122

Reid, G. E. and S. A. McLuckey (2002) 'Top down' protein characterization via tandem mass spectrometry. J.Mass Spectrom. 37.7: 663-75.

Roepstorff P, Fohlman J (1984) Proposal for a common nomenclature for sequence ions in mass spectra of peptides. Biomed. Mass Spectrom. 11:601

Sadygov RG, Eng J, Durr E, Saraf A, McDonald H, MacCoss MJ, Yates JR, III (2002) Code developments to improve the efficiency of automated MS/MS spectra interpretation. J. Proteome. Res. 1:211-215

Sadygov RG, Yates JR, III (2003) A hypergeometric probability model for protein identification and validation using tandem mass spectral data and protein sequence databases. Anal. Chem. 75:3792-3798

Sakurai T, Matsuo T, Matsuda H, Katakuse I (1984) Paas 3: A computer program to determine probable sequence of peptides from mass spectrometric data. Biomed. Mass Spectrom. 11(8):396-399

Scarberry RE, Zhang Z, Knapp D (1995) Peptide sequence determination from high-energy collision-induced dissociation spectra using artificial neural networks. Journal of the American Society for Mass Spectrometry 6:947-961

Schaefer H, Chamrad DC, Marcus K, Reidegeld KA, Bluggel M, Meyer HE (2005) Tryptic transpeptidation products observed in proteome analysis by liquid chromatography-tandem mass spectrometry. Proteomics 5:846-852

Scherl A, Couté Y, Deon C, Calle A, Kindbeiter K, Sanchez JC, Greco A, Hochstrasser D, Diaz JJ (2002) Functional proteomic analysis of human nucleolus. Mol. Biol. Cell 13:4100-4109

Scherl A, Francois P, Bento M, Deshusses JM, Charbonnier Y, Converset V, Huyghe A, Walter N, Hoogland C, Appel RD, Sanchez JC, Zimmermann-Ivol CG, Corthals GL, Hochstrasser DF, Schrenzel J (2005) Correlation of proteomic and transcriptomic profiles of Staphylococcus aureus during the post-exponential phase of growth. J. Microbiol. Methods 60:247-257

Scopes RK (1993) Protein purification: principles and practice. Springer-Verlag, New-York

Searle BC, Dasari S, Turner M, Reddy AP, Choi D, Wilmarth PA, McCormack AL, David LL, Nagalla SR (2004) High-throughput identification of proteins and unanticipated sequence

modifications using a mass-based alignment algorithm for MS/MS de novo sequencing results. Anal. Chem. 76:2220-2230

Searle BC, Dasari S, Wilmarth PA, Turner M, Reddy AP, David LL, Nagalla SR (2005) Identification of protein modifications using MS/MS de novo sequencing and the OpenSea alignment algorithm. J. Proteome. Res. 4:546-554

Shevchenko A, Sunyaev S, Loboda A, Shevchenko A, Bork P, Ens W, Standing KG (2001) Charting the proteomes of organisms with unsequenced genomes by MALDI-quadrupole time-of-flight mass spectrometry and BLAST homology. Anal. Chem. 73:1917-1926

Spengler B (2004) De novo sequencing, peptide composition analysis, and composition-based sequencing: a new strategy employing accurate mass determination by fourier transform ion cyclotron resonance mass spectrometry. J. Am. Soc. Mass Spectrom. 15:703-714

Sunyaev S, Liska AJ, Golod A, Shevchenko A, Shevchenko A (2003) MultiTag: multiple error-tolerant sequence tag search for the sequence-similarity identification of proteins by mass spectrometry. Anal. Chem. 75:1307-1315

Tabb DL, MacCoss MJ, Wu CC, Anderson SD, Yates JR, III (2003a) Similarity among tandem mass spectra from proteomic experiments: detection, significance, and utility. Anal. Chem. 75:2470-2477

Tabb DL, Saraf A, Yates JR, III (2003b) GutenTag: High-Throughput Sequence Tagging via an Empirically Derived Fragmentation Model. Anal. Chem. 75:6415-6421

Tabb DL, Smith LL, Breci LA, Wysocki VH, Lin D, Yates JR, III (2003c) Statistical characterization of ion trap tandem mass spectra from doubly charged tryptic peptides. Anal. Chem. 75:1155-1163

Tanner S, Shu H, Frank A, Wang LC, Zandi E, Mumby M, Pevzner PA, Bafna V (2005) InsPecT: Identification of Posttranslationally Modified Peptides from Tandem Mass Spectra. Anal. Chem. 77:4626-4639

Taylor JA, Johnson RS (1997) Sequence database searches via de novo peptide sequencing by tandem mass spectrometry. Rapid Commun. Mass Spectrom. 11:1067-1075

Wilkins MR, William KL, Appel RD, Hochstrasser DF (1997) Proteome Research: New Frontiers in Functional Genomics. Springer-Verlag,

Wysocki VH, Tsaprailis G, Smith LL, Breci LA (2000) Mobile and localized protons: a framework for understanding peptide dissociation. J. Mass Spectrom 35:1399-1406

Yang XJ (2004) The diverse superfamily of lysine acetyltransferases and their roles in leukemia and other diseases. Nucleic Acids Res. 32:959-976

Yates JR, III, Griffin PR, Hood LE, Zhou JX (1991) Computer aided interpretation of low energy ms/ms spectra of peptides. In: Techniques in Protein Chemistry II. Academic Press, San Diego

Yates JR, III, Speicher S, Griffin PR, Hunkapiller T (1993) Peptide mass maps: a highly informative approach to protein identification. Anal. Biochem. 214:397-408

Zhang N, Aebersold R, Schwikowski B (2002) ProbID: a probabilistic algorithm to identify peptides through sequence database searching using tandem mass spectral data. Proteomics. 2:1406-1412

Zhang W, Chait BT (2000) ProFound: an expert system for protein identification using mass spectrometric peptide mapping information. Anal. Chem. 72:2482-2489

Zhang Z (2004) Prediction of low-energy collision-induced dissociation spectra of peptides. Anal. Chem. 76:3908-3922

Zidarov D, Thibault P, Evans MJ, Bertrand MJ (1990) Determination of the primary structure of peptides using fast atom bombardment mass spectrometry. Biomed. Environ. Mass Spectrom 19:13-26

Zubarev, R. A., et al. (2000) Electron capture dissociation for structural characterization of multiply charged protein cations. Anal.Chem. 72.3: 563-73.

今日はいい天気だったから、ずっと川で浮かんでいた。
ぼくは一生なんにも考えずに生きていけるのではないかと考えた。

Comme il faisait beau aujourd'hui,
j'ai passé tout le temps allongé dans l'eau.
Je pensais que je pouvais vivre toute ma vie
sans rien penser.

Today was a beautiful day.
I spent the whole time lying in the water.
I thought I could live my whole life
without thinking.

Figure adapted from "Monsieur Hippopotame" by Tanikawa Shuntarô