

**UNIVERSITE DES SCIENCES ET  
TECHNOLOGIES DE LILLE**

**UFR DE BIOLOGIE**

**Thèse de doctorat**

**Biologie Santé**

Présentée par

**OLIVIA JARDIN-MATHE**

**DEVELOPPEMENT D'UN LOGICIEL UNIVERSEL  
D'IMAGERIE PAR SPECTROMETRIE DE MASSE ET  
APPLICATION AU MODELE SANGSUE ET AUX  
MALADIES NEURODEGENERATIVES**

*Thèse codirigée par le Pr. Michel SALZET et le Dr. Isabelle FOURNIER*

Soutenue le vendredi 27 juin 2008

**Jury :**

**M. Michel MATHIEU**

**M. Eduardo MACAGNO**

**M<sup>me</sup> Maude PUPIN**

**M. Philippe LOPEZ**

**M. Michel SALZET**

**M<sup>me</sup> Isabelle FOURNIER**

**Rapporteur**

**Rapporteur**

**Examineur**

**Examineur**

**Examineur**

**Examineur**

# Remerciements

---

Je souhaite exprimer ma reconnaissance au Professeur Michel Salzet pour m'avoir accueillie dans son laboratoire et proposé de prolonger mon stage de DESS par cette thèse qu'il a encadrée. Il m'a fourni une expérience enrichissante durant laquelle j'ai admiré son inventivité et sa capacité de travail sans bornes.

J'associe à ces remerciements Isabelle Fournier pour avoir codirigé cette thèse avec un dynamisme et un enthousiasme quotidiens.

Egalement tous mes remerciements au Professeur Eduardo Macagno qui a accepté de juger mon travail. De plus je lui adresse aussi ma reconnaissance pour m'avoir reçue dans son laboratoire de l'Université de Californie à San Diego pendant trois mois. Il m'a permis d'approfondir mes connaissances en biologie en les mettant en pratique, ainsi que de découvrir la Californie ; ce voyage inoubliable restera l'un de mes meilleurs souvenirs et dans ce cadre, je joins Dan Lindsley à mes remerciements pour son merveilleux accueil.

J'exprime toute ma gratitude au Professeur Michel Mathieu pour avoir accepté d'être rapporteur de ma thèse, ainsi qu'à Philippe Lopez et Maude Pupin pour leur participation à ce jury.

Je félicite toute l'équipe imagerie, avec qui j'ai passé de très bons moments, pour la bonne humeur régnante, ainsi que l'ensemble des permanents du laboratoire pour leur sympathie.

Et plus particulièrement, je tiens à remercier les garçons : Maxence, Rémi, Jonathan, Julien, Karim, David et Mohamed pour leurs explications sur la spectrométrie de masse et leur endurance face à mes questions, à mes interruptions et à mes bavardages. Je n'oublie pas les filles, même si elles ont été présentes moins longtemps avec moi : Nadia, Carine, Sonia, Valérie, Amélie, Céline, Muriel et, bien sûr, Aurore. J'ajoute Annie à cette liste pour sa bienveillance et toutes les petites attentions dont elle a le secret.

Je salue la patience d'Edwige et de Marie lors de leurs relectures de ce manuscrit et les remercie pour leurs corrections judicieuses et pertinentes. Je joins à mes pensées tous les amis que j'ai délaissés afin de mener ce travail à bien ; merci de votre compréhension et de votre soutien.

Jérôme reçoit ma plus sincère reconnaissance pour son indulgence et son aide autant en ce qui concerne la bioinformatique que le reste et pour me rendre la vie plus belle.

Enfin, mes derniers remerciements s'adressent à mes parents pour leur présence et leur affection particulièrement durant ces longues études.

La spectrométrie de masse a pris une place importante au sein de la biologie. Elle est devenue un outil puissant et fiable pour la caractérisation des biomolécules, notamment dans le domaine de la protéomique. Cependant, de nouveaux développements technologiques permettant d'augmenter encore les potentialités de cette technique, que ce soit en termes de sensibilité, de rapidité ou de champs d'investigation, sont nécessaires et laissent encore de nombreuses voies d'exploration ouvertes. Ainsi, l'analyse directe de tissus par spectrométrie de masse de type MALDI est une approche en développement particulièrement importante en ce qui concerne l'analyse de biopsie ou de bloc de tissus issus de tumorothèques. Une telle technologie offre l'intérêt de s'affranchir de toutes les étapes particulièrement lourdes d'extraction, de purification et de préparation des échantillons habituellement nécessaires avec les techniques classiques de protéomique. De plus, elle permet d'obtenir une information supplémentaire en conservant la localisation des composés au sein du tissu puisque l'analyse est réalisée *in situ* par le balayage avec le faisceau laser. L'automatisation de ce balayage permet la création d'une matrice (intensité du signal versus coordonnées X/Y) ; la conversion de ces données par un logiciel bioinformatique est donc la clé de voûte de cette technologie.

L'accès aux données directes indépendamment du constructeur, la possibilité de connaître les paramètres de calcul permettant de créer l'image et la transparence ont donc été les points d'orgue pour la création d'un nouveau logiciel d'imagerie : MITICS (MALDI Imaging Team Imaging Computing System).

Pour illustrer le fonctionnement de ce logiciel, nous avons entrepris la réalisation d'études sur le protéome au niveau de l'embryon de sangsue médicinale et sa comparaison à celui de l'adulte. L'idée sous jacente était de rechercher des peptides présents chez l'embryon et se re-exprimant chez l'adulte lors de la régénération nerveuse. Ces travaux chez cet invertébré ont ensuite été confrontés aux études réalisées chez des animaux traités au 6-hydroxydopamine afin de mimer la maladie de Parkinson. La recherche de biomarqueurs et la réalisation de carte moléculaire en liaison avec la pathologie devait également permettre la corrélation entre la présence de biomarqueurs issus de la réexpression d'éléments embryonnaires.

Bien que la réalisation d'images moléculaires ait pu être effectuée, il fallait donner un nom aux peptides identifiés. Or ce travail nécessite la connaissance soit du génome, soit la possession d'EST. Dans ce contexte, j'ai participé à l'annotation des EST du système nerveux de la sangsue médicinale. Grâce à ces EST, on pourra donc à l'avenir, sur la base de MITICS, corréler l'image au nom de la biomolécule identifiée pour mieux appréhender sa fonction.

## Summary

---

Mass spectrometry settled itself as an important part of biology. It became a reliable and powerful tool for biomolecules characterization, notably in the proteomic area. Nevertheless new technological developments in order to even more increase the potentiality of this technique, regarding sensibility, rapidity or investigation fields, are necessary and let many possibilities open. So, direct analysis of tissues by MALDI mass spectrometry is a growing approach, and very important especially concerning biopsy analysis. Such a technology offers the interest to avoid all specific time and material consuming steps like extraction, purification and preparation of the sample which are common features used in proteomic techniques. Moreover it allows to get additional information by preserving composite localization in the tissue thanks to the analysis realized *in situ* by a laser beam scan. The automation of this scan enables creation of a matrix (signal intensity versus X Y coordinates); thus data conversion by bioinformatics' software is the keystone of this technology.

So access to direct data, independently from the manufacturer, accessibility to computing parameters allowing the creation of the picture and transparency have been our priorities during the creation of our new imaging software: MITICS (MALDI Imaging Team Imaging Computing System).

In order to illustrate the efficiency of this program, the survey of the medicinal leech embryos proteome versus medicinal leech adult proteome was undertaken. The underlying idea was to search peptides presents in the embryo that are re-expressed in the adult during nervous regeneration. This work on this invertebrate has next been compared to studies realized on 6-hydroxydopamin-treated animals in order to simulate Parkinson disease. Research of biomarkers and achievement of a molecular map linked with the Parkinson pathology should allow a correlation between biomarkers from nervous regeneration.

Although the creation of molecular pictures has been done, identified peptides need to be named. This work needs either knowledge of genome, or possession of EST. In this context, I participated to the annotation of medicinal leech nervous system EST. Thanks to these EST, it will later be possible with MITICS to correlate the picture to the name of the identified biomolecule in order to better apprehend its function.

# Table des matières

---

<b>INTRODUCTION</b>	<b>10</b>
<hr/>	
<b>IMAGERIE MOLECULAIRE PAR SPECTROMETRIE DE MASSE MALDI</b>	<b>13</b>
<hr/>	
<b>2.1 SPECTROMETRIE DE MASSE</b>	<b>13</b>
<b>2.2 LA DESORPTION/IONISATION LASER ASSISTEE PAR MATRICE</b>	<b>14</b>
<b>2.3 L'ANALYSEUR A TEMPS-DE-VOL (TOF)</b>	<b>16</b>
<b>2.4 L'IMAGERIE</b>	<b>19</b>
2.4.1 HISTORIQUE	19
2.4.2 PRINCIPE	19
2.4.3 IMAGERIE MALDI ET DEVELOPPEMENTS	24
2.4.4 APPLICATIONS DE L'IMAGERIE MALDI	30
<hr/>	
<b>MALDI-IMS &amp; LOGICIELS ASSOCIES</b>	<b>35</b>
<hr/>	
<b>3.1 LES OUTILS POUR L'IMS DEDIES APPLIED BIOSYSTEMS</b>	<b>37</b>
3.1.1 MMSIT	37
3.1.2 BIOMAP	38
3.1.3 4000 SERIES IMAGING	40
3.1.4 TISSUEVIEW	41
<b>3.2 LES OUTILS POUR L'IMS DEDIES BRUKER DALTONICS</b>	<b>43</b>
3.2.1 FLEXIMAGING	43
3.2.2 CREATE TARGET ET ANALYZE THIS!	45
3.2.2.1 Create Target	45
3.2.2.2 Analyze This !	49
<b>3.3 OUTILS POUR L'IMS DEDIES AUX AUTRES INSTRUMENTS</b>	<b>52</b>
3.3.1 AXIMA VIZUALISATION SOFTWARE	52
3.3.2 AXIMA2ANALYZE	52
3.3.3 IMT	53
<b>3.4 COMPARAISON DES DIFFERENTS LOGICIELS DEDIES A L'IMS</b>	<b>53</b>
<hr/>	
<b>RESULTATS &amp; DISCUSSIONS</b>	<b>55</b>
<hr/>	
<b>4.1 LE LOGICIEL MITICS</b>	<b>55</b>
4.1.1 PRE-ACQUISITION : MITICS CONTROL	56
4.1.2 POST-ACQUISITION : MITICS IMAGE	60
4.1.2.1 Premier choix : Nouveau projet	61
Dans le cas d'une analyse avec un spectromètre de masse Applied Biosystems	63
Dans le cas d'une analyse avec un spectromètre Bruker Daltonics	66
4.1.2.2 Deuxième choix : Utiliser des données déjà chargées dans la base de données	67
4.1.2.3 Troisième choix : Configuration des paramètres de la base de données	74
4.1.3 INSTALLATION DE MITICS	76
<b>4.2 APPLICATIONS BIOLOGIQUES DES DEVELOPPEMENTS BIOINFORMATIQUES</b>	<b>79</b>
4.2.1 APPLICATIONS A LA SANGSUE	79
4.2.1.1 Présentation du modèle sangsue <i>Hirudo medicinalis</i>	79
4.2.1.2 IMS des embryons de sangsue	81
4.2.2 IMAGERIE DANS LE CADRE DE LA MALADIE DE PARKINSON	85
<b>4.3 LA BANQUE EST DE SANGSUE</b>	<b>93</b>
4.3.1 PRESENTATION DU PROJET	93

4.3.2 LE CLUSTERING DES ESTS	93
4.3.2.1 Phred	94
4.3.2.2 Phrap	94
4.3.2.3 Consed	94
4.3.3 ANNOTATION	95
4.3.3.1 Vue d'ensemble	95
4.3.3.2 Leechmaster	96

---

<b>CONCLUSION</b>	<b>98</b>
-------------------	-----------

---

<b>PERSPECTIVES</b>	<b>99</b>
---------------------	-----------

---

<b>BILAN</b>	<b>102</b>
--------------	------------

---

<b>REFERENCES</b>	<b>104</b>
-------------------	------------

---

<b>ANNEXES</b>	<b>116</b>
----------------	------------

**8.1 PUBLICATIONS**

**8.2 CODE SOURCE**

## TABLE DES FIGURES

FIGURE 1: PRINCIPE DE LA SPECTROMETRIE DE MASSE .....	14
FIGURE 2: PRINCIPE DU MALDI .....	14
FIGURE 3: PRINCIPE DE LA SOURCE MALDI .....	16
FIGURE 4: PRINCIPE D'UN ANALYSEUR TEMPS DE VOL (EN MODE LINEAIRE).....	17
FIGURE 5: PRINCIPE GENERAL DE L'IMAGERIE PAR SPECTROMETRIE DE MASSE.....	21
FIGURE 6: BALAYAGE D'UNE COUPE DE CERVEAU DE RAT PAR LE FAISCEAU LASER ET POSITION DES DIFFERENTS ENDROITS OU LE LASER A FRAPPE LA PLAQUE .....	22
FIGURE 7: EXEMPLE DE SPECTRES OBTENUS EN DIFFERENTS POINTS DE LA PLAQUE POUR LA MEME MASSE <sup>25</sup> .....	23
FIGURE 8: EXEMPLE DE PLUSIEURS IMAGES OBTENUES EN CHOISSANT DIFFERENTES MASSES SUR UN SPECTRE COLLECTE LORS D'UNE ANALYSE SUR UNE COUPE DE CERVEAU DE RAT .....	23
FIGURE 9 : COMPARAISON D'IMAGES MOLECULAIRES MALDI OBTENUES A PARTIR DE COUPES DE CERVEAUX DE RAT EN 1999 ET CERVEAUX DE SOURIS EN 2008 (EQUIPE MIT, LILLE).....	24
FIGURE 10: MALDI MS IMAGE TOOL ( <a href="http://www.maldi-msi.org/">HTTP://WWW.MALDI-MSI.ORG/</a> ) .....	37
FIGURE 11 : EXEMPLE D'IMAGE RECONSTRUITE AVEC MMSIT SUR UNE COUPE DE CERVEAU DE RAT REALISEE AU LABORATOIRE. ....	38
FIGURE 12: EXEMPLES D'IMAGES RECONSTRUITES A PARTIR DE DONNEES PROVENANT DE DIFFERENTES TECHNOLOGIES ET RECONSTRUITES SOUS BIOMAP ( <a href="http://www.maldi-msi.org/">HTTP://WWW.MALDI-MSI.ORG/</a> ) .....	39
FIGURE 13 : EXEMPLE D'IMAGE RECONSTRUITE AVEC BIOMAP A PARTIR D'UNE COUPE DE CERVEAU DE RAT REALISEE AU LABORATOIRE. ....	40
FIGURE 14: INTERFACE DU LOGICIEL 4700 IMAGING .....	41
FIGURE 15 : INTERFACE DU LOGICIEL TISSUEVIEW MONTRANT DES IMAGES OBTENUES A PARTIR D'UNE COUPE SAGITTALE DE TETE DE RAT, ET LA FENETRE DE VISUALISATION DES SPECTRES. ....	42
FIGURE 16 : VISUALISATION DE REGIONS D'INTERET SUR UNE COUPE SAGITTALE DE CERVEAU DE RAT AVEC TISSUEVIEW.....	42
FIGURE 17: INTERFACE DE FLEXIMAGING PRESENTANT UNE IMAGE HISTOLOGIQUE D'UNE COUPE DE CERVEAU DE RAT REALISEE AU LABORATOIRE. ....	43
FIGURE 18: EXEMPLE DE VISUALISATION DE LA LOCALISATION DU PEPTIDE DE M/Z 1068 SUR UNE COUPE DE CERVEAU DE RAT REALISEE AU LABORATOIRE. ....	45
FIGURE 19: CREATION DE LA GRILLE AVEC CREATETARGET .....	46
FIGURE 20: VISUALISATION D'UNE COUPE DE CERVEAU DE SOURIS ADULTE SUR LA PLAQUE AVEC L'INTERFACE VIDEO DE CREATE TARGET. ....	47
FIGURE 21: VUE DE L'INTERFACE ANALYZE THIS! PRESENTEE SUR UN EXEMPLE OU DES DONNEES AVEC 45 688 POINTS PAR SPECTRE SONT CONVERTIES EN DONNEES AVEC 4 569 POINTS PAR SPECTRE.....	49
FIGURE 22: IMAGE RECOMPOSEE REPRESENTANT, A GAUCHE, UNE COUPE DE CERVEAU DE SOURIS ADULTE ET, A DROITE, L'IMAGE RECONSTRUITE AVEC ANALYZE THIS! ET BIOMAP ; EN BAS, LE SCHEMA CORRESPONDANT A L'ATLAS DU CERVEAU DE SOURIS. ....	51
FIGURE 23 : IMAGE D'UNE COUPE DE CHAINE NERVEUSE DE SOURIS (2.5 X 4 MM). LE SPECTRE MONTRE LES PICS PRESENTS A UN ENDROIT DONNE SUR LA CARTE, DETERMINE PAR L'UTILISATEUR. ....	52
FIGURE 24: INTERFACE DU LOGICIEL AXIMA2BIOMAP ( <a href="http://www.maldi-msi.org/">HTTP://WWW.MALDI-MSI.ORG/</a> ) .....	52
FIGURE 25 : IMAGES RECONSTRUITES AVEC IMT.....	53
FIGURE 26: SCHEMA DU FONCTIONNEMENT DU LOGICIEL MITICS POUR DEUX SERIES D'INSTRUMENTS DISTINCTES .....	56
FIGURE 27: INTERFACE POUR LA CREATION DU FICHIER .BIC .....	57
FIGURE 28: PARAMETRES REGLABLES PAR L'UTILISATEUR (EN ROUGE).....	58
FIGURE 29: INTERFACE POUR LA CREATION DU FICHIER .PLT .....	58
FIGURE 30: INTERFACE POUR LA CRATION DU FICHIER .PLT A L'AIDE D'UNE VUE DE LA PLAQUE.....	59
FIGURE 31: INTERFACE POUR LA CREATION DU FICHIER .SEQ.....	60
FIGURE 32: PAGE D'ACCUEIL DE MITICS POUR LA RECONSTRUCTION D'IMAGE.....	61
FIGURE 33: INTERFACE PERMETTANT DE CONVERTIR LES FICHIERS .DAT AU FORMAT .XML .....	64
FIGURE 34: INTERFACE POUR UN PROJET SUR UN APPAREIL APPLIED BIOSYSTEMS.....	65
FIGURE 35: INTERFACE POUR UN PROJET SUR UN APPAREIL BRUKER DALTONICS.....	66
FIGURE 36: INTERFACE POUR ACCEDER AUX PROJETS DEJA CHARGES DANS LA BASE DE DONNEES.....	67
FIGURE 37: AFFICHAGE DES PROJETS PRESENTS .....	67
FIGURE 38: INTERFACE RESUMANT LES DONNEES ET PERMETTANT DE CONFIGURER LES PARAMETRES DE L'IMAGE .....	68
FIGURE 39 : COMPARAISON ENTRE LES IMAGES RECREES AVEC MITICS A L'AIDE DE L'AIRE ET CELLES RECREES A L'AIDE DU MAXIMUM DU PIC SUR UNE COUPE DE CERVEAU DE RAT REALISEE AU LABORATOIRE .....	69

FIGURE 40 : EXPLOITATIONS DISTINCTES DES TROIS SPECTRES DE REFERENCE DANS LE CAS D'IMAGES RECONSTRUITES D'APRES UNE COUPE DE CERVEAU DE RAT REALISEE AU LABORATOIRE. ....	70
FIGURE 41: INTERFACE PRESENTANT L'IMAGE RECONSTRUITE.....	71
FIGURE 42: PALETTE DES JEUX DE COULEURS PREDEFINIS .....	72
FIGURE 43: IMAGE RECONSTRUITE SELON UNE ECHELLE MONOCHROMATIQUE .....	73
FIGURE 44: INTERFACE PERMETTANT DE CONFIGURER LES PARAMETRES DE LA BASE DE DONNEES .....	74
FIGURE 45: STRUCTURE DE LA BASE DE DONNEES .....	75
FIGURE 46: PREMIERE FENETRE DE L'INTERFACE D'INSTALLATION DU LOGICIEL .....	76
FIGURE 47: CHOIX DU REPERTOIRE D'INSTALLATION DU LOGICIEL .....	77
FIGURE 48: BILAN DES CHOIX D'INSTALLATION .....	77
FIGURE 49 : EVOLUTION DU NOMBRE DE PUBLICATIONS EN IMAGERIE MALDI DEPUIS SON INTRODUCTION EN 1997 JUSQU'AU MOIS DE MAI 2008.....	78
FIGURE 50: <i>HIRUDO MEDICINALIS</i> ET SA CLASSIFICATION PHYLOGENETIQUE .....	79
FIGURE 51: LE SYSTEME NERVEUX D' <i>HIRUDO MEDICINALIS</i> <sup>110</sup> .....	80
FIGURE 52: GANGLIONS D' <i>HIRUDO MEDICINALIS</i> (LABORATOIRE DE NEUROIMMUNOLOGIE DES ANNELIDES) .....	81
FIGURE 53: IMAGE DE L'EMBRYON AVANT DEPOT DE LA MATRICE.....	82
FIGURE 54 : IMAGES D'EMBRYON DE SANGSUE RECONSTRUITES POUR M/Z=3507 ET 6443 AVEC FLEXIMAGING (B ET C) PUIS AVEC MITICS (D ET E) DANS TROIS CONDITIONS DIFFERENTES .....	83
FIGURE 55 : IMAGES MOLECULAIRES RECONSTRUITES A PARTIR D'UNE TRANCHE D'UN METAMERE D'UNE SANGSUE ADULTE .....	84
FIGURE 56 : REPARTITION DES IONS TRYPSIQUES DE CRMP2 (SOMME DES M/Z 643, 728, 813, 1083 ET 1460), SCAN DE LA COUPE AVANT ET APRES DEPOT DE LA TRYPSINE, IMAGE DU PLAN DE COUPE SELON L'INDEX BREGMA ET IMAGE DE LA REPARTITION DES ARNm PAR HYBRIDATION IN SITU SELON LE ALLAN BRAIN ATLAS .....	85
FIGURE 57 : IMAGES DE LA REPARTITION DES IONS DE L'UBIQUITIN SONT INCLUS : LE SCAN DE LA COUPE, L'IMAGE DU PLAN DE COUPE SELON L'INDEX BREGMA, IMAGE DE LA REPARTITION DES ARNm PAR HYBRIDATION IN SITU SELON ALLEN BRAIN ATLAS .....	88
FIGURE 58 : COMPARAISON DE LA REPARTITION DES IONS DE LA VIMENTINE ENTRE FLEXIMAGING AVEC OU SANS NORMALISATION ET MITICS ET SCAN DE LA COUPE. ....	89
FIGURE 59 : COMPARAISON DE LA REPARTITION D'UN ION DE L'APOLIPOPROTEINE E ENTRE FLEXIMAGING AVEC OU SANS NORMALISATION ET MITICS. SONT INCLUS LE SCAN DE LA COUPE ET DES GROSSISSEMENTS DES AIRES DE REPARTITION DU PEPTIDE .....	90
FIGURE 60 : IMAGES MOLECULAIRES AVEC MITICS DE LA REPARTITION DE DEUX IONS DE L'APOLIPOPROTEINE E EN PRENANT EN COMPTE SOIT L'AIRES, SOIT L'INTENSITE MAXIMALE DU PIC.....	91
FIGURE 61 : IMAGES MOLECULAIRES AVEC MITICS DE LA REPARTITION D'UN ION DE L'ACTINE EN PRENANT EN COMPTE SOIT L'AIRES, SOIT L'INTENSITE MAXIMALE DU PIC .....	91
FIGURE 62 : REPRESENTATION SCHEMATIQUE DU FONCTIONNEMENT DU LOGICIEL PHRED .....	94
FIGURE 63: PAGE D'ACCUEIL DU SITE LEECHMASTER.UCSD.EDU.....	97

## Liste des tableaux

TABLEAU 1 : MATRICES LES PLUS COURAMMENT UTILISEES EN SPECTROMETRIE DE MASSE MALDI-TOF POUR L'ANALYSE DE COMPOSES BIOLOGIQUES ET SPECIALEMENT DES PEPTIDES ET PROTEINES. ....	27
TABLEAU 2 : TABLEAU COMPARATIF DES DIFFERENTS LOGICIELS D'IMAGERIE PAR SPECTROMETRIE DE MASSE .....	<b>ERREUR ! SIGNET NON DEFINI.</b>
TABLEAU 3: DIFFERENTS BIOMARQUEURS IDENTIFIES PAR IMAGERIE MALDI DANS LE CADRE DE L'ANALYSE DE RATS INJECTES 6-OHDA ET PAR COMPARAISON AVEC DES RATS TEMOINS, AINSI QUE LEUR SEQUENCE ET LES PEPTIDES ISSUS DE LA DIGESTION ENZYMATIQUE PAR LA TRYPSINE IN SILICO .....	87

## Abbreviations

---

<b>6OHDA :</b>	6-Hydroxydopamine
<b><math>\alpha</math>-CHCA :</b>	Acide $\alpha$ -Cyano-4-hydroxycinnamique
<b>ABI (ou AB):</b>	Applied Biosystems
<b>ADN/ARN :</b>	Acide Désoxybonucléique/ Ribonucléique
<b>BD :</b>	Bruker daltonics
<b>CGI :</b>	Common Gateway Interface
<b>CRMP2 :</b>	Collapsin Response Mediator Protein 2
<b>CT :</b>	Computed Tomography
<b>DESI:</b>	Desorption ElectroSpray Ionization
<b>DHB:</b>	Acide 2,5-Dihydroxybenzoïque
<b>Ec:</b>	Energie cinétique
<b>ESI:</b>	ElectroSpray Ionisation
<b>EST:</b>	Expressed Sequenced Tag
<b>FAB:</b>	Fast Atom Bombardement
<b>FFPE :</b>	Formalin Fixed and Paraffine Embedded
<b>GUI:</b>	Graphical User Interface
<b>HTML:</b>	HyperText Markup Language
<b>HTTP:</b>	HyperText Transfer Protocol
<b>IDL:</b>	Interactive Data Language
<b>IHC :</b>	Immunohistochimie
<b>IRM :</b>	Imagerie par résonance magnétique (ou MRI)
<b>IMS:</b>	Imaging Mass Spectrometry (Imagerie par Spectrométrie de Masse)
<b>IRM :</b>	Imagerie par Résonance Magnétique
<b>ITO:</b>	Idium-Tin Oxyde (ou oxyde d'indium-étain)
<b>LDI:</b>	Laser Desorption Ionization
<b>m/z:</b>	Rapport Masse sur Charge
<b>MALDI:</b>	Matric Assisted Laser Desorption Ionization
<b>MIT:</b>	MALDI Imaging Team
<b>MITICS:</b>	MIT Imaging Computing System
<b>MMSIT:</b>	MALDI Mass Spectrometry Imaging Tool
<b>MS/MS:</b>	Spectrométrie de Masse en Tandem
<b>NC :</b>	non communiqué
<b>ODBC:</b>	Open Database Connectivity
<b>PAF:</b>	Paraformaldéhyde
<b>PCA:</b>	Analyse en Composantes Principales
<b>PET:</b>	Tomographie par emission de positons
<b>PHP :</b>	Hypertext Preprocessor
<b>PSD:</b>	Post Source Decay
<b>ROI:</b>	Regions Of Interest
<b>RT:</b>	Run Time
<b>Sa:</b>	Acide sinapinique
<b>SIMS:</b>	Secondary Ion Mass Spectrometry
<b>SQL:</b>	Structured Query Language
<b>TDC :</b>	Time to Digital Converter
<b>TOF:</b>	Temps de vol
<b>UV:</b>	Ultraviolet
<b>XML:</b>	eXtensible Markup Language

---

## Introduction

La Science peut être définie comme un ensemble cohérent de connaissances relatives à certains faits ou d'objets ou de phénomènes obéissant à des lois, et les découvertes scientifiques qui en découlent sont le fruit de réflexions et d'observations. En effet, l'analyse de l'histoire des Sciences révèle ce mélange ingénieux de différents éléments tels que la réflexion initiée par l'homme et les avancées technologiques aboutissant à de nouvelles observations qui ont permis d'accroître les connaissances scientifiques.

Ainsi tout au long de l'histoire, depuis la Grèce antique jusqu'à ce jour, les observations scientifiques ont initié les découvertes les plus significatives. De très nombreux exemples jalonnent notre histoire avec Buffon, Linnée, Darwin qui, de l'observation, ont élaboré la théorie de l'évolution, ou avec Watson et Crick qui, d'une image au rayon X d'ADN cristallisé obtenue par Rosalind Franklin, ont découvert la structure de l'ADN en 1952.

Les hommes ont, de cette façon, développé de tout temps des méthodes descriptives basées sur l'observation et les technologies associées. Tous les domaines scientifiques, de la Physique, à la chimie, à la biologie ou à la médecine, ont émergé par l'intermédiaire des avancées techniques d'observation. De la première loupe développée en 1664 par Antoine van Leeuwenhoek, aux premiers IRM en 1980 par Paul Lauterbur et Peter Mansfield (Nobel, 2003), elles ont permis d'établir de nouveaux dogmes sans cesse remis en question. Les techniques d'imagerie du vivant ont depuis accompagné le scientifique dans la plupart de ses investigations.

La constante évolution de ces techniques d'imagerie a permis de repousser toujours plus loin les limites de l'observation, comme l'illustre la progression de la microscopie optique vers la microscopie électronique. La gamme de leur analyse s'étend largement allant de la visualisation d'organes entiers (IRM) à l'acquisition de données topographiques de surfaces jusqu'à une échelle atomique (microscopie à force atomique, électronique...); leur progression a conduit à la perception de molécules au sein des tissus. Cependant, afin d'obtenir ces résultats, la plupart de ces techniques requièrent une présélection des molécules avec des marqueurs fluorescents ou des anticorps, nécessitant la connaissance à l'avance de la molécule d'intérêt à suivre. Malgré l'abondance des informations fournies, ces approches peuvent pâtir d'un manque de sensibilité et de spécificité.

Idéalement, la technique d'imagerie parfaite serait définie par la détection et l'identification simultanées de multiples composés connus et inconnus présents dans les tissus biologiques, représentés avec une résolution spatiale approchant la taille d'une cellule tout en tenant compte du fait que les composés d'intérêt varient grandement en taille, en propriété physicochimique, en concentration ou en structure. Cette description traduit l'espoir qui repose sur l'imagerie par spectrométrie de masse. En effet, la spectrométrie de masse est devenue, à l'heure actuelle, un outil incontournable, en particulier dans le domaine de la

protéomique comme en témoignent les nombreuses recherches menées et publiées. Des études ont mené à la caractérisation sur de nombreuses protéines provenant de différents organismes, voire de différents organes ou de cellules dans des états physiologiques distincts. De plus, il a pu être mis en évidence des variations différentielles de l'expression sous ou sur-modulée de protéines entre différents états physiologiques. Cependant, l'inconvénient majeur de cette stratégie réside dans la nécessité d'homogénéiser l'échantillon avant extraction puis séparation et caractérisation par spectrométrie de masse. Cela conduit souvent à la mise en évidence de protéines majeures comme les protéines du cytosquelette ou du métabolisme aux dépens de composés faiblement exprimés dont le signal peut être irrémédiablement perdu. De plus, ces techniques ne permettent pas de conserver les informations concernant la localisation des molécules au sein du tissu d'origine.

L'analyse directe sur coupe a été le premier pas vers l'imagerie car elle peut pallier à certains des problèmes présentés puisqu'elle permet la détection de molécules faiblement exprimées tout en gardant leur localisation<sup>1-3</sup>. Elle répond aussi à beaucoup des exigences concernant la technique d'imagerie idéale. En effet la spécificité chimique d'une molécule dépend d'un seul paramètre universel qui définit sa composition atomique : sa masse. Ainsi aucun marquage préalable ne s'impose. De plus la localisation spatiale de plusieurs centaines de composés dans un échantillon complexe tel qu'une coupe de tissu devient possible avec cette technique. Toutefois, toute technologie possède des limites ; en effet un certain nombre de contraintes apparaissent. Les grosses protéines de plus de 50 kDa, les chaînes d'oligonucléotides sont difficiles à détecter du fait de leur faible ionisation, ou de la sensibilité des détecteurs pour les hautes masses.

L'imagerie par spectrométrie de masse est donc une automatisation de l'analyse directe sur coupes avec une représentation sur une image des éléments moléculaires retrouvés sur la coupe. Cette représentation aurait pu être matricielle (l'intensité du pic en fonction des coordonnées X et Y), mais la représentation sous forme d'image permet de visualiser et conceptualiser cette matrice. Cette image moléculaire est donc dépendante de la conversion des données par un logiciel spécifique dédié et dépendant de cette technologie.

Finalement l'étude du monde qui nous entoure et sa compréhension qui en découle, aussi anciennes soient-elles, sont donc les conclusions d'observations évolutives de plus en plus fines, sans cesse remises à jour. Cette description n'a pu être possible que par la création et l'utilisation de nouveaux outils technologiques supports de ces observations. Le tout premier de ces outils fut le microscope. L'accès à l'observation de tissus a permis à la biologie de faire une avancée extraordinaire vers l'élément essentiel qui les compose *i.e.* la cellule et son environnement. L'image regardée au travers de l'objectif du microscope est donc devenue support et initiateur des théories qui ont bouleversé la biologie descriptive vers la biologie cellulaire et moléculaire.

L'image, qu'elle soit issue du microscope ou d'un l'appareil photographique, demeure la représentation instantanée de tout événement qu'il soit cellulaire ou moléculaire ou sociétal. Ainsi elle se révèle dans la société le vecteur de ses contrastes, ses peurs, ses paroxysmes et

surtout sa réalité. L'image, normalement, ne triche pas et reflète une vérité instantanée ; alors la question suivante se pose : Est-ce que l'imagerie MALDI reflète réellement la nature et la composition moléculaire des molécules présentes dans la coupe sans distorsion liée à l'analyse bioinformatique ? Il nous faut donc désormais des logiciels ouverts permettant une totale compréhension des événements permettant la conversion des données machines vers l'image. C'est dans ce contexte, que cette thèse a été réalisée et a abouti à un logiciel open source, MITICS (MALDI Imaging Team Imaging Computing System).

Ce mémoire s'articulera en cinq parties.

Le premier chapitre présentera le cadre général de la thèse : il introduira la spectrométrie de masse, et plus précisément la technique de désorption ionisation laser assistée par matrice (MALDI) puis il abordera les principes et les applications de l'imagerie par spectrométrie de masse de type MALDI. Un état de l'art sera exposé dans une deuxième partie, reprenant les différents logiciels existants dans le domaine de l'imagerie par spectrométrie de masse. La troisième partie portera sur les résultats de la thèse *i.e.* la mise au point d'un nouveau logiciel d'imagerie par spectrométrie de masse universel et ses applications dans le cadre des études de la neurogénèse sur l'embryon de la sangsue médicinale et dans le cadre de la maladie de Parkinson chez un modèle rat traité au 6OHDA. En vue de faciliter l'identification des peptides détectés par cette méthode chez *Hirudo medicinalis*, l'annotation d'EST de ce modèle, liée au couplage de l'interrogation de la base de données pour obtenir les séquences d'intérêt, sera ensuite présentée. Enfin la dernière partie clôturera ce mémoire avec les conclusions et perspectives.

# Imagerie moléculaire par spectrométrie de masse MALDI

A l'heure où l'identification, la caractérisation et la localisation des protéines et de leurs partenaires devient un enjeu majeur pour l'élucidation des mécanismes cellulaires et moléculaires tout comme l'identification des voies de signalisation, les techniques d'imagerie du vivant deviennent un atout majeur. Cette dernière décennie a vu s'accroître considérablement les différentes technologies dans ce domaine permettant la localisation et la co-localisation des protéines. Plus récemment l'imagerie MALDI est venue s'adjoindre aux techniques bien connues d'Hybridation *in situ* et d'immunohistochimie. L'imagerie MALDI offre le très net avantage d'être une technique moléculaire non ciblée permettant d'obtenir une analyse reflétant la composition locale d'un tissu. Elle permet donc de réaliser une approche protéomique directement à partir d'un tissu tout en évitant les étapes d'extraction et de séparation et en ajoutant une dimension de localisation aux données classiquement obtenues. La force de l'imagerie MALDI repose sur les atouts de la spectrométrie de masse dont la capacité séparative, la sensibilité, la flexibilité et la robustesse.

## 2.1 Spectrométrie de masse

La spectrométrie de masse est une technique permettant d'obtenir la masse moléculaire d'un composé par mesure du rapport masse sur charge ( $m/z$ ) d'ions en phase gazeuse. Ainsi le spectre de masse résultant présente l'intensité du courant mesuré (en abscisse) en fonction du rapport  $m/z$  (en ordonnée). Avec l'amélioration et la diversification des technologies, des informations structurales sur le composé étudié peuvent également être obtenues.

Tout spectromètre de masse se compose de trois parties ayant chacune sa fonction propre (Figure 1) : la source, l'analyseur et le détecteur. La source de l'instrument a pour but d'obtenir à partir de l'échantillon initial des ions en phase gazeuse. L'analyseur, quant à lui, assure le tri des ions formés en fonction de leur rapport  $m/z$ . La troisième et dernière partie est constituée par le détecteur qui permet de convertir les ions, ou courant ionique, en un courant électronique. Par la suite, les informations sont transférées vers le système d'exploitation informatique permettant le traitement des données enregistrées.

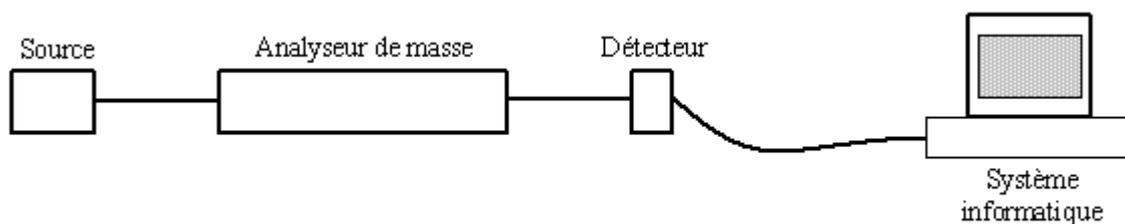


Figure 1: Principe de la spectrométrie de masse

## 2.2 La Désorption/Ionisation Laser Assistée par Matrice

La spectrométrie de masse vise tout d'abord à obtenir la masse moléculaire des composés d'intérêt. Dans ce contexte les molécules doivent rester intactes, soit ne pas subir de fragmentations. Pour éviter les collisions sur des molécules de gaz résiduelles, les instruments sont donc placés sous un vide poussé. Néanmoins celles-ci ne constituent pas la seule source de fragmentation. L'énergie interne que possèdent les ions en phase gazeuse conduit également à des dissociations. Elle dépend de la méthode de production d'ions (c'est-à-dire la source) utilisée, et n'est pas toujours contrôlable. Dans le cas où les ions après formation possèdent peu d'énergie interne, ils restent intacts jusqu'au détecteur. La méthode de production des ions est alors dite "douce". A ce titre deux méthodes se sont rapidement dégagées comme particulièrement douces, à savoir l'ESI (ElectroSpray Ionisation) et le MALDI (Matrix Assisted Laser Desorption Ionization).

Le MALDI ou Désorption Ionisation Laser Assistée par Matrice, introduit en 1988 par Hillenkamp et Karas en Allemagne <sup>4, 5</sup> et sous une autre forme par Tanaka et collaborateurs au Japon <sup>6</sup> a révolutionné le domaine de l'analyse des biomolécules par spectrométrie de masse tout comme quelques années auparavant les sources ESI. En effet le MALDI permet d'obtenir des ions en phase gazeuse à partir de composés de polarités et masses variées et peut atteindre des composés de très hautes masses et très polaires comme les protéines.

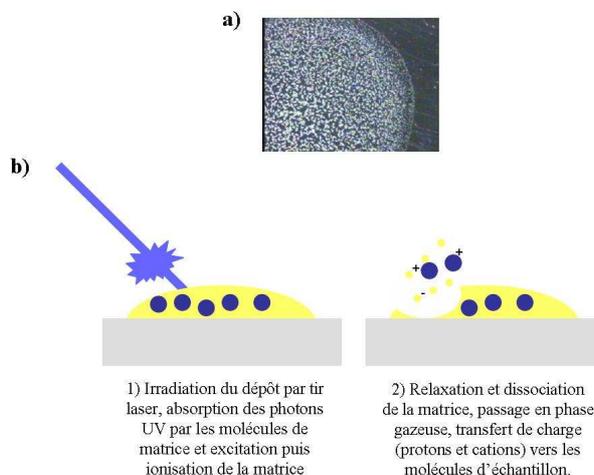


Figure 2: Principe du MALDI

Le MALDI est une technique de désorption basée sur l'irradiation d'un échantillon solide par un faisceau de photons incidents produit par un laser. Afin d'assurer un transfert d'énergie efficace, un petit composé organique nommé matrice est utilisé et co-cristallisé en excès avec les analytes de manière à obtenir une solution solide. La matrice est donc choisie afin de présenter une bande d'absorption à, ou à proximité de, la longueur d'onde du laser utilisé, mais cette condition nécessaire est non suffisante. La co-cristallisation par évaporation du solvant entraîne la formation de micro-cristaux de matrice (Figure 2a) dans le réseau cristallin desquels se seront inclus les analytes. Le dépôt est en général réalisé sur un support conducteur (pour ne pas altérer les champs électriques de la source) qui peut être transféré dans le spectromètre de masse où il sera alors soumis à l'irradiation par le faisceau laser pulsé. Dans des conditions classiques, les MALDI utilisés fonctionnent avec des lasers émettant dans l'UV des photons très énergétiques (UV-MALDI). Historiquement la grande majorité des instruments étaient équipés de laser N<sub>2</sub> émettant à 337nm. La faible durée de vie de ces laser ainsi que leur limitation en fréquence de répétition des tirs lasers ont conduit ces dernières années à revenir vers des laser solides, dont la troisième harmonique est utilisée pour obtenir une longueur d'onde de 355 nm, présentant une durée de vie bien supérieure et pouvant atteindre 200 Hz de fréquence de répétition des tirs laser. Ces lasers sont, donc, logiquement bien mieux adaptés au domaine de la protéomique à haut débit ou aux nouvelles techniques émergentes comme l'imagerie MALDI. Lors de l'irradiation du dépôt par le faisceau laser, les photons émis sont absorbés par les molécules de matrice, et induisent des transitions électroniques vers des états excités au sein du solide. La relaxation de l'énergie au sein de ce solide conduira à l'éjection en phase gazeuse d'une partie du matériau, soit la matrice, entraînant avec elle l'analyte, le tout sous forme d'un plasma qui va s'étendre rapidement en phase gazeuse sous vide. Ainsi, le processus de désorption/ionisation conduit très rapidement après le tir laser (échelle de la nanoseconde) à l'obtention d'ions en phase gazeuse (Figure 2.b). Malheureusement ce processus ne conduit à l'observation que d'un faible pourcentage d'ions sur la quantité totale de matière éjectée. Il faut également noter que les ions formés sont très divers et comprennent des ions positifs et négatifs ainsi que des ions de matrice et d'analyte (Figure 3).

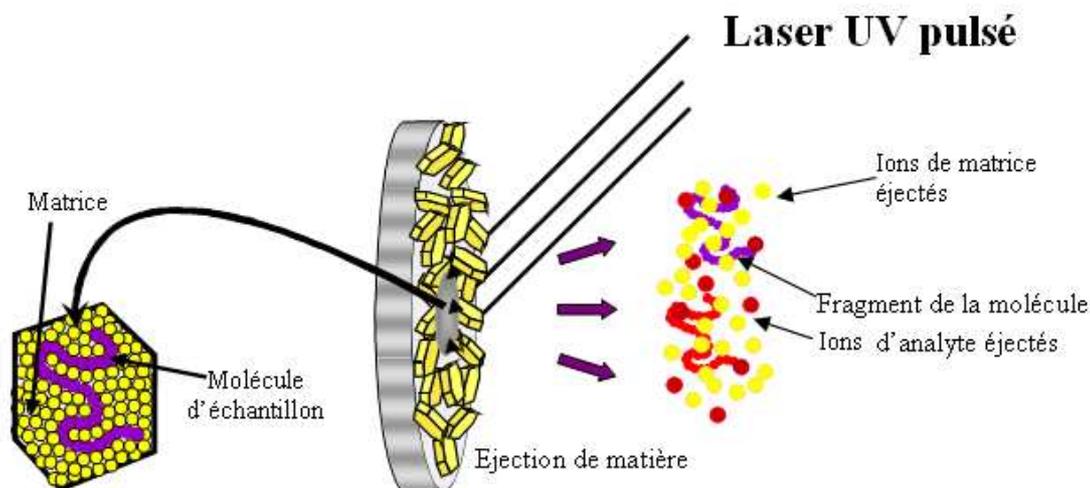


Figure 3: Principe de la source MALDI

En MALDI, les ions formés sont principalement monochargés, majoritairement sous la forme de molécules protonées  $[M+H]^+$  même si une petite partie des espèces se retrouvera sous la forme d'espèces cationisées par des sels, par exemple par le sodium  $[M+Na]^+$  ou par le potassium  $[M+K]^+$ , et cela d'autant plus que l'analyte présente une affinité pour ces cations et que ces derniers sont présents à l'état de trace dans les préparations. Si les ions sont principalement monochargés, des ions multichargés sont néanmoins observables en MALDI et les états de charge seront d'autant plus prépondérants que la masse de l'analyte croîtra.

Les ions produits par le processus de désorption/ionisation en MALDI possèdent des énergies internes suffisamment faibles pour être, dans la grande majorité des cas, stables sur la durée de l'analyse. Cette caractéristique permet l'analyse de nombreuses familles de molécules sous leur forme intacte et jusqu'à de hauts poids moléculaires. Cette technique est donc bien adaptée à l'analyse des différentes biomolécules composant les cellules telles que les peptides, les protéines, les sucres et les lipides et dans une certaine mesure les oligonucléotides.

## 2.3 L'analyseur à temps-de-vol (TOF)

De par son mode de fonctionnement pulsé et sa caractéristique à former peu d'ions multichargés, les analyseurs à Temps-de-Vol (TOF) sont les analyseurs les plus facilement et logiquement couplés aux sources MALDI. Les spectromètres de masse utilisés au laboratoire sont dotés d'un analyseur temps-de-vol (Time-of-Flight ou TOF) <sup>7</sup>. Le principe de fonctionnement des TOF repose sur la mesure du temps (temps-de-vol) nécessaire aux ions pour arriver jusqu'au détecteur après accélération dans un champ électrique constant et séparation dans une région libre de champ (tube de vol) de ces derniers en fonction de leur rapport masse-sur-charge ( $m/z$ ).

Les analyseurs TOF sont donc constitués de deux parties (Figure 4):

1-La zone d'accélération dans laquelle règne le champ électrique constant. Elle est physiquement délimitée d'un côté par la plaque métallique sur laquelle a été déposé l'échantillon et de l'autre par une électrode d'extraction. Dans cette zone les ions sont soumis à une différence de potentiel importante (en général 20-25 kV) sur une courte distance (quelques mm).

2-Le tube de vol qui s'étend de la grille d'extraction au détecteur. Dans cette région aucun champ électrique n'est appliqué (la différence de potentiel entre l'électrode d'extraction et l'électrode se situant juste devant le détecteur est nulle), elle est donc dite libre de champ.

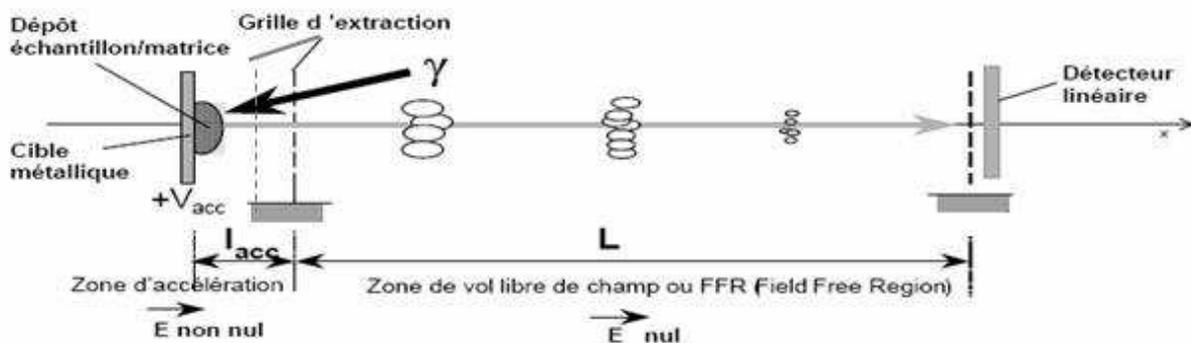


Figure 4: Principe d'un analyseur Temps de vol (en mode linéaire)

Les ions sont donc soumis à l'accélération dès leur éjection et acquièrent une vitesse en sortie de la région d'accélération inversement proportionnelle à la racine carrée de leur rapport  $m/z$ . Ils évoluent alors librement dans la zone libre de champ (appelée aussi tube de vol), continuant leur trajectoire à vitesse constante, et se séparent progressivement dans le tube de vol, en fonction de  $m/z$ , jusqu'à leur arrivée au détecteur. Ils parviennent ainsi dans l'ordre des  $m/z$  croissants au détecteur. La mesure du temps de vol des ions est effectuée en prenant un signal de départ synchronisé sur le tir laser et un signal d'arrivée lorsque l'ion atteint le détecteur. La relation existant entre le temps de vol et la grandeur  $m/z$  permet, si les paramètres géométriques de l'instrument ainsi que les conditions d'accélération sont connus avec précision, de recalculer la valeur de  $m/z$ . Donc un spectre de masse est obtenu par reconversion d'un spectre en temps enregistré sur la base d'une unité de mesure. Ainsi, le courant en sortie du détecteur sera mesuré sur un intervalle de temps spécifié. Plus l'intervalle est petit et mieux le spectre sera décrit. La principale limitation aux mesures est la capacité de stockage des oscilloscopes ou s'il s'agit d'un convertisseur TDC (Time to Digital) sa rapidité de conversion. Ainsi, en général il faut sacrifier sur la précision de mesure pour augmenter la gamme de masse analysable. Les données générées correspondent donc en réalité à des histogrammes qui reportent le courant moyen mesuré sur l'intervalle de temps considéré. Les données sont par la suite présentées sous forme de courbe pour une facilité de visualisation des données.

Le principal défaut des analyseurs TOF a longtemps été leur manque de résolution et notamment dans le cas d'ions provenant de sources de désorption/ionisation. Ainsi au cours des années différents systèmes permettant d'augmenter la résolution se sont développés.

Parmi ceux-ci se trouvent les systèmes dits d'extraction retardée développés en 1955 par Wiley et Mc Laren<sup>8</sup>. Dans ces systèmes les ions évoluent librement pendant une période de quelques centaines de nanosecondes avant d'être accélérés. Il est ainsi possible d'obtenir une compensation des disparités spatiales et d'énergie cinétique des ions issus de la source de l'instrument en recréant un plan de focalisation des ions dans le tube de vol. L'ajustement des paramètres de tension et du temps de retard à l'extraction permettent pour un  $m/z$  donné de repousser ce plan sur la surface du détecteur.

Des systèmes dits de miroir électrostatique ou réflecteur ont également été développés par Mamyrin et Karataev dans les années 1970<sup>9, 10</sup> pour compenser les dispersions en énergie cinétique des ions. Ces systèmes consistent en une série d'électrodes permettant de créer un champ électrique axial extrêmement uniforme mais s'opposant au mouvement des ions dans le tube de vol. Les ions vont donc pénétrer dans le miroir, ralentir sous l'effet du champ électrique inverse puis s'arrêter et être réaccélérés en sens inverse. Le miroir agit comme un filtre en énergie cinétique et les ions de même  $m/z$  mais d'énergie cinétique ( $E_c$ ) différente se réfléchiront donc sur des équipotentielles différentes, l'ion d' $E_c$  la plus grande allant le plus profondément dans le miroir. Ainsi en respectant certaines conditions géométriques sur l'instrument il est alors possible d'assurer la refocalisation des ions sur un même plan, celui du détecteur.

Les instruments actuels présentent en général une combinaison de ces deux systèmes, l'extraction retardée permettant en fait de convertir des disparités spatiales en disparités d' $E_c$  qui sont alors compensées par l'utilisation du réflecteur si nécessaire. Le mode linéaire extraction retardée est donc bien moins résolutif que le mode réflecteur avec extraction retardée mais offre en revanche une plus grande sensibilité. Il n'est donc pas rare d'effectuer des analyses dans les deux modes. De plus, sur les appareils commerciaux, il est rarement possible d'accéder à une gamme de masse supérieure à  $m/z$  10 000 en mode réflecteur.

Les analyseurs TOF sont donc très avantageux de par leur sensibilité et leur gamme de masse illimitée et présentent à l'heure actuelle une bonne résolution. Ils sont donc parfaitement adaptés pour les sources MALDI et sont souvent utilisés pour les analyses protéomiques.

## 2.4 L'imagerie

### 2.4.1 Historique

La facilité d'introduction d'échantillons solides dans la source MALDI a conduit dès le début des années 1990 à utiliser cette méthode pour l'analyse d'échantillons bruts, tout comme pour d'autres méthodes de désorption avant elle (LDI, FAB, SIMS, ...). La spécificité du MALDI par rapport aux autres techniques de désorption étant sa capacité d'analyse des composés polaires et de haute masse, l'analyse de peptides/protéines à partir de tissus (organes, organelles, sections) a donc constitué l'un des premiers domaines explorés à l'aide de cette technique. Ainsi l'analyse directe par MALDI d'échantillons biologiques bruts s'est rapidement révélée être un outil très efficace pour répondre à certaines problématiques particulières, la capacité d'analyse d'un échantillon en différentes localisations permettant d'ajouter une dimension de régionalisation aux expériences classiquement menées.

Les premières études ont été réalisées chez un modèle invertébré la Lymnée par l'équipe de K. Li et C. Jimenez<sup>11, 12, 13, 14</sup> puis par l'équipe de J. Sweedler sur les céphalopodes<sup>15</sup>. Les études ont porté sur la comparaison de l'expression des neuropeptides dans différentes régions du cerveau de ces animaux *e.g.* le soma et les terminaisons axonales au niveau de l'aire neurohémale<sup>16-18, 13, 19</sup>. Ces travaux ont été étendus à d'autres modèles invertébrés comme les crustacés<sup>19, 20</sup>, les insectes<sup>21</sup> puis les vertébrés<sup>3, 22</sup>.

De l'analyse directe de tissus par MALDI à l'imagerie MALDI ne restait qu'un pas. Il fut franchi par B. Spengler en 1994<sup>23</sup> en montrant les premières images d'un neuropeptide : la substance P, à l'aide d'un microscope confocal dans lequel un laser venait balayer la surface de l'échantillon. Puis, en 1997, R.M. Caprioli et coll.<sup>22</sup>, réalisèrent une automatisation de la technique, suivie d'un traitement bioinformatique des données pour obtenir une représentation sous forme d'images de coupes de tissus. Depuis 1997, l'imagerie MALDI a progressé dans de nombreux domaines, que ce soit sur le plan instrumental, analytique ou encore bioinformatique ; et les applications de cette technologie sont de plus en plus nombreuses.

### 2.4.2 Principe

La spectrométrie de masse MALDI permet de réaliser l'analyse des composés présents en un point défini d'une surface. Même si les processus MALDI restent encore mal connus, des études fondamentales ont permis de démontrer que l'éjection de matière consécutivement à l'irradiation par le faisceau laser ne se produit que sur une aire correspondant approximativement à la zone d'intersection du faisceau incident et de l'échantillon, soit que les voies de dissipation de l'énergie au sein du solide se font principalement sur la profondeur du cristal. Il est donc possible, dans ces conditions, d'analyser une aire définie d'un tissu par exemple. L'obtention d'une surface au maximum plane est nécessaire à l'analyse puisque toute variation d'épaisseur du tissu entraînera des variations du TOF mesuré des ions et donc des  $m/z$  correspondants pouvant rendre le traitement des données difficile. Ainsi, si l'objet à

analyser n'est pas originellement une surface plane, il devra être sectionné. Dans le domaine de la biologie animale ou végétale (en particulier animale dans notre cas) les techniques classiquement utilisées pour l'histologie (e.g. cryostat, microtome) fournissent un excellent moyen de produire ce type de surfaces. En MALDI, l'analyse sera promue au travers de l'utilisation de la matrice qui, dans ce cas, sera déposée à la surface de l'échantillon. Le choix de la matrice reste aussi pour l'analyse de surfaces un critère primordial qui définira la qualité spectrale (intensité des signaux, bruit de fond, rapport signal/bruit, résolution du signal) et permettra d'orienter de manière plus ou moins spécifique les analyses vers une famille de composés en particulier.

L'imagerie MALDI correspond à une suite d'événements programmés et automatisés permettant de réitérer cette opération d'analyse de manière systématique suivant un quadrillage défini. Ainsi l'ensemble de la surface à analyser est couvert et une collection de données est générée, dont le traitement pourra se transcrire par une visualisation sous forme d'image. La résolution des images sera définie, comme pour toute image, par le nombre de pixels la constituant ; d'un point de vue analytique cela préconise que les pixels soient les plus petits possible et juxtaposés si possible. Ceci implique que l'aire irradiée par le faisceau laser soit réduite au maximum tout comme le pas entre deux points d'analyse. Dans le premier cas, la solution la plus immédiate consiste à refocaliser le faisceau laser incident et dans le deuxième cas cela implique que le pas doit être égal au diamètre de la zone irradiée sachant que celle-ci est de forme circulaire à elliptique dépendamment de l'angle d'incidence du faisceau laser. Pour chaque point analysé les données seront stockées avec leurs coordonnées dans une matrice de deux colonnes reportant les différents  $m/z$  et leur intensité correspondante. Après acquisition, ces données seront traitées de manière à extraire pour un pic (soit un  $m/z$  donné en prenant le centroïde de ce pic) l'intensité ou l'aire de ce pic pour toutes les coordonnées effectuées puis d'affecter une couleur en rapport avec cette mesure sur une échelle de couleur choisie par l'utilisateur (e.g. monochromatique ou multi chromatique). Ainsi la couleur de départ sera affectée aux pics d'intensité la plus faible et la dernière couleur sera affectée aux pics d'intensité maximum. Sur ce principe, l'image reconstruite permet de visualiser la répartition d'un composé donné au sein du tissu d'origine.

Ainsi, l'imagerie MALDI se classe dans la famille des imageries dites « moléculaires » et présente le net avantage de conduire, après acquisition, à la reconstruction d'autant d'images que de pics observés dans les spectres, sans nécessiter de connaissance préalable sur les composés recherchés. Une séquence complète d'imagerie MALDI comporte plusieurs étapes de la préparation de l'échantillon à la visualisation finale des images des différents composés. Son principe général est présenté sur la Figure 5.

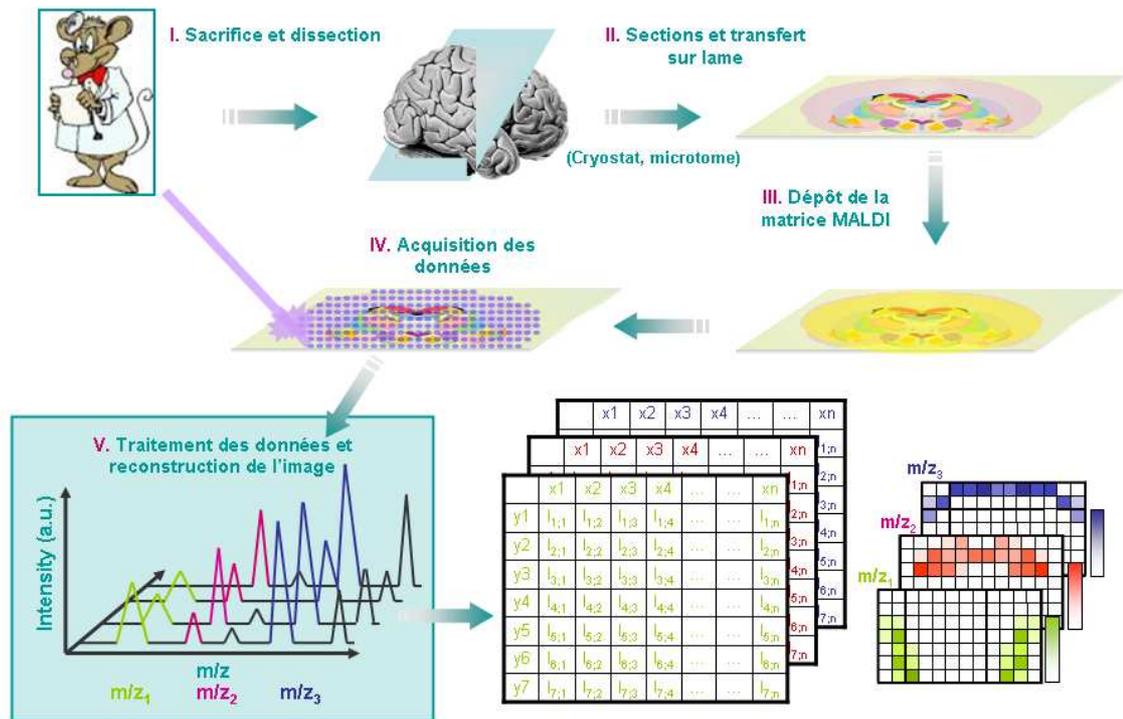


Figure 5: Principe général de l'imagerie par spectrométrie de masse

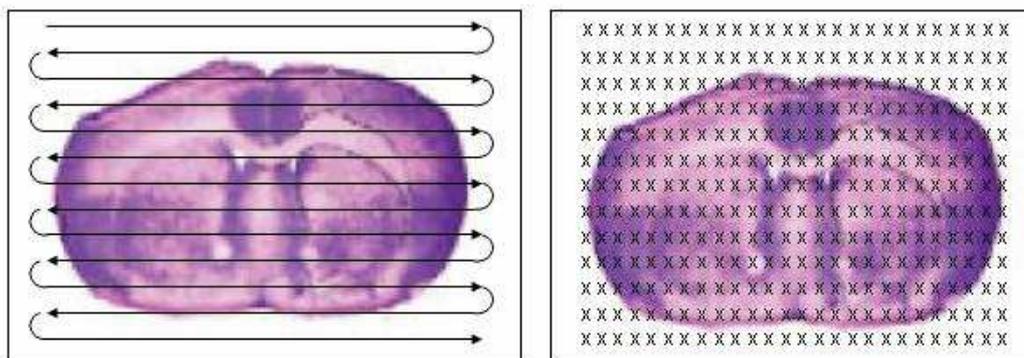
Globalement, une séquence classique d'imagerie comporte les étapes suivantes :

### 1 - Préparation de l'échantillon

- Obtention d'une coupe la plus plane possible à partir de l'échantillon (organe, biopsie, plante, feuille, matériau...)
- Transfert de la coupe d'échantillon sur une lame de verre conductrice
- Dépôt de la matrice MALDI

### 2 - Acquisition des données

- Transfert de l'échantillon dans la source du spectromètre
- Analyse de l'intégralité de l'échantillon en déplaçant le support sous le faisceau laser d'un pas défini par l'utilisateur et en réalisant ainsi un quadrillage régulier recouvrant tout l'échantillon (**Figure 6**).



**Figure 6: Balayage d'une coupe de cerveau de rat par le faisceau laser et position des différents endroits où le laser a frappé la plaque**

- Pour chaque point de l'échantillon un spectre qui sera représentatif de la composition en ce point est enregistré ; il représente une moyenne de spectres correspondant au nombre de tirs laser effectués en ce point et défini par l'utilisateur. Comme le montre clairement la **Figure 7<sup>24</sup>**, pour trois spectres moyens types enregistrés dans des régions différentes d'une coupe de cerveau de rat, des variations importantes d'abondances sont observées pour les différents pics. En particulier, le pic à  $m/z$  5444 est présent au niveau du cortex cérébral et du striatum, mais pas dans le corps calleux.
- Pour chacun de ces spectres, les coordonnées spatiales localisant le point où l'analyse a été effectuée sont sauvegardées.
- Génération d'une collection de données de type matricielle

### 3-Traitement bioinformatique des données

- Mesure pour un  $m/z$  donné de l'intensité ou de l'aire du pic pour toute la collection de spectres avec leur localisation
- Affectation d'un code couleur en rapport avec cette mesure en chaque point donné de localisation
- Reconstruction d'une image dont la couleur évoluera selon l'intensité de mesure. L'échelle de couleur variera avec la concentration du composé dans la région analysée. Ainsi, sur la **Figure 8**, l'absence de la molécule étudiée est traduite par du noir. Sa présence est représentée par l'échelle de couleurs allant du rouge pour une très petite quantité au blanc pour la plus forte densité.

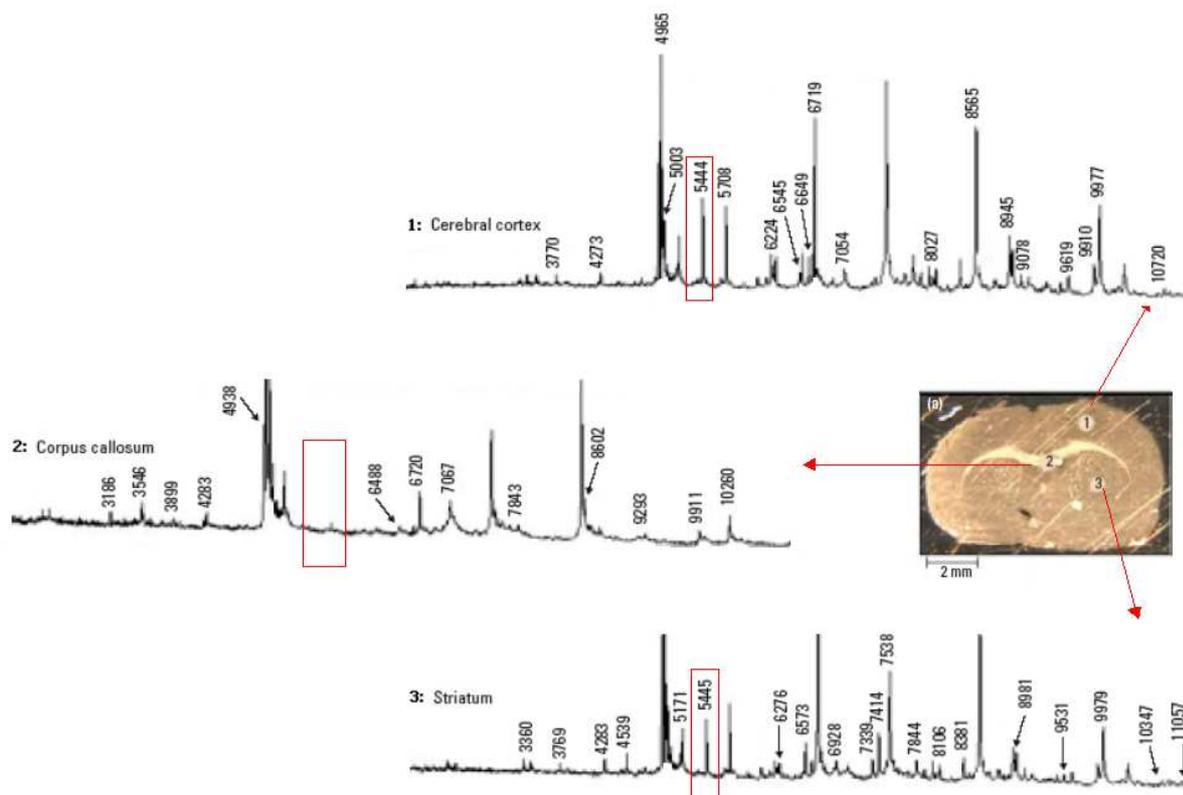


Figure 7: Exemple de spectres obtenus en différents points de la plaque pour la même masse <sup>25</sup>

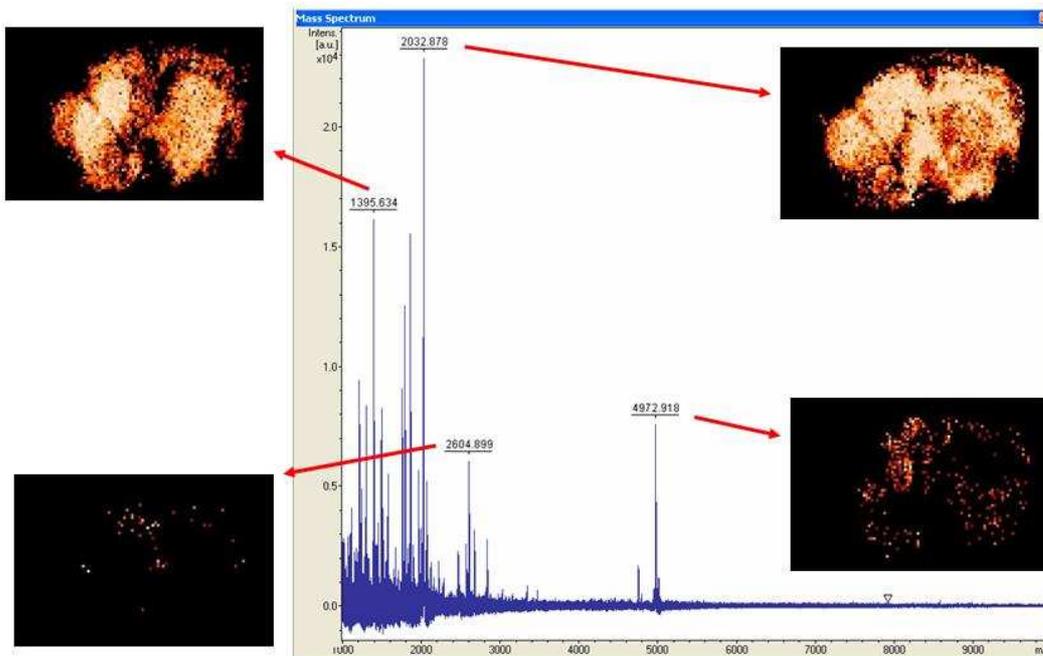


Figure 8: Exemple de plusieurs images obtenues en choisissant différentes masses sur un spectre collecté lors d'une analyse sur une coupe de cerveau de rat

### 2.4.3 Imagerie MALDI et développements

L'imagerie MALDI est une technique à fort potentiel applicatif pour les problématiques biologiques mais qui, comme toute technique, a nécessité et nécessite encore de nombreux développements. Ceux-ci s'articulent principalement autour de trois axes qui sont les suivants : développements instrumentaux et/ou fondamentaux, développements analytiques et de préparation d'échantillon et développements liés au traitement des données. Après une décennie d'existence, l'imagerie MALDI a déjà connu de nombreux développements et la simple comparaison des premières images présentées avec celles obtenues maintenant fournit un constat suffisamment parlant pour se rendre compte de l'évolution de la technologie, en particulier sur le plan du traitement des données et de la reconstruction des images (**Erreur ! Source du renvoi introuvable.**).

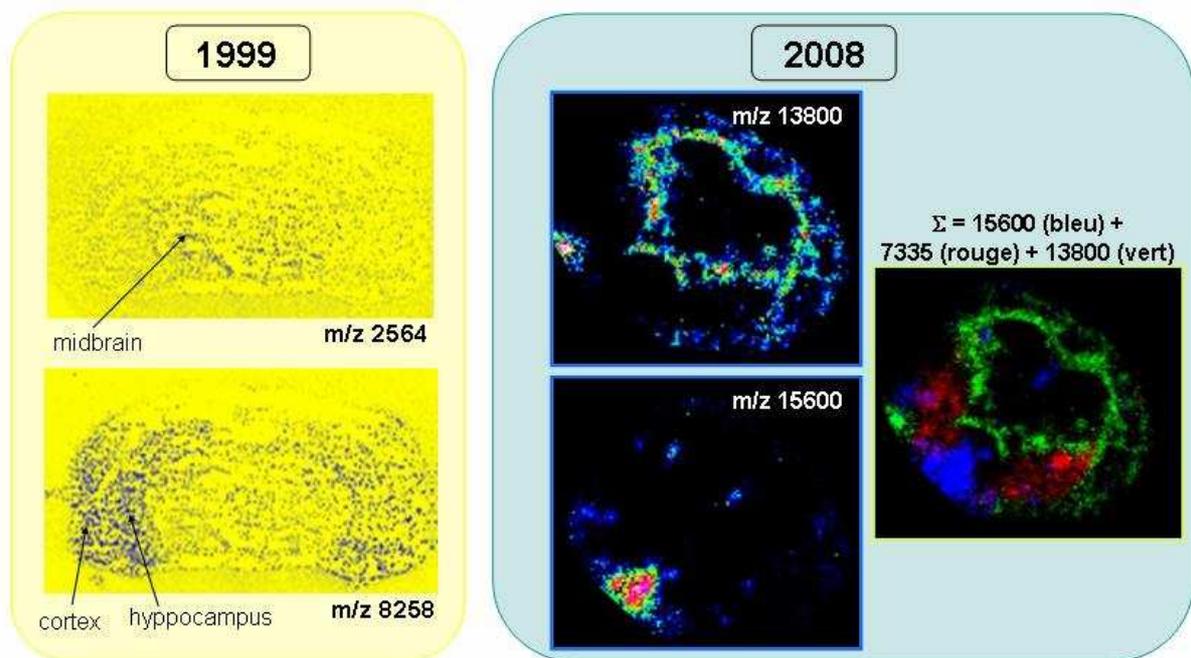


Figure 9 : Comparaison d'images moléculaires MALDI obtenues à partir de coupes de cerveaux de rat en 1999 et cerveaux de souris en 2008 (Equipe MIT, Lille).

#### Préparations d'échantillons et stratégies pour tissus fixés

Les différentes expériences réalisées ont rapidement démontré l'importance de la préparation des échantillons et de la stratégie à choisir dépendamment du mode de conservation de l'échantillon. Il faut distinguer les stratégies employées pour les tissus conservés congelés et celles pour les échantillons fixés, ou fixés et inclus en paraffine.

Les échantillons d'analyse les plus simples pour l'imagerie MALDI sont les échantillons congelés. Ils ne nécessitent pas de traitement forcé de l'échantillon, et la plupart des études parues dans la littérature ont été réalisées sur des échantillons de ce type. Pour ces échantillons, et notamment dans le cadre de l'analyse de peptides/protéines, différents traitements ont été testés avec un relatif succès. Par exemple, P. Chaurand et coll.<sup>27</sup> ont pu

montrer que les traitements des tissus avec des mélanges Eau/EtOH permettaient une augmentation de l'intensité des signaux de protéines par retrait partiel des sels, ainsi que de certains composés hydrophiles. R. Lemaire et coll.<sup>28</sup> ont, quant à eux, mis en évidence que l'utilisation de solvants organiques comme le chloroforme, l'hexane, l'acétone ou le toluène, conduisait à une augmentation conjointe de l'intensité des signaux peptidiques et protéiques, ainsi que du nombre de composés détectés (~25-30%). Les études réalisées démontrent que le traitement des tissus avec ces solvants permet de délipider partiellement les coupes de tissus et facilite l'accès aux peptides/protéines, et que chaque solvant permet l'extraction d'une classe de lipides plus particulière. Des études ont également été menées pour déterminer la compatibilité des analyses de tissus en MALDI avec des colorations histologiques. Il paraît en effet particulièrement important de pouvoir confronter les données histologiques et moléculaires obtenues à partir d'une même coupe de tissu. Les expériences réalisées montrent malheureusement que toutes les colorations ne sont pas compatibles avec l'analyse MALDI<sup>27</sup>. Par exemple, le bleu de méthylène ou le crésyl violet offrent une compatibilité correcte et les pertes de signal sur coupes présentant ces colorations sont minimales. En revanche, des colorations qui sont très utilisées en milieu hospitalier, comme l'Hématoxyline/Eosine et L'Hématoxyline Eosine Safran, conduisent à des spectres difficilement exploitables.

Les échantillons fixés, ou fixés et paraffinés, présentent un autre problème. La plupart des fixations utilisées conduisent à un pontage des molécules et en particulier des protéines au sein des tissus. La fixation la plus couramment utilisée est celle au paraformaldéhyde (PAF). Elle génère la formation de ponts méthylène entre les protéines et notamment par les fonctions amines libres. Elle est difficilement réversible et rend ardu l'accès aux molécules. Cependant, si les fixations sont une difficulté pour les analyses moléculaires, elles offrent en revanche de nombreux avantages. Les échantillons fixés et inclus en paraffine, en particulier, sont très stables dans le temps et peuvent être conservés sur plusieurs décennies (contrairement aux échantillons congelés dont les dégradations sont notables au cours du temps). Elles permettent également une meilleure préservation des structures morphologiques fines du tissu lors de la réalisation des coupes. De plus, les échantillons hospitaliers provenant de patients (biopsies entre autres) sont, pour ces raisons, stockés dans les archives après fixation et inclusion en paraffine (Formalin Fixed and Paraffine Embedded FFPE tissues). Il a donc été d'un enjeu tout particulier pour l'imagerie MALDI d'arriver à travailler à partir de tels échantillons. R. Lemaire et coll.<sup>29</sup> ont ainsi montré que les tissus FFPE pouvaient être exploités pour l'imagerie MALDI selon deux stratégies dépendamment du temps de conservation de l'échantillon. En effet, il a été mis en évidence que les échantillons de moins d'un an pouvaient être exploités par simple utilisation d'une matrice MALDI réactive particulière. Par contre, les tissus conservés sur des périodes plus longues nécessitent le développement de stratégies de digestion enzymatique in situ. Dans ce cas, l'image des protéines doit être obtenue indirectement sur la base des images de répartition de ses différents fragments de digestion. Ces travaux ont permis d'ouvrir l'imagerie MALDI au domaine de la protéomique clinique.

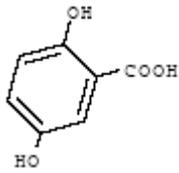
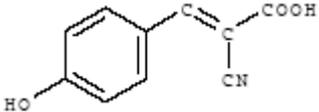
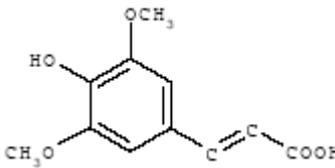
Les stratégies de digestions *in situ* sont également applicables aux tissus congelés<sup>30</sup>. Elles ont l'avantage d'être compatibles avec les différentes colorations histologiques<sup>29</sup> et offrent donc la possibilité de corréler les données moléculaires et histologiques. Elles permettent également de générer des peptides de m/z compatibles avec les techniques de fragmentations en mode MS/MS conduisant l'imagerie MALDI à permettre également l'identification *in situ* des composés. Elles requièrent en revanche l'utilisation de modes de dépôts particuliers pour l'enzyme afin d'assurer la non délocalisation des peptides issus de la digestion.

### **Matrices MALDI et modes de dépôt**

La matrice a un rôle primordial dans les analyses MALDI et il peut être montré sans surprise que tel est également le cas lors de l'analyse de tissus. Elle est la clef de voûte du MALDI. Elle dicte les familles de composés qui seront observés et affecte la qualité spectrale. Le manque de connaissance des processus MALDI n'a pas permis à ce jour d'établir de règle permettant de prédire quelles seraient les composés présentant les propriétés requises pour être des matrices performantes. Il est bien connu que la matrice doit présenter une bande d'absorption à la longueur d'onde du laser utilisé mais, si cela est une condition nécessaire, il ne s'agit pas d'une condition suffisante. Les matrices MALDI classiquement employées proviennent de tests empiriques. Leur utilisation en MALDI pour l'analyse des peptides et protéines révèle que celles-ci présentent des propriétés relativement similaires, mais n'ont pas toutes un comportement tout à fait identique (Tableau 2). L'acide 2,5-dihydroxybenzoïque (2,5-DHB) est difficilement applicable pour l'analyse de tissus de par sa cristallisation particulièrement hétérogène sur les tissus qui laisse de nombreuses zones sans cristaux. L'acide  $\alpha$ -cyano 4-hydroxycinnamique ( $\alpha$ -CHCA) et l'acide sinapinique (SA) présentent, eux, une cristallisation adaptée.  $\alpha$ -CHCA permet l'analyse des peptides à partir des tissus, mais ne présente pas de succès pour les protéines. Et SA se révèle particulièrement performant pour les protéines, mais sans grand succès pour les petites masses, contrairement aux observations en mode MALDI classique. La recherche de nouvelles matrices a donc été d'un enjeu tout particulier pour l'imagerie MALDI. Les matrices ioniques solides se sont révélées être des matrices particulièrement performantes dans ce contexte. Dans leurs travaux, R. Lemaire et coll. ont montré que certaines de ces matrices présentaient des propriétés intéressantes pour l'analyse des peptides/protéines à partir de tissus. En effet, elles conduisent, par rapport aux matrices classiques, à une augmentation du signal et du nombre de composés détectés, à la fois pour le mode positif et négatif. De plus, ces matrices présentent une plus grande reproductibilité d'un tir à l'autre et d'un point à l'autre avec une très grande stabilité sous vide, propriété se révélant intéressante dans le cas d'acquisitions longues.

D'un autre côté, le mode de déposition de la matrice a du être optimisé. En effet, elle doit être déposée tout en assurant que les analytes ne soient pas délocalisés par les solvants utilisés. De plus, la taille des cristaux formés doit être minimisée pour augmenter la résolution

des images. Enfin le dépôt doit être contrôlable et reproductible. Ainsi, le simple dépôt de la matrice à l'aide d'une micropipette ne permettait pas de répondre à ces critères. L'utilisation de systèmes de spray pneumatiques<sup>22</sup> s'est révélée, dans un premier temps, être une solution alternative. Cependant, le manque de contrôle sur le système rendait les expériences peu reproductibles et parfois aléatoires. Il faut aussi garder en mémoire que le MALDI est source de compromis. Il a pu en effet être démontré que, si les dépôts de matrice par Electrospray sont de loin la méthode qui permet l'obtention d'une couche de micro cristaux de matrice la plus fine et la plus reproductible, les quantités de solvants mis en jeu deviennent alors trop faibles pour conduire à une incorporation des analytes dans les cristaux de matrice et les résultats analytiques sont alors de piètre valeur. Ces dernières années ont ainsi vu éclore de nouveaux systèmes de dépôt de la matrice. Principalement deux types de dépôts sont proposés, par microgouttes ou par nébulisateurs contrôlés. Les systèmes microgouttes consistent à déposer la solution de matrice suivant un quadrillage le plus resserré possible, qui sera utilisé pour l'acquisition de l'image. Différents systèmes peuvent être utilisés pour cela comme le dépôt par micro-capillaire<sup>31</sup>, le dépôt par système piézo-électrique<sup>30</sup> ou encore le dépôt par éjection acoustique<sup>32</sup>. Alternativement, pour un coût réduit, une imprimante jet d'encre<sup>33</sup> peut également être utilisée. Le principal désavantage de ces systèmes demeure le temps nécessaire au dépôt (hormis pour l'éjection acoustique) et la taille du dépôt obtenu qui reste d'environ 150 µm. Les nébulisateurs contrôlés sont nettement plus rapides et génèrent des gouttes d'un diamètre moyen d'environ 50 µm, mais offrent moins de souplesse pour certaines utilisations spécifiques.

Abréviation	Nom	Structure	Application
DHB	Acide 2,5-hydroxybenzoïque		Peptides, protéines, lipides, glycoprotéines, oligosaccharides
$\alpha$ -CHCA	Acide $\alpha$ -cyano-4-hydroxycinnamique		Peptides, protéines, lipides, oligosaccharides, PSD.
SA	Acide 3,5-diméthoxy-4-hydroxycinnamique, ou acide sinapinique		Peptides, protéines, glycoprotéines

**Tableau 1 : Matrices les plus couramment utilisées en Spectrométrie de masse MALDI-TOF pour l'analyse de composés biologiques et spécialement des peptides et protéines.**

### **Acquisition des données**

L'acquisition des données a également connu différents progrès au cours de ces dernières années, tant sur la vitesse d'acquisition que sur les logiciels. La durée d'acquisition des données reste un critère important et conditionné par différents paramètres : la fréquence de répétition des tirs laser, la vitesse de déplacement de l'échantillon et le nombre de tirs laser (soit de spectres) moyennés par point d'analyse. L'implantation au sein des instruments MALDI, ces 4 dernières années, de lasers atteignant des fréquences de répétition de tirs de 200 Hz a grandement contribué à diminuer les temps d'analyse. Là où, pour l'acquisition d'un jeu de données sur une section de cerveau de rat de 2x1 cm avec un point tous les 75 µm et 100 tirs lasers moyennés par position, il aurait fallu 325 heures avec un laser de fréquence 3 Hz, désormais moins de 5 heures suffisent pour acquérir une telle image. Le nombre de spectres moyennés par point peut, en étant réduit, conduire à une nette diminution du temps d'acquisition des images. Cependant, plus le nombre de tirs est grand et plus l'analyse sera statistiquement proche de la réalité. Il est donc avantageux de pouvoir au moins moyennner 50 tirs par position.

### **Résolution des images**

L'amélioration de la résolution des images a été bien évidemment l'un des fers de lance dans le développement de l'imagerie MALDI. L'un des plus gros handicaps reconnu à l'heure actuelle à cette technologie reste encore sa résolution spatiale. En effet, même avec une résolution actuelle de 50 µm en routine, l'imagerie MALDI fait pâle figure à côté de l'imagerie par spectrométrie de masse avec source SIMS (Secondary Ion Mass Spectrometry) qui fonctionne en routine avec une résolution à l'échelle du µm. Mais, si les images MALDI sont moins résolutes, le SIMS ne permet pas, quant à lui, d'analyser de composés très polaires et de masse supérieure à 500 u, ce qui le cantonne à l'analyse de petits composés organiques comme les lipides ou les acides aminés. Il faut néanmoins prendre en compte les importantes améliorations connues par le domaine de l'imagerie sur cet aspect avec un passage d'une résolution d'environ 200 µm à 50 µm en quelques années. L'amélioration des systèmes de refocalisation des lasers au sein des instruments est à l'origine de ce progrès. Des systèmes de focalisation des lasers poussés ont été mis au point et étudiés mais ne comblent malheureusement par les espoirs attendus. Ainsi, en 2002, B. Spengler et coll.<sup>34</sup> ont publié un instrument présentant une résolution à l'échelle du µm. Mais, si la focalisation du laser est possible, les résultats ne sont pas probants par manque d'ions produits. En effet, il a pu être démontré<sup>35</sup> qu'en dessous d'un diamètre de 30 à 50 µm du faisceau laser, la quantité d'ions produits diminue drastiquement, et il devient alors impossible de détecter d'autres ions que ceux de matrice. Actuellement cette limite n'a pu être franchie, et nécessite de l'investissement au niveau fondamental des mécanismes MALDI pour être résolue. Il faut noter cependant qu'un instrument de géométrie particulière a été présenté en 2007<sup>36</sup> et atteint une résolution de 30 µm.

### **Traitement des données**

Le traitement des données demeure évidemment une part importante de la séquence mais également une part délicate. Il faut trouver les traitements qui permettent au mieux d'extraire les données des spectres, tout en conservant des données qui décrivent la réalité. Ainsi, les traitements peuvent être nettement bénéfiques ou terriblement néfastes s'ils ne sont pas utilisés à propos. Seulement depuis quelques années des solutions réellement dédiées au domaine de l'imagerie MALDI commencent à se développer. Certains traitements de base des spectres MALDI restent valables et réutilisables pour l'imagerie MALDI, comme les algorithmes de lissages des spectres, de calibration ou encore de déconvolution des pics. D'autres outils ont dû et doivent encore être réellement spécifiquement développés pour cette application, comme ceux liés directement à la reconstruction de l'image. Dans le cadre du traitement des données, plusieurs problèmes ont maintenant été évoqués et notamment celui de réaligement des spectres (variations de  $m/z$  liées aux variations d'épaisseur) sur l'ensemble du jeu de données, sachant qu'un décalage systématique fausserait les mesures d'intensité. Ont également été abordés les problèmes de correction de la ligne de base, de soustraction du bruit de fond, ou encore de normalisation des données<sup>37, 38</sup>. D'autres auteurs se sont également intéressés à l'application d'analyses type PCA dans le cas d'analyses comparatives entre deux séries d'échantillons distinctes ou pour la classification de régions d'intérêts (ROI) à l'intérieur d'un même échantillon<sup>39, 40</sup>. Les problématiques liées à la reconstruction de l'image par elle-même ont été moins abordées dans la littérature, même si, comme nous le verrons plus avant dans la suite de ce mémoire, différents logiciels d'imagerie MALDI sont maintenant proposés sur le marché. Sur le plan de la reconstruction d'images, différentes questions se posent et plusieurs fonctionnalités restent encore à explorer. Par exemple, la plupart des logiciels travaillent sur la mesure de l'intensité des pics. Mais la mesure de l'aire du pic n'est-elle pas une donnée plus réaliste ? Doit-on supposer que les fluctuations du signal doivent conduire à définir des seuils limites minimal et maximal respectivement en deçà duquel le signal est ignoré et au-delà duquel le signal est compté comme maximum ? Peut-on augmenter la qualité de l'image en reconstruisant cette dernière non pas sur un  $m/z$  donné correspondant à une molécule mais sur l'ensemble des ions (des  $m/z$ ) observés et décrivant cette molécule ( $MH^+$ ,  $MNa^+$ ,  $MK^+$ ,  $MH_2^{2+}$ ,  $[M+m]^+$ , ...) ? Si les outils de base pour la reconstruction d'image existent déjà pour d'autres technologies (IRM, PetScan, Confocal) ceux-ci doivent être retravaillés et adaptés au traitement de données issues de la spectrométrie de masse MALDI.

## 2.4.4 Applications de l'imagerie MALDI

L'imagerie MALDI peut toucher tout type d'applications où un composé est recherché au sein d'une surface. Néanmoins, par comparaison à d'autres techniques d'imagerie par spectrométrie de masse et compte tenu de ses caractéristiques, elle trouve tout naturellement ses applications dans le domaine de la biologie. Historiquement les premières applications de l'imagerie MALDI se situent dans le domaine de l'analyse des peptides/protéines sur coupes d'organes d'animaux, mais, à l'heure actuelle, le domaine d'applications de l'IMS s'est considérablement accru tant en terme de familles de composés analysés (lipides, ARNm, acides aminés, neurotransmetteurs, sucres, carbohydrates, saponines,...) que d'échantillons (coupe de tissus d'animaux, biopsies, éléments provenant de plantes, ...). La possibilité de localiser ces composés au sein de tissus en fait bien évidemment un outil de choix pour l'étude de pathologies comme le prouvent les nombreux exemples de la littérature.

Le champ de la clinique protéomique a crû de manière significative ces 10 dernières années et, dans ce domaine, l'imagerie par spectrométrie de masse ouvre les portes vers la protéomique histopathologique. Le but de la clinique protéomique réside dans la caractérisation des mécanismes cellulaires ainsi que la compréhension de l'impact de la maladie et de la thérapie sur les réseaux cellulaires en obtenant des informations sur la détection, le traitement et le suivi de la pathologie. Les avancées technologiques principales de l'imagerie MALDI dans ce domaine consistent en l'identification et la caractérisation directe *in situ* de nouveaux marqueurs à partir de biopsies de tissus frais/fixés et inclus en paraffine (échantillons archivés)<sup>31, 41</sup>. Plusieurs études cliniques et pathologiques du cancer ou de maladies neurodégénératives ont démontré que l'imagerie MALDI est la technologie clef pour la recherche de biomarqueurs, leur localisation et leur validation croisée<sup>3, 25, 42-51</sup>. L'utilisation d'échantillons en blocs de paraffine archivés dans les banques hospitalières des départements de pathologies représentent une mine d'or d'informations<sup>31</sup>.

L'application de l'imagerie MALDI à ces échantillons archivés devrait permettre la création d'une banque internationale de marqueurs de pathologies, et ainsi rendre possible l'élaboration d'un diagnostic plus précoce de ces pathologies et fournir une aide au suivi de la progression de la maladie.

### **Imagerie MALDI pour la recherche et la validation de biomarqueurs du cancer**

A l'heure actuelle, l'un des principaux domaines d'application de l'imagerie MALDI reste le domaine du cancer, avec des études sur sections humaines ou de rongeurs de cancers ovariens, du sein, des poumons, du colon ou encore de gliomes<sup>41, 46, 47, 50, 52-56</sup>. La majorité des expériences sont basées sur les profils protéiques obtenus de sections de tissu afin de distinguer les tissus cancéreux de tissus sains, ou encore entre différents sous-types de cancer. Des algorithmes statistiques peuvent être appliqués aux données MS et faciliter la mise en évidence des divergences inter échantillons<sup>46, 50, 56</sup>. Ainsi, par exemple, dans le cas du cancer du poumon, les auteurs ont pu démontrer une forte corrélation entre les profils MS et la

classification des tissus et ce indépendamment de l'algorithme statistique utilisé. Dans ce cas, les échantillons non tumoraux pouvaient être distingués des échantillons au stade III de la tumeur avec une prédiction de 95% par utilisation de l'analyse discriminante symbolique (ADS) sur une série de 4 versus 4 échantillons et de 89% par utilisation de la méthode de covariance (Weighted Flexible Compound Covariate Method WFCCM) sur un jeu de 38 données<sup>50, 56</sup>. Une étude menée sur le cancer de la prostate à partir de 22 sections de prostate (11 avec et 11 sans cancer de la prostate) par utilisation d'un algorithme permettant de classifier les régions cancéreuses et non cancéreuses a permis de globalement valider à 88% les échantillons avec une sensibilité de 85,21% et une sélectivité de 90,74%. Les résultats obtenus soulignent la capacité de la technique à identifier des marqueurs potentiels plus fiables du cancer de la prostate<sup>54</sup>. Dans le cas du cancer de l'ovaire, une étude mono centrique réalisée sur 25 carcinomes ovariens (stade III et IV) et 23 pathologies bénignes, par analyse MALDI directe, a permis d'identifier 8 biomarqueurs potentiels de ce cancer. Le biomarqueur de plus grande prévalence (80% des cas de cancer) a été complètement caractérisé par combinaison des techniques MALDI-TOF MS et nanoESI-Q/TOF MS et MS/MS. Ce marqueur (m/z 9744) correspond à un fragment C-terminal de 84 acides aminés d'une protéine du complexe protéasome 11s, nommée PA28 alpha ou Reg Alpha. La validation de ce marqueur a pu être obtenue en croisant les études d'imagerie MALDI, d'immunohistochimie (IHC), de western blot et d'imagerie spécifique MALDI à l'aide d'un anticorps dirigé contre la partie C-terminale de la protéine ; il montre une localisation dans les cellules épithéliales de ce fragment avec un adressage différent pour les échantillons bénins (nucléaire) et cancéreux (cytoplasmique)<sup>46</sup>. Pour le cancer du colon, au contraire des autres études, les auteurs se sont intéressés aux lipides et plus particulièrement aux phospholipides comme la phosphatidylcholine (PC) et la sphingomyéline (SM)<sup>55</sup>. L'étude montre l'existence de différences majeures entre les parties saines et cancéreuses du tissu. En effet, la sphingomyéline augmente de manière importante dans les zones cancéreuses, et souligne clairement que les lipides sont, au même titre que les peptides/protéines, des marqueurs de telles pathologies.

Ces exemples démontrent clairement que l'imagerie MALDI est une technique robuste et puissante dans le domaine du cancer pour la recherche et la validation de biomarqueurs cliniques diagnostiques.

### **Imagerie MALDI en neurosciences**

Le second domaine d'application privilégié de l'imagerie MALDI réside dans les neurosciences<sup>57</sup>. Une compréhension complète des fonctions neuronales nécessite la connaissance de l'abondance et de la localisation pour un état physiologique particulier des molécules fonctionnelles importantes, comme les canaux ioniques, les récepteurs aux neurotransmetteurs, les peptides et autres petits composés qui ont un rôle critique dans la signalisation inter et intra cellulaire, soit toute molécule impliquée dans les fonctions neurales. Les techniques de microscopie optique et confocale ont largement été utilisées dans ce

domaine pour répondre à de telles questions. Cependant, l'imagerie MALDI apporte une vision complémentaire à ces technologies par l'accès aux données moléculaires.

Les premières applications de l'imagerie MALDI au système nerveux ont été réalisées sur des modèles invertébrés présentant un système simple et bien adapté à de telles études, comme les lophotrochozoaires et notamment les annélides ou les mollusques. Par exemple, le soma neuronal des mollusques est souvent grand (jusqu'à 1 mm de diamètre chez l'*Aplysie*), les neurones peuvent donc facilement être identifiés et étudiés de manière reproductible dans différents contextes expérimentaux. De plus, les fonctions neurales sont généralement ségréguées dans des ganglions séparés ou régionalisées dans le ganglion, et les tissus neuraux peuvent être facilement et rapidement disséqués et préparés pour l'imagerie<sup>12, 14, 58-60</sup>. Les premières études de tissus nerveux d'invertébrés par MALDI MS furent conduites sur les mollusques et tout d'abord sur le gastéropode *Lymnaea stagnalis*<sup>12, 14</sup> puis, plus tard, sur plusieurs céphalopodes<sup>15, 61, 62</sup>. Plus récemment, l'imagerie par spectrométrie de masse SIMS (Secondary ion Mass Spectrometry) et MALDI furent appliquées à des cryosections de ganglions cérébraux de *Lymnaea stagnalis*<sup>63</sup>. L'imagerie MALDI du neuropeptide APGWamide a permis, quant à elle, de confirmer que ce neuropeptide est localisé de manière prédominante dans le groupe de neurones qui régule le comportement reproductif du mâle chez la Lymnée<sup>64, 65</sup>. Chez la sangsue médicinale, l'imagerie MALDI réalisée chez des adultes et des embryons, montre une forte régionalisation de certains ions. En effet, certains ions apparaissent distribués dans une région latérale incluant des parties contiguës des territoires ventraux et dorsaux et sont absents du système nerveux central ou de ses environs, alors que d'autres ions sont présents dans la partie dorsale de l'animal ou fortement présents dans le système nerveux<sup>66</sup>.

Chez les ecdyzozoaires, la combinaison de l'imagerie MALDI avec la MS<sup>n</sup> a permis d'identifier plusieurs neuropeptides dans les corps cardiaques et allates de *Acheta domesticus* par interrogation de la base de donnée Sweep<sup>67, 68</sup>, les ions fragments diagnostics de basse masse permettant de distinguer les lipides des neuropeptides<sup>69, 70</sup>.

Chez les crustacés, l'imagerie MALDI fut utilisée sur des tissus neuronaux (organe péri cardiaque et cerveau) pour démontrer que deux familles de peptides RF-amide et un peptide fragment de l'orcokinine présentaient des répartitions très différentes des autres membres de leurs familles respectives. L'imagerie MALDI des tissus neuronaux (péricarde et cerveau) a permis par exemple de mettre en évidence la localisation spécifique et distincte de deux peptides RF-amide et de la forme tronquée de l'orcokinine. Plus de 30 neuropeptides précédemment caractérisés ont été identifiés sur la base de leur masse. L'étude en imagerie MALDI au niveau de ces deux organes a permis d'élucider les inter-relations pouvant exister entre de multiples isoformes de peptides de même famille ainsi que leur distribution<sup>71</sup>.

Chez les vertébrés, étant donné les connaissances déjà existantes relatives à la répartition des neuropeptides dans le cerveau de rat ou de souris, ces derniers ont été considérés les modèles en imagerie MALDI<sup>3, 69</sup>. Ces cartes de répartition des neuropeptides en imagerie MALDI corrélaient avec celles obtenues auparavant en immunocytochimie. Plus

d'une centaine de molécules peuvent être détectées lors d'une simple analyse et à côté des peptides et des protéines, la distribution moléculaire de différentes classes de lipides peut également être déterminée<sup>72-75</sup>. De plus, des études différentielles ont également été reportées par des analyses directes de système nerveux de rat injectés ou non de lipopolysaccharides afin de mimer un challenge bactérien<sup>3</sup>. La présence de vasopressine dans le noyau supra optique avant traitement, puis son absence ensuite, confirmait la réalisation de telles études par analyses directes de tissus<sup>3</sup>. A l'heure actuelle, l'imagerie MALDI est utilisée dans le cadre des maladies neurodégénératives<sup>43, 48, 51, 76-79</sup>. Dans ce contexte, elle permet d'obtenir de nouvelles informations en ce qui concerne la recherche et la validation de biomarqueurs de pathologies comme la maladie de Parkinson<sup>48, 76, 77</sup> ou la cartographie de différentes cibles avec une grande précision, ce qui est impossible lors d'études à partir d'extraits<sup>78, 79</sup>. Il est clair que l'imagerie MALDI corrobore et complète les études antérieures obtenues par anticorps et permet la découverte de nouveaux biomarqueurs à partir de tissus fixés et paraffinés<sup>31</sup> ou l'accès à des protéines minoritaires que les techniques classiques de protéomique n'arrivent pas à obtenir.

### **Etudes de la biodistribution des drogues par imagerie MALDI**

Le dernier challenge en imagerie MALDI consiste à devenir une alternative à l'autoradiographie dans le cadre d'études de métabolomiques<sup>80, 81, 82, 83</sup>.

Des études de toxicité et d'impact sur l'expression des protéines ont été conduites à l'aide de cette technologie. Par exemple, l'injection de gentamicine a révélé des dommages irrémédiables au niveau de la région du cortex du rein, une réduction de la synthèse des protéines et une altération du métabolisme des phospholipides<sup>47</sup>. L'approche par imagerie MALDI permet de mettre en évidence les changements dynamiques du protéome qui n'avaient pas été détectés par les autres techniques, montrant que cette technologie possède un champ d'application dans le cadre de l'histopathologie.

L'étude de la distribution de drogue par imagerie MALDI apparaît un enjeu futur pour les compagnies pharmaceutiques. Des analyses de la distribution de drogues au sein du système nerveux, comme la clozapine<sup>84</sup> et la repiridone<sup>85</sup>, ont montré que le traitement de la rispéridone est accompagné d'un phénotype comportemental d'origine extrapyramidale lié à des altérations au sein du striatum des afférences dopaminergiques<sup>86</sup>. Ces altérations ont pu être mises en évidence par imagerie MALDI, tout comme la modification du protéome lié à cette drogue à ce niveau<sup>85</sup>. Les études sur des coupes d'animaux entiers ont montré le réel potentiel de cette technologie pour la biodistribution de drogues et de leurs métabolites<sup>87</sup>, mettant ainsi la technologie de l'autoradiographie au rang des technologies passées.

### **Imagerie MALDI chez les plantes**

L'imagerie MALDI a également été développée chez les plantes dans le cadre de la localisation d'herbicides et de fongicides<sup>88-90</sup>.

Les études de l'herbicide mesotrione (2-(4-mesy-2-nitrobenzoyl)cyclohexane-1,3-dione) et du fongicide azoxystrobin (methyl (E)-2-<sup>89</sup>-3-methoxyacrylate), ont été réalisées à la surface de la feuille de soja par analyses directes à différentes périodes après l'application des pesticides. Des analyses de tiges après addition de 9-aminoacridine et d'acide alpha-cyano-4-hydroxycinnamique (CHCA) dans la solution de nutrition de la plante en croissance hydroponique ont permis de mesurer les métabolites phosphorylés chez la plante <sup>88</sup>.

L'ensemble de ces données montre que l'imagerie MALDI n'est plus restreinte au règne animal mais trouve aussi des applications dans le règne végétal, en vue d'études en environnement, en génétique et en développement.

### **Devenir des applications en Imagerie MALDI**

Les nouvelles directions prises par l'imagerie MALDI s'expliquent par son lien avec l'imagerie tomographique par émission de positon (PET), l'instrumentation de type X-ray CT, et l'imagerie par Résonance Magnétique (IRM) dans le cadre d'études précliniques et cliniques. La complémentarité des techniques non invasives et moléculaires obtenues par imagerie MALDI permettra une meilleure précision pour un meilleur diagnostic. De plus, l'introduction de nouvelles techniques *in vivo* comme la Désorption ElectroSpray Ionization (DESI) <sup>91</sup> offre la possibilité d'analyser la surface des tumeurs. En partant du postulat que le profil protéique des tumeurs varie comparé à celui de cellules saines, les analyses de surfaces devraient permettre de diagnostiquer en temps réel et d'aider le chirurgien dans son acte. En effet, il pourra, en balayant la surface de l'organe, obtenir une image permettant de différencier les zones où les cellules sont saines de celles qui sont cancéreuses mais aussi celles changeant de phénotype. La technique de Jet désorption/ionisation laisse espérer que le diagnostic moléculaire sera rapide et très précis.

Finalement, le diagnostic moléculaire semble sur le point de pouvoir être réalisé par imagerie par spectrométrie de masse à partir de coelioscopie ou de frottis couplé <sup>92, 93</sup>. Et le pronostic sera effectué à l'aide de biopuces sur tissu en utilisant des aptamères, des anticorps, ou des lectines taggés<sup>94</sup>. L'imagerie MALDI deviendra une technologie de routine au niveau clinique et un outil important en complément pour les pathologistes afin d'offrir un diagnostic histo-pathologique moléculaire.

### MALDI-IMS & Logiciels Associés

Historiquement le premier logiciel dédié à l'imagerie MALDI a été le fait de M. Stoeckli lors de ses travaux avec R.M. Caprioli. En effet, les instruments MALDI-TOF commerciaux de l'époque permettaient de fonctionner en mode automatique pour traiter de grandes quantités d'échantillons et permettaient donc, bien que difficilement, l'acquisition des données. Cependant ils n'offraient en aucun cas la possibilité de traiter ces données pour reconstruire des images. C'est ainsi que M. Stoeckli, en 1999, s'est intéressé au développement d'un premier logiciel permettant de se substituer au logiciel d'acquisition du spectromètre de masse pour assurer l'acquisition puis obtenir une reconstruction simple et rudimentaire de l'image. L'instrument utilisé à l'époque étant de la marque Applied Biosystems, les premiers logiciels ont donc été proposés pour ces appareils. Par la suite, M. Stoeckli lui-même a proposé différentes versions de ces logiciels d'acquisition pour les différents modèles de MALDI-TOF et MALDI-TOF/TOF MS de chez Applied Biosystems, ainsi qu'un logiciel de reconstruction d'image, Biomap, beaucoup plus sophistiqué et performant comme nous le verrons dans ce chapitre. Ensuite, les différents constructeurs de spectromètres MALDI-TOF et utilisateurs se sont progressivement intéressés au domaine de l'imagerie et, au cours du temps, plusieurs programmes ont été développés. Ainsi différents logiciels sont accessibles aujourd'hui, soit librement sur internet, soit fournis avec les instruments de spectrométrie de masse.

Les logiciels se repartissent en deux sous-catégories. La partie logicielle concernant l'acquisition des données et la partie logicielle concernant la reconstruction de l'image et le traitement des données spectrales par elles-mêmes. Certains programmes sont uniquement dédiés à l'une de ces deux parties, d'autres plus rares combinent les deux. Globalement les logiciels, qu'il s'agisse de l'acquisition ou du traitement des données, fonctionnent sur le même schéma.

Tous les spectromètres MALDI commerciaux présentent depuis de nombreuses années des possibilités d'acquisition en mode automatique permettant le traitement d'un grand nombre d'échantillons. Ces logiciels fonctionnent sur la base de tables (xls par exemple) qui intègrent l'ensemble des dépôts à analyser sous la forme d'un tableau de coordonnées. De plus, le mode automatique propose généralement un grand nombre d'options de contrôles comme le nombre de tirs laser moyennés par point ou l'ajout de critères permettant des variations des paramètres expérimentaux en cas de difficultés d'analyse. Ces logiciels constituent en fait un point de départ pour l'imagerie MALDI. Ils peuvent d'ailleurs être utilisés tels quels pour directement acquérir les données, mais l'existence de petits logiciels dédiés à l'imagerie permet cependant un gain de temps considérable. Ainsi, le plus simplement, les logiciels d'acquisition dédiés à l'imagerie doivent constituer une interface

efficace ne nécessitant l'implémentation que des paramètres directement liés à ce type d'expérience. L'interface doit donc permettre de :

1 - définir un point d'origine

2 - recréer une table en (x,y) automatique qui définira l'aire à quadriller par calcul en se basant sur le pas de déplacement entre deux points de l'image et le nombre de points à réaliser,

3 - sélectionner une méthode contenant l'ensemble des paramètres instrumentaux pour l'analyse,

4 - conduire à l'enregistrement du jeu de données.

Cette base simple est, en général, commune à tous les logiciels. Elle peut être plus ou moins sophistiquée et ergonomique avec différentes options comme la possibilité de définir la zone à balayer par outils de dessins, ou encore permettre de travailler à partir d'une photographie de la coupe de tissu avant analyse. Le point 4 reste peut-être la partie la plus délicate. Le choix du mode d'enregistrement des données devant dépendre du logiciel servant au traitement des données. Les données peuvent être stockées au format propriétaire classique ou le logiciel peut extraire les données et les enregistrer sous un format propre défini.

La partie de reconstruction d'image est plus délicate dans la mesure où il faut développer des outils de traitement des données adaptés aux données de spectrométrie de masse, et que le recul encore assez faible sur la technologie laisse de nombreuses questions en suspens quant aux traitements les plus appropriés pour garantir la meilleure qualité d'image tout en s'assurant que ceux-ci n'entraînent pas de distorsion de la réalité de l'expérience. Néanmoins la base commune au traitement des données reste :

1 - Extraction des données et/ou exportation pour une relecture dans le langage utilisé.

2 - Présentation d'un spectre moyen servant d'interface à l'utilisateur.

3 - Sélection d'un pic (ion) d'intérêt sur ce spectre moyen.

4 - Extraction de la mesure (intensité ou aire du pic) pour cet ion avec ces coordonnées correspondantes.

5 - Représentation des données sous forme d'une image affectant un code couleur à la mesure sur la base d'une échelle de couleur définie et reportant ce code couleur aux coordonnées correspondantes.

Les nombreuses techniques d'imagerie et notamment *in vivo* permettent de disposer à l'heure actuelle de logiciels de reconstruction d'images performants. Ces logiciels peuvent servir de base aux logiciels d'IMS pour les outils dédiés à l'image par elle-même, tout en développant la partie dédiée aux traitements des données. Comme nous le verrons par la suite les différents logiciels présentent tous des spécificités différentes, la plus grande difficulté résidant dans la relecture des jeux de données et le choix des outils de traitement du signal.

## 3.1 Les outils pour l'IMS dédiés Applied Biosystems

### 3.1.1 MMSIT

Ce logiciel est le premier logiciel pour l'imagerie MALDI. Il a été développé en 1999 par Markus Stoekli <sup>26</sup> et est accessible gratuitement sur le site dédié à l'imagerie par spectrométrie de masse et proposé par M. Stoekli à l'adresse suivante <http://www.maldi-msi.org/>. Il contient toutes les caractéristiques nécessaires pour l'acquisition d'images par spectrométrie de masse sur les appareils de la gamme « Voyager STR » d'Applied Biosystems, dont une base de données d'échantillons et un outil d'évaluation d'images. Ce logiciel agit en commun avec le logiciel d'acquisition de l'instrument « Voyager Control Panel ». Il gère l'acquisition des données ainsi que la reconstruction des images dans la même application (Figure 10).

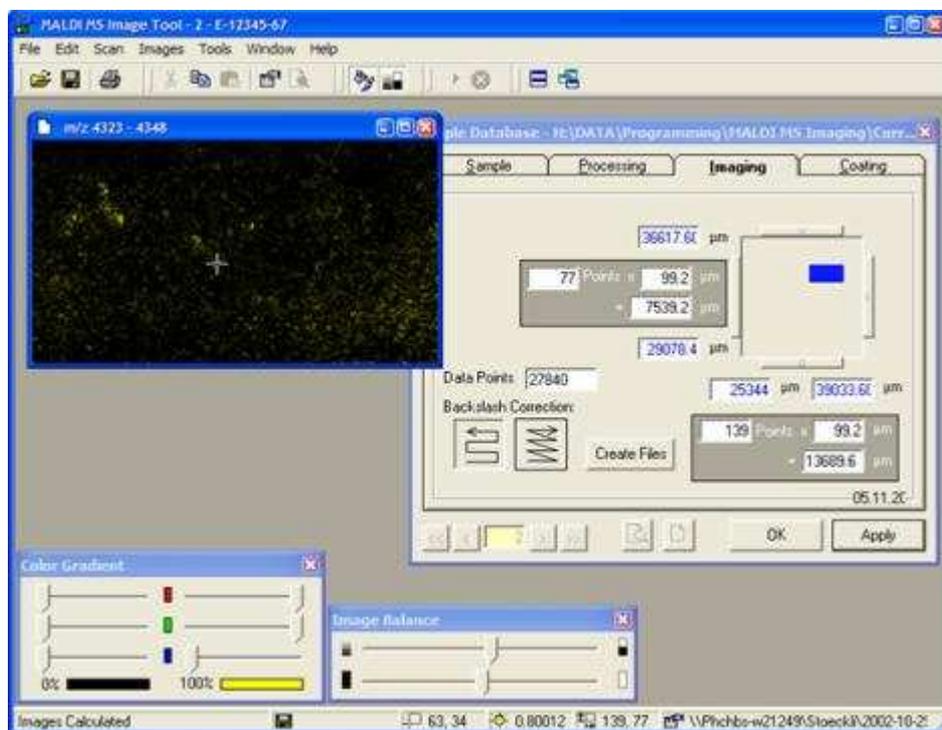
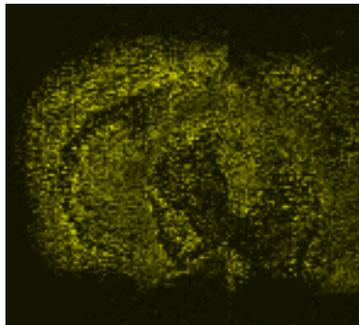


Figure 10: MALDI MS Image Tool (<http://www.maldi-msi.org/>)

Pour la partie d'acquisition sont à définir : le point de départ du balayage, en prenant comme référence les porte-échantillons types utilisés pour l'instrument, ainsi que le nombre de points, et le pas de déplacement entre deux points dans les deux dimensions, et ce afin de pouvoir définir l'aire totale à quadriller pendant l'acquisition. Il est également possible, si désiré, d'importer les coordonnées à balayer directement à partir du mode d'exécution automatique de l'instrument dans « Voyager Control Panel ». Il faut ensuite définir au logiciel une méthode d'acquisition en choisissant parmi les méthodes d'acquisition déjà créées pour l'instrument et qui contiennent les paramètres instrumentaux (tensions d'accélération, gamme

de masse, temps d'échantillonnages des mesures, tension sur le détecteur,...). Avant l'acquisition il est nécessaire à l'utilisateur d'effectuer un spectre de référence sensément représentatif de l'échantillon qui servira au traitement des données.

Une fois l'acquisition réalisée, le logiciel ouvre une interface sur le logiciel de traitement des données de l'instrument « Data Explorer » où s'affiche le spectre de référence. L'utilisateur peut alors avoir accès aux différents traitements habituels des spectres comme la correction de ligne de base ou le retrait du bruit de fond. Une fois les traitements effectués, l'utilisateur sélectionne sur le spectre de référence un pic d'intérêt et le logiciel va alors rechercher dans la collection de spectres l'intensité de ce pic et la renvoie sous la forme d'une image de code couleur monochromatique. Les informations sur le balayage effectué ainsi que sur l'image créée sont enregistrées dans le fichier .dat de données du spectre de référence créé par Voyager Control Panel. Ce premier logiciel assez rudimentaire a néanmoins permis la reconstruction des premières images issues du MALDI. Il gère parfaitement l'acquisition des données mais ne dispose que de peu d'outils pour la reconstruction de l'image.



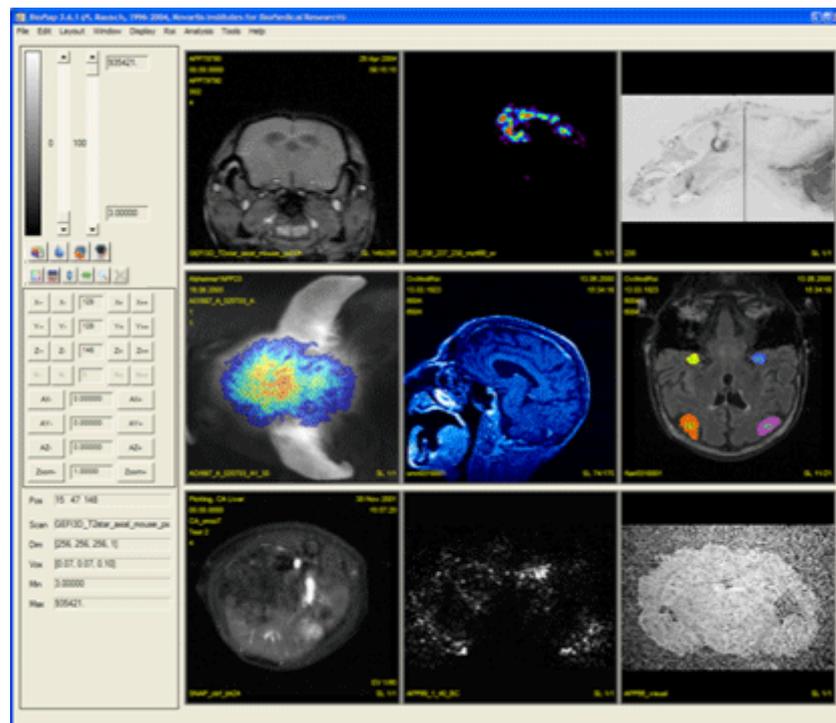
**Figure 11 : Exemple d'image reconstruite avec MMSIT sur une coupe de cerveau de rat réalisée au laboratoire.**

### **3.1.2 Biomap**

Biomap, écrit en IDL par Martin Rausch, puis modifié par Markus Stoekli, est également gratuit et téléchargeable sur le site MADLI-MSI (<http://www.maldi-msi.org/>). Contrairement à MMSIT, il s'agit ici purement d'un logiciel de reconstruction d'images et il ne gère pas l'acquisition des données. Développé à l'origine pour l'évaluation de données d'imagerie de résonance magnétique nucléaire (MRI) en recherche biomédicale, il a été amélioré depuis 1996 pour s'adapter à d'autres techniques d'imagerie, comme l'imagerie optique, l'imagerie nucléaire (tomographie par émission de positon : PET ou Computed Tomography : CT), l'imagerie par fluorescence proche de l'infrarouge (NIRF) et l'imagerie par spectrométrie de masse (MSI). Il fournit une plateforme de visualisation et de stockage qui peut être utilisée pour la visualisation de données de n'importe quelle source, tant qu'il existe un filtre pour importer ce format.

Ainsi, si à l'origine il a été développé en imagerie par spectrométrie de masse pour reconstruire des images à partir de données collectées avec un appareil MALDI TOF Applied Biosystems, les données provenant d'autres instruments comme celles provenant des

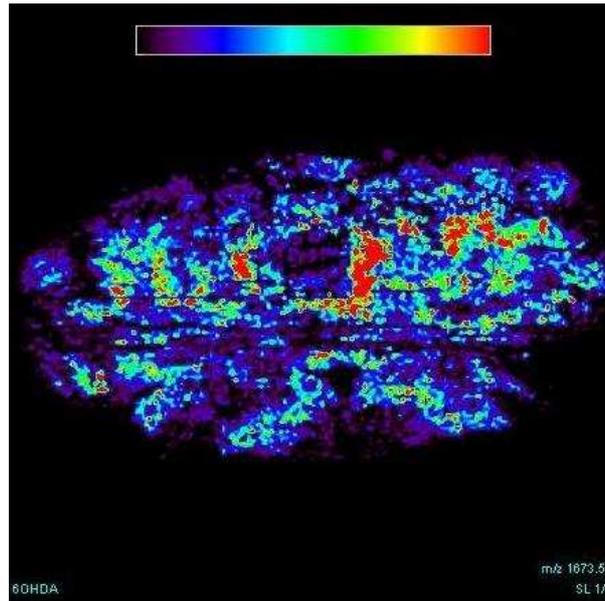
instruments Bruker Daltonics peuvent être importées. Pour la partie spectrométrie de masse il propose de nombreux outils d'analyse adaptés comme, par exemple, la reconstruction de l'image correspondant à un ion donné (pic) d'intérêt sur un spectre, la superposition de l'image moléculaire avec des photos de la coupe avant analyse ou de coupes adjacentes colorées, l'aperçu du spectre correspondant à un pixel de l'image en se déplaçant sur celle-ci et la définition de régions d'intérêts (ROI) pour les études comparatives et statistiques.



**Figure 12: Exemples d'images reconstruites à partir de données provenant de différentes technologies et reconstruites sous Biomap (<http://www.maldi-msi.org/>)**

Cette plateforme peut être étendue par des paquetages logiciels divers, respectivement adaptés pour l'analyse de jeux de données spécifiques. La visualisation est basée sur une reconstruction multiplans permettant l'extraction de tranches arbitraires d'un volume 3D (Figure 12). D'autres caractéristiques liées à la visualisation sont la superposition de deux jeux de données individuels ou la définition de zones d'intérêt. Biomap utilise une représentation en quatre dimensions des données d'imagerie. Les trois premières dimensions sont utilisées pour décrire la position d'un point dans l'espace, la quatrième est réservée pour des paramètres indépendants tels que le temps, la masse, la longueur d'onde ou le poids de diffusion. En plus des données des points, un en-tête est utilisé pour stocker des informations pour l'identification du scan, la position physique de l'objet dans l'espace (important pour l'enregistrement commun de plusieurs scans) ou le protocole d'imagerie. Les données d'image peuvent être importées de différentes sources. Les données calculées par contre seront toujours stockées dans le format Analyze. Ce format a été utilisé car il est largement distribué et peut être chargé par plusieurs autres outils, comme ANALYZE (Mayo clinics) ou

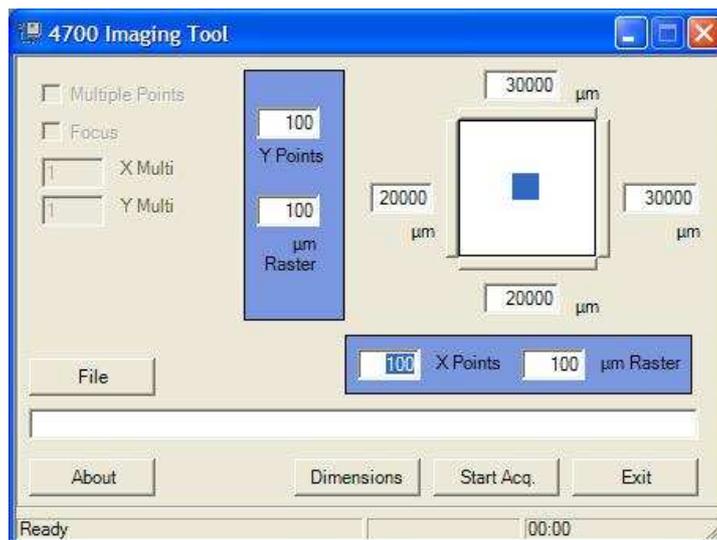
SPM (Wellcome department of cognitive neurology, London). Biomap contient tous les outils requis pour traiter des données d'un sujet particulier, pour combiner des résultats provenant de plusieurs projets ou sessions et pour documenter le résultat final d'une étude. L'import et l'export de données partiellement traitées vers d'autres outils logiciels peut donc être évité. Il faut noter que Biomap n'acceptait qu'un maximum de 32 768 points par spectre, ce qui restreint relativement les jeux de données utilisables. Cependant cette limitation semble avoir été corrigée dans les dernières versions.



**Figure 13 : Exemple d'image reconstruite avec BioMap à partir d'une coupe de cerveau de rat réalisée au laboratoire.**

### **3.1.3 4000 Series Imaging**

Ce logiciel est un logiciel uniquement d'acquisition des images pour les séries d'instruments MALDI-TOF/TOF de la société Applied Biosystems. La société Applied Biosystems est longtemps restée en marge de l'imagerie MALDI faisant le choix de ne pas s'investir dans le domaine et ne propose donc pas de solution dédiée pour ces utilisateurs. C'est ainsi que M. Stoeckli a créé une interface permettant de piloter également ces séries d'instruments. Il supporte le chargement multi-plaques des MALDI-TOF/TOF série 4700 qui permet une acquisition d'un ensemble de plus de 24 images. Il a été spécifiquement écrit pour l'acquisition d'images sur les MALDI-TOF/TOF série 4000. Il s'intègre donc sans problème avec le logiciel 4000 Series Explorer et permet l'acquisition d'images en mode MS/MS. Les données sont stockées dans un format à part et n'entrent pas dans la base de données de l'instrument.



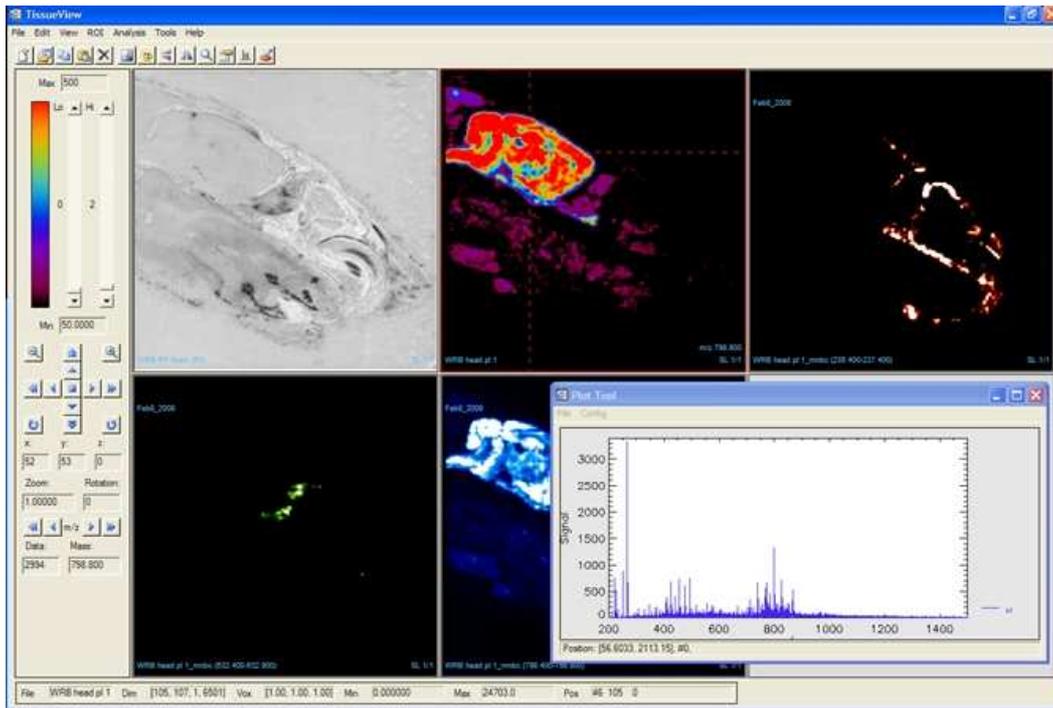
**Figure 14: Interface du logiciel 4700 Imaging**

Le fonctionnement de ce logiciel est proche de celui de MMSIT pour sa partie acquisition. Il faut définir au logiciel des coordonnées de départ ainsi que le nombre de points et le pas de déplacement dans les deux dimensions afin de déterminer l'aire d'acquisition et son quadrillage (Figure 14). Il est également possible à l'utilisateur de positionner la plaque à l'aide de la caméra CCD puis de réimporter les coordonnées correspondantes. Il faut ensuite charger un fichier contenant les paramètres d'acquisition de l'instrument. Sur le même mode que MMSIT, le logiciel requiert avant acquisition la réalisation en manuel d'un spectre de référence de l'échantillon. Pendant l'acquisition le logiciel met à jour continuellement la position actuelle et le temps d'acquisition restant. Il faut noter que ce logiciel présente également une limitation à 32 767 points par spectre ce qui limite les données et impose soit de réduire le nombre de mesures (moins bonne description du spectre), soit de travailler sur une gamme de masse plus restreinte.

### 3.1.4 TissueView

TissueView est le dernier né des logiciels d'imagerie consacrés aux appareils Applied Biosystems. Il gère l'acquisition des données et la reconstruction des images. Il incorpore beaucoup de fonctionnalités primordiales comme l'enregistrement commun d'images, le calcul d'aires, la juxtaposition et superposition de multiples images, et des vues en trois dimensions. Il permet aussi d'afficher les données sous forme d'histogramme, de les stocker dans une base de données et dispose d'outils statistiques pour travailler sur les données.

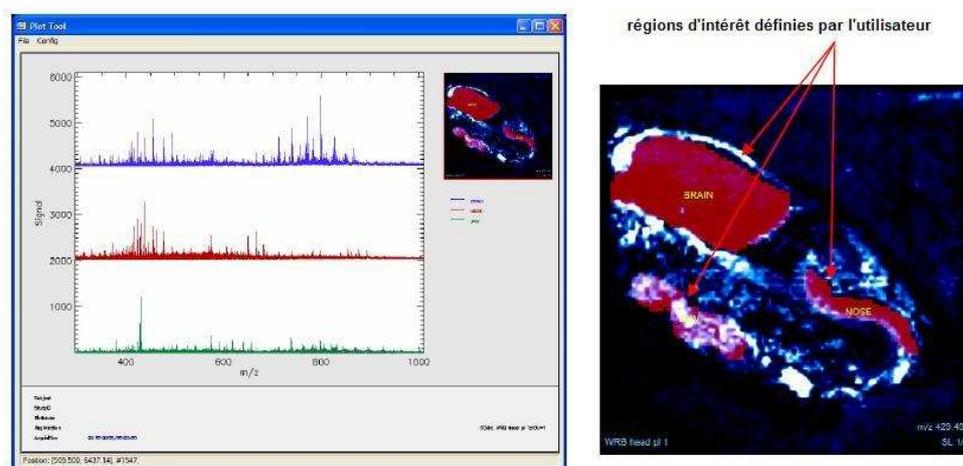
Le logiciel TissueView est compatible avec les spectromètres Applied Biosystems MALDI TOF/TOF 4700 et 4800, et QStar.



**Figure 15 : Interface du logiciel TissueView montrant des images obtenues à partir d'une coupe sagittale de tête de rat, et la fenêtre de visualisation des spectres.**  
 (<https://products.appliedbiosystems.com/ab/en/US/adirect/ab?cmd=catNavigate2&catID=605174>)

Quand le curseur est placé sur la zone de l'image, TissueView permet d'afficher le spectre correspondant au pixel situé sous le curseur ; et ce spectre se met à jour automatiquement lorsque la souris est déplacée. Et inversement, lorsque le curseur se déplace sur un spectre, il affiche l'image pour le pic situé sous le curseur.

TissueView dispose aussi d'une fonctionnalité gérant les régions d'intérêt. Ainsi il est possible de définir sur l'image une région pour laquelle le logiciel va calculer le spectre moyen qu'il sera ensuite possible de comparer avec celui d'une autre région ou d'un autre échantillon.

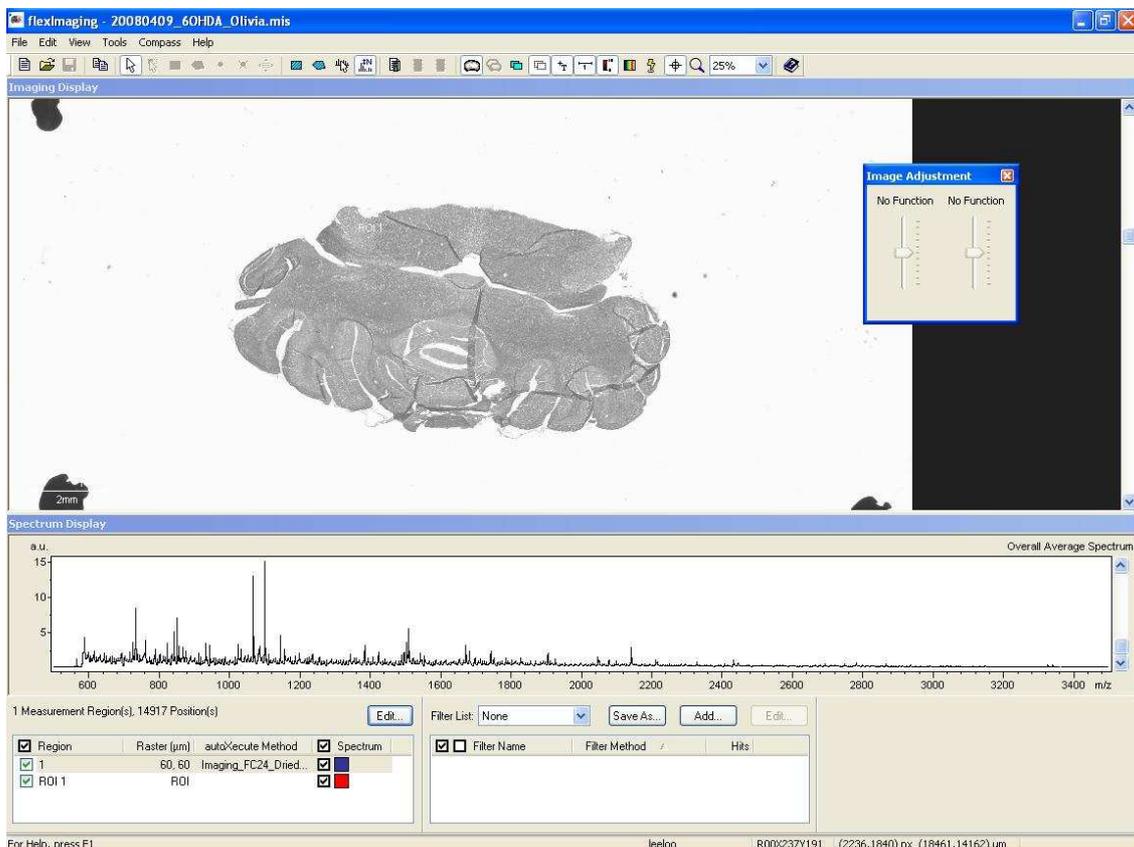


**Figure 16 : Visualisation de régions d'intérêt sur une coupe sagittale de cerveau de rat avec TissueView**

## 3.2 Les outils pour l'IMS dédiés Bruker Daltonics

### 3.2.1 FlexImaging

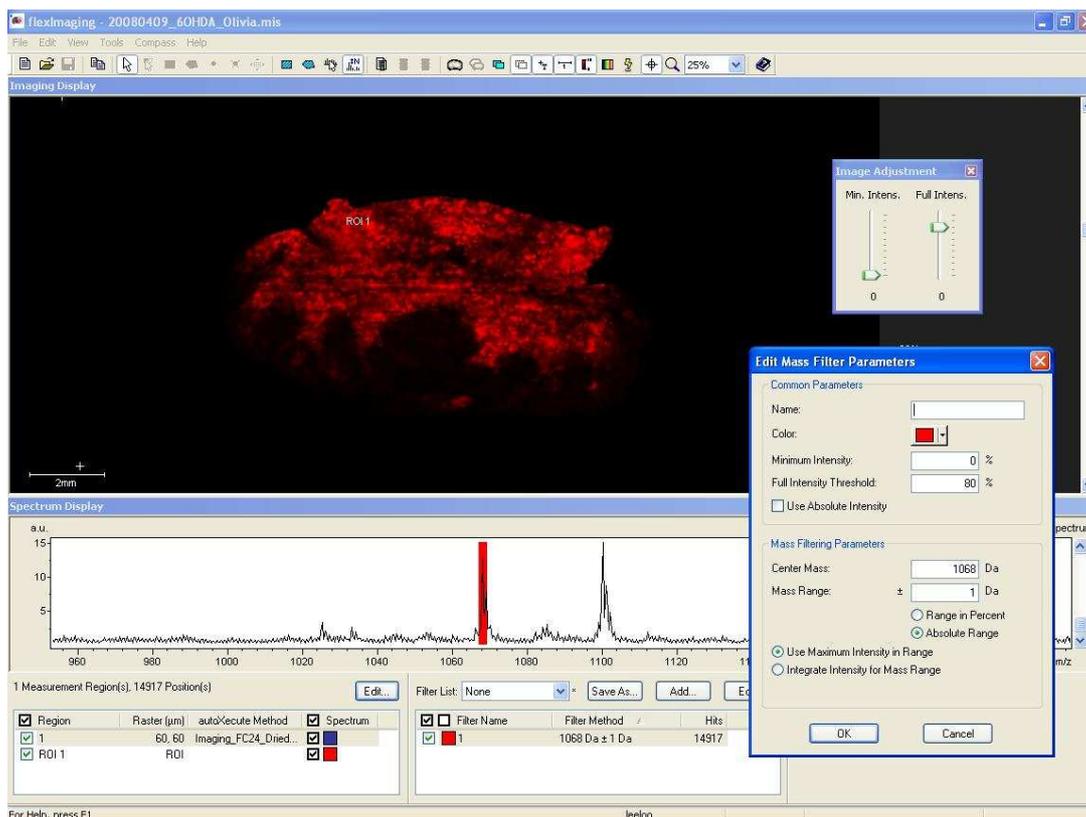
La société Bruker Daltonics a choisi de se lancer dans le domaine de l'imagerie MALDI depuis quelques années et a développé un logiciel propre à ses instruments. Le logiciel proposé, « FlexImaging », gère l'acquisition et la reconstruction des images (Figure 17). Il fonctionne en supplément des logiciels traditionnels d'acquisition et de traitement des spectres proposés sur les instruments.



**Figure 17: Interface de FlexImaging présentant une image histologique d'une coupe de cerveau de rat réalisée au laboratoire.**

FlexImaging est entièrement intégré avec les logiciels classiques et s'appuie sur ceux-ci tout en présentant une interface dédiée. Il a beaucoup évolué ces trois dernières années. Pour l'acquisition, il propose tout d'abord de réaligner l'image prise par la caméra CCD de l'instrument avec une photographie de la coupe avant expérience. Ainsi, il est possible de travailler par la suite directement à partir de la morphologie de l'animal tel qu'observé sur la photographie. Cette étape nécessite un apprentissage sur trois points pour effectuer le réalignement : il faut donc trouver trois points facilement repérables sur le tissu ou ajouter

manuellement sur la lame trois points faciles à identifier. Ensuite le logiciel nécessite de définir la méthode automatique à utiliser, les paramètres instrumentaux pour l'expérience ainsi que la zone à quadriller et le pas de quadrillage. Pendant l'acquisition, le logiciel permet de visualiser l'avancée de l'expérience en retraçant progressivement le quadrillage sur la photographie du tissu. L'avantage du logiciel FlexImaging sur l'acquisition réside dans le stockage des données. Ainsi, si l'expérience s'arrête suite à un problème quelconque ou si l'image est arrêtée par l'utilisateur, les données obtenues jusqu'au point d'arrêt sont disponibles et visualisables, ce qui n'est pas le cas par exemple avec MMSIT où toutes les données sont perdues si la séquence ne peut atteindre son terme. Le logiciel offre également au moment de l'acquisition le choix à l'utilisateur d'enregistrer les données avec un post-traitement automatique des spectres comme le retrait de bruit de fond ou la correction de ligne de base. Lors de la reconstruction, le logiciel propose de visualiser un spectre moyen où classiquement l'utilisateur viendra choisir le pic d'intérêt pour la reconstruction de l'image (Figure 18). Le logiciel permet de définir la largeur du pic à considérer pour la mesure du maximum ainsi que des valeurs de seuils. Ces seuils sont modulables et permettent de définir l'intensité minimum à prendre en compte pour considérer l'existence d'un pic réel (soit distinguer les pics réels du bruit de fond). L'ajustement de ce seuil permet la reconstruction d'images moins bruiteuses sachant que dépendamment des expériences et des tissus, le bruit de fond spectral varie. Il permet également de fixer un seuil d'intensité maximum, c'est-à-dire que tous les pics d'intensité supérieure à ce seuil sont considérés comme au maximum d'intensité. Ceci permet en fait de tenir compte de variations de signaux inter points d'analyse. Le logiciel permet également de superposer l'image ionique reconstruite avec la photographie de la coupe d'origine permettant de visualiser l'image moléculaire en corrélation avec la morphologie du tissu. Le logiciel permet également d'accéder aux fonctions habituelles de traitement de spectres ainsi qu'à une normalisation du jeu de données permettant de tenir compte des fluctuations inter-analyses. Malheureusement l'algorithme de normalisation utilisé par ce logiciel n'est pas donné et n'a pu être obtenu. Enfin, le logiciel propose des fonctions comme la réalisation de régions d'intérêts (ROI) permettant de comparer plusieurs zones du tissu et d'appliquer par la suite un logiciel d'analyses statistiques à ces données particulières ; le logiciel en lien avec FlexImaging utilise une analyse de type Analyse en Composante Principale (PCA). Enfin le logiciel offre différentes fonctions de présentations de données habituellement trouvées dans ce type de logiciel comme la superposition de 2 ou plus d'images de composés différents afin de mettre en évidence la colocalisation de molécules ou la possibilité de travailler sur différentes échelles de couleur.



**Figure 18: Exemple de visualisation de la localisation du peptide de m/z 1068 sur une coupe de cerveau de rat réalisée au laboratoire.**

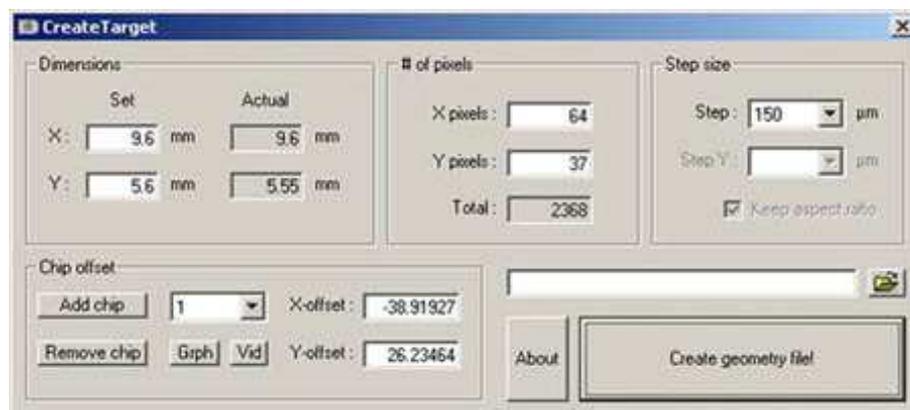
### 3.2.2 Create Target et Analyze This!

Les logiciels appelés CreateTarget et AnalyzeThis !<sup>95</sup> créés par Stefan Clerens sont eux aussi disponibles sur le site [maldi-msi.org](http://maldi-msi.org) ou sur simple demande par mail ([lut.arkens@bio.kuleuven.be](mailto:lut.arkens@bio.kuleuven.be)). La première partie du logiciel (CreateTarget) est dédiée à l'acquisition des images pour les séries d'instruments Reflex ou Ultraflex de Bruker Daltonics. La seconde partie du logiciel « AnalyzeThis ! » est dédiée à la conversion des données de manière à pouvoir reconstruire des images sous Biomap. Les spectres obtenus au format Analyze 7.5 sont convertis en un format compatible avec Biomap.

#### 3.2.2.1 Create Target

Create Target crée une grille de haute densité aux dimensions de la coupe de tissu (Figure 19), et convertit ces paramètres en un fichier de plaque cible virtuelle (fichier géométrique) qui doit ensuite être importé dans le logiciel d'acquisition habituel de l'instrument (FlexControl). La grille peut alors être traitée par le logiciel d'acquisition comme une plaque d'analyse classique. Puis en utilisant le module AutoXecute, il est facile d'acquérir automatiquement des spectres pour toutes les positions d'échantillon. Un redémarrage est nécessaire pour mettre à jour le logiciel à chaque création d'une nouvelle plaque virtuelle. Un fichier peut éventuellement être créé dans le répertoire contenant les

fichiers géométriques standards sur l'ordinateur où se trouve FlexControl afin de contourner ce problème. Le logiciel nécessite toujours un redémarrage la première fois, mais pour les fois suivantes, il suffit de remplacer le fichier existant par le nouveau créé par Create Target.



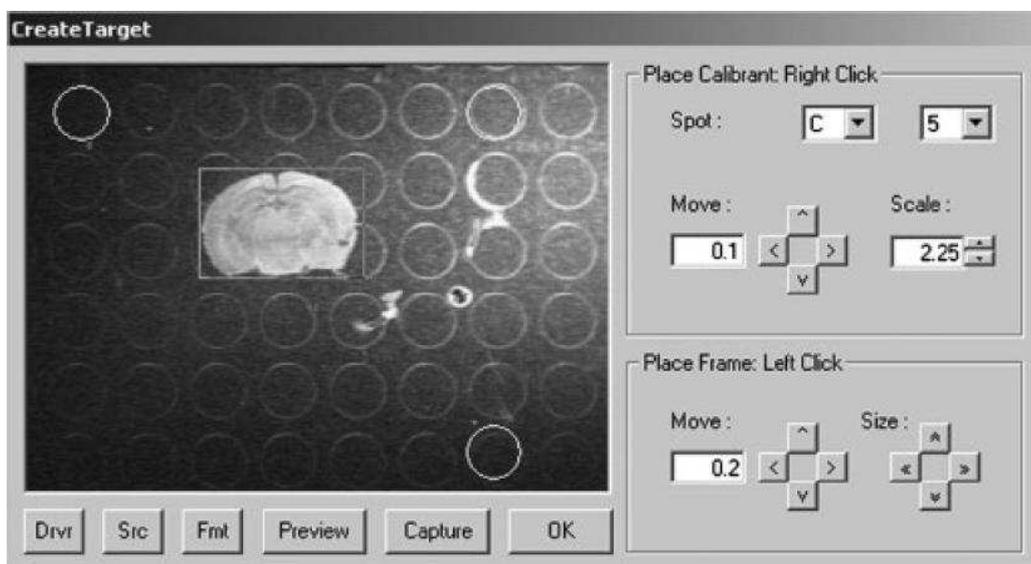
**Figure 19: Création de la grille avec CreateTarget<sup>95</sup>. Dans cet exemple, une coupe de tissu de dimensions 9.6 x 5.6 mm a été utilisée. Avec une distance de 150µm entre deux pixels, le programme calcule la valeur la plus proche comme étant 64 x 37 pixels, correspondant à 9.6 x 5.55mm. Le décalage (chip offset, i.e. la position de la coupe sur la plaque cible) est déterminée par mesure, interface graphique ou interface vidéo, cf Figure 16)**

La création de la grille de points conformément aux dimensions de la coupe de tissu requiert une bonne compréhension du logiciel contrôlant le MALDI-TOF et qui s'appelle FlexControl. Celui-ci permet d'utiliser différentes plaques cibles dans l'instrument, avec différentes quantités et emplacements des spots d'échantillon, comme des plaques de format 96-, 384- et 156-puits, des plaques cibles Applied Biosystems, etc. Les propriétés de ces modèles sont décrites dans les fichiers géométriques, qui contiennent les dimensions de la plaque, le diamètre des spots, les coordonnées exactes de tous les spots d'échantillon, relatives au centre de la plaque, et d'autres paramètres. Etant donné que FlexControl permet l'import de nouveaux fichiers géométriques, il est possible de définir et d'importer un motif très dense de spots d'échantillon, qui constitue finalement une grille d'imagerie. FlexControl traitera une telle grille comme une plaque cible MALDI virtuelle. Donc, quand cette plaque virtuelle a les dimensions d'une coupe de tissu, le logiciel peut acquérir des données d'imagerie par spectrométrie de masse de la coupe en acquérant les spectres de chaque spot d'échantillon de la grille. Sachant cela, la création de la cible virtuelle par le logiciel Create Target semble logique. Il écrit les dimensions de la coupe (autrement dit la plaque cible virtuelle), sa position sur la plaque actuelle, et la densité du motif de la grille dans le format d'un fichier géométrique qui pourra être importé dans FlexControl.

Create Target consiste en une interface utilisateur pour entrer les paramètres définissant la grille de points, et un module pour créer le fichier géométrique correspondant. L'interface présente à l'utilisateur une première boîte de dialogue où les dimensions du rectangle encadrant la coupe de tissu peuvent être entrées (Figure 19). Une taille de pas (c'est-à-dire la distance entre deux points) doit alors être sélectionnée. Le pas est utilisé pour calculer le

nombre de pixels s'adaptant effectivement au rectangle défini. Le nombre de pixels et le pas sont alors utilisés pour calculer les dimensions réelles qui vont être couvertes par cette configuration, et ces valeurs sont affichées pour l'utilisateur afin qu'il les vérifie. Le pas et/ou le nombre de pixels peuvent être ajustés, avec un changement de chaque valeur entraînant une mise à jour de l'autre valeur pour concorder avec les dimensions désirées aussi précisément que possible.

La position de la plaque cible doit être définie ensuite. Cela peut être déterminé par la mesure de la distance du centre de la plaque. Cependant, une vue graphique d'une plaque cible standard de 384 spots est disponible. Un rectangle aux dimensions définies à la première étape est lié au curseur de la souris, permettant un positionnement simple de ce rectangle. La comparaison sur l'écran du motif de la plaque 384-spots avec la plaque cible réelle rend plus facile la correction de la position à l'écran.



**Figure 20: Visualisation d'une coupe de cerveau de souris adulte sur la plaque avec l'interface vidéo de Create Target<sup>95</sup>. Dans cette fenêtre, la position exacte de la coupe sur la plaque cible est définie. Trois cercles sont utilisés pour calibrer la position et l'échelle de l'image, après quoi les dimensions de la coupe sont définies par le traçage d'un rectangle.**

Cependant, la méthode la plus intuitive pour définir la position de la coupe sur plaque cible est fournie par un aperçu vidéo en direct de la plaque cible (Figure 20). Cette option qui utilise un contrôle vidéo ActiveX requiert une caméra vidéo connectée à l'ordinateur (par USB ou une carte d'acquisition vidéo), et placée perpendiculairement à la plaque cible. Les sources d'entrée vidéo, drivers et format vidéo corrects peuvent être choisis en utilisant les trois boutons de configuration. Un aperçu en direct permet à l'utilisateur de positionner et ajuster la zone contenant la section. L'image peut alors être capturée pour définir les paramètres de la coupe. La première étape est de corriger le grossissement de l'image dû aux différents paramètres de zoom ou à une distance différente entre la plaque cible et la lentille.

Cela est réalisé par la mise en correspondance d'un jeu de trois cercles de calibration sur trois cercles de spots d'échantillon sur la plaque cible. Le spot d'échantillon auquel le cercle en haut à gauche correspond doit être spécifié de façon à ce que le logiciel puisse identifier l'emplacement sur la plaque cible d'où l'image provient. La position des trois cercles peut être changée par un cliquer-glisser avec le bouton droit de la souris, ou bien par incrémentations en cliquant sur les boutons disponibles dans l'interface. Le diamètre des cercles et leurs distances relatives peuvent être modifiés uniquement en changeant un facteur d'échelle, afin qu'une parfaite concordance avec la plaque cible combinée avec les autres paramètres soit suffisante pour calibrer l'image vidéo. Un rectangle entourant la coupe de tissu peut ensuite être dessiné en faisant un cliquer-glisser avec le bouton droit de la souris. La taille et la position du rectangle pourront être réajustées plus tard par incrémentations successives à l'aide des boutons dans l'interface utilisateur.

Ces étapes permettent d'avoir une méthode visuelle de détermination de l'emplacement et des dimensions de la coupe. Les deux méthodes supportent actuellement une plaque cible 384-spots pour déposer le tissu. Pour le moment le logiciel n'est pas adapté pour d'autres formats de plaque.

Etant donné que l'exactitude du fichier géométrique repose sur cet enregistrement, une légère erreur, éventuellement introduite lors de la création du fichier, peut causer un recouvrement imparfait entre la grille dans le fichier géométrique et la coupe réelle. Cela oblige à ajouter une marge d'erreur quand on dessine le rectangle autour de la coupe.

Dans le cas où les paramètres sont entrés manuellement, ou en utilisant la vue graphique, il est aussi possible de définir de multiples instances de la coupe à différents emplacements de la plaque cible. Ces occurrences, appelées chips, doivent toutes avoir les mêmes dimensions à cause de la méthode avec laquelle elles sont implémentées dans FlexControl. De multiples chips ne sont pas supportées dans le cas de l'utilisation de la fonction vidéo.

Une fois renseigné, le logiciel crée le fichier géométrique. La première partie de ce fichier consiste en un en-tête contenant des informations comme le nombre de spots, le nombre de lignes et de colonnes, le nombre de chips et leur distance par rapport au centre de la plaque, etc. La deuxième partie du fichier contient la table de mappage, qui liste les coordonnées de chacun des points de la grille. Les coordonnées sont référencées par rapport au centre du chip et sont précises jusque 18 chiffres après la virgule. Quand ce fichier .geo est copié dans le répertoire des fichiers géométriques sur l'ordinateur du spectromètre de masse MALDI-TOF, et que FlexControl est redémarré, la nouvelle configuration de plaque correspondant au fichier .geo tout juste créé devient disponible. Pour contourner la nécessité de redémarrer FlexControl, il est possible de créer un fichier type nommé par exemple IMG.geo dans le répertoire des fichiers géométriques et de l'écraser lors de la description d'une nouvelle cible virtuelle en donnant le même nom au nouveau fichier géométrique créé.

Quand le fichier géométrique en question est chargé dans FlexControl, la grille de haute densité est proposée. Postérieurement, le module AutoXecute de FlexControl peut être configuré pour acquérir automatiquement les spectres de tous les points du fichier géométrique à la suite. Puisque les spots dans le fichier géométrique sont nommés en fonction de leurs coordonnées de pixel XY, les spectres sont automatiquement enregistrés avec les coordonnées XY correctes dans le répertoire de ce nom.

### 3.2.2.2 Analyze This !

Ce programme va lire les fichiers de spectres acquis lors de l'analyse et les convertir dans le format Analyze 7.5 qui peut être importé dans Biomap pour l'analyse d'images. Quand le répertoire contenant le tableau de spectres est sélectionné, le programme affiche les propriétés du jeu de données : nombre de pixels, nombre de points de données, début et fin de la gamme de masse, taille des fichiers, type d'appareil utilisé, nom des répertoires et fichiers concernés ainsi que des informations sur l'avancement du traitement des données (Figure 21).

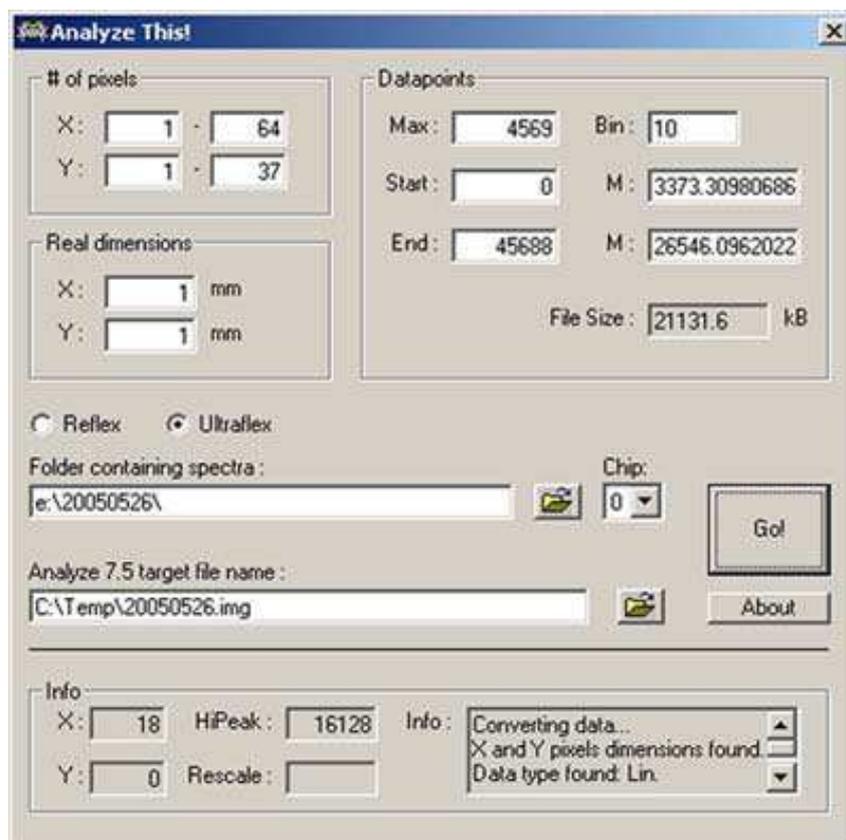


Figure 21: Vue de l'interface Analyze This! présentée sur un exemple où des données avec 45 688 points par spectre sont converties en données avec 4 569 points par spectre. <sup>95</sup>

Comme nous l'avons vu précédemment, le logiciel BioMap utilise le format Analyze 7.5 pour le traitement d'images. Or une analyse avec un spectromètre de masse Bruker Daltonics ne fournit pas de fichier de ce type. Afin de pouvoir recréer des images à partir de données d'un spectromètre Bruker Daltonics avec le logiciel BioMap, l'équipe de Stefan

Clerens a développé le logiciel Analyze This ! pour s'affranchir du logiciel FlexImaging. Analyze This ! lit tous les spectres de données de format binaire acquis lors d'une expérimentation d'imagerie par spectrométrie de masse avec AutoXecute et stocke les données spectrales dans un fichier Analyze 7.5.

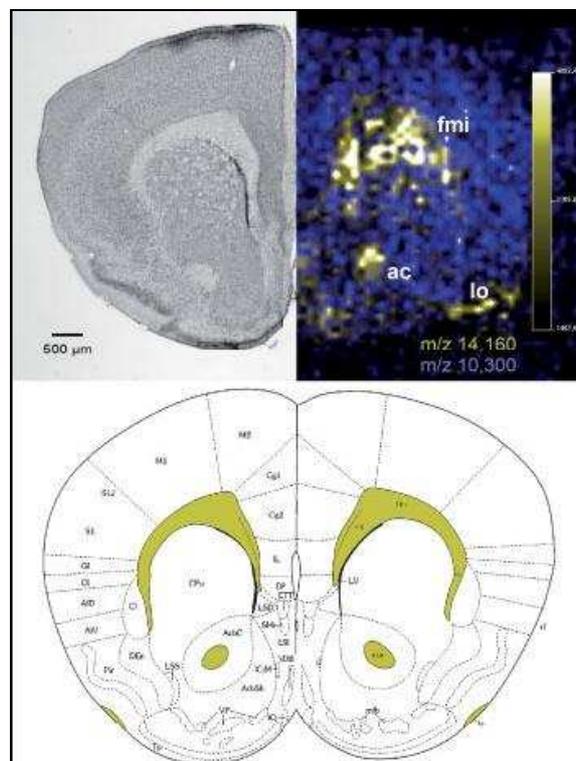
Le programme consiste en une interface utilisateur, un tableau de rapport de progression et un module exécutant la conversion réelle (Figure 21). Avant que le répertoire contenant les spectres ne soit renseigné, l'utilisateur doit indiquer si les spectres ont été acquis sur un instrument Reflex ou Ultraflex à cause de la structure différente dans les répertoires. Une fois le répertoire contenant les spectres choisi, le programme affiche le nombre de pixels dans les dimensions X et Y, le nombre de points dans chaque spectre, et les valeurs de début et de fin dans les spectres. Comme Biomap n'accepte qu'un maximum de 32 768 points par spectre, il a fallu inclure une routine pour réduire le nombre de points par spectre. Des groupes de points (bins) sont automatiquement combinés en un seul point afin d'aboutir à un nombre de points inférieur ou égal à 32768. Il est également possible de choisir un facteur de regroupement plus grand, afin de réduire la taille du fichier Analyze 7.5 final. Le programme permet aussi à l'utilisateur d'entrer les dimensions réelles de la coupe, qui seront utilisées plus tard par Biomap.

Une fois qu'un nom de fichier de destination a été choisi, la conversion peut commencer. Pendant la première phase de la conversion, les données binaires des points de chaque spectre sont lues et combinées si nécessaire. Les spectres sont traités ligne par ligne, en commençant par le pixel en haut à gauche pour finir par le pixel en bas à droite. Pendant la compression, l'intensité du plus haut pic dans le bin est retenue comme intensité du bin. Les points sont écrits dans un fichier temporaire (.tmp) comme valeurs 32-bits (long integers). Dans la seconde partie de la conversion, les valeurs 32-bits sont converties en valeurs 16-bits (integers) et écrites dans le fichier image final (.img), après quoi le fichier temporaire est détruit. Dans le fichier temporaire et dans le fichier image, les points de chaque spectre sont écrits comme de l'information continue, dans l'ordre dans lequel les spectres sont traités. Le fichier image final requiert un fichier en-tête (.hdr) qui décrit la structure des données dans un fichier image. Dans ce fichier en-tête, les points contigus sont définis comme des blocs d'information avec une longueur d'un spectre (première dimension), ordonnés selon un certain nombre de pixels horizontaux (deuxième dimension) et de pixels verticaux (troisième dimension). Lors de l'import dans Biomap, cette information qui est plus facile et plus rapide à écrire par rapport au tableau de spectres est convertie dans la forme plus intuitive : X pixels (première dimension), Y pixels (deuxième dimension), troisième dimension non utilisée (dans la mesure où le fichier décrit une simple coupe et non un organe entier), ainsi que les points des spectres (quatrième dimension). Etant donné que le fichier image contient seulement les intensités de chaque point, un fichier séparé est nécessaire pour décrire la valeur de la masse de chaque point. Pour ce fichier appelé .t2m, les valeurs de masse correspondant aux points dans le fichier image final (c'est-à-dire de chaque bin des spectres originaux) sont calculées

en utilisant les valeurs de calibration lues dans le fichier de paramètres d'acquisition accompagnant les spectres sur le disque.

La limite théorique du logiciel pour le nombre de pixels est 1 million (de X(000)/Y(000) à X(999)/Y(999)) ; cependant, cette limite n'a pas été testée. De très grands nombres de pixels ralentiront l'opération sur l'ordinateur exécutant l'acquisition, en particulier sur les anciens systèmes. Le temps d'acquisition d'une image sur un UltraFlex est largement inférieur à celui sur un Reflex IV, non seulement à cause de la fréquence de répétition des tirs laser (jusqu'à 200 Hz sur l'Ultraflex II et seulement 50 Hz sur le Reflex IV), mais aussi à cause du mécanisme d'orientation. En effet, le déplacement de la plaque entre deux points d'acquisition dans le cas de l'Ultraflex II ne nécessite qu'environ 1 seconde alors qu'il peut nécessiter jusqu'à 8 secondes sur un Reflex. Le temps mis par Analyze This ! pour convertir les spectres en un fichier image .img est d'approximativement 25 minutes.

A titre d'exemple, la Figure 22 présente la comparaison entre la photographie d'une coupe colorée au crésyl violet (en haut à gauche) et l'image moléculaire issue des données par spectrométrie (en haut à droite) pour les signaux correspondant aux ions à  $m/z$  14160 (en jaune) et  $m/z$  10300 (en bleu). L'image dans la partie inférieure présente un schéma provenant de l'atlas du cerveau de rat, dans le plan de coupe correspondant aux coupes étudiées, sur lequel ont été annotées en jaune des régions correspondant à la localisation de l'ion à  $m/z$  14160.



**Figure 22: Image recomposée représentant, à gauche, une coupe de cerveau de souris adulte colorée au violet de Crésol et, à droite, l'image reconstruite avec Analyze This! et BioMap pour le signal enregistré à  $m/z$  14 160 (jaune), avec la zone de la coupe visible grâce au  $m/z$  10 300 (bleu) ; en bas, le schéma correspondant à l'atlas du cerveau de souris.<sup>95</sup>**

## 3.3 Outils pour l'IMS dédiés aux autres instruments

### 3.3.1 Axima Vizualisation software

Les appareils Axima de chez Shimadzu disposent d'une fonctionnalité logicielle appelée « Functionality Mapping » permettant de visualiser la localisation de peptides d'une masse d'intérêt choisie ou de la totalité des ions courants. Ce logiciel très récemment délivré est encore en cours de développement et peu d'informations sont disponibles à l'heure actuelle sur son mode de fonctionnement.

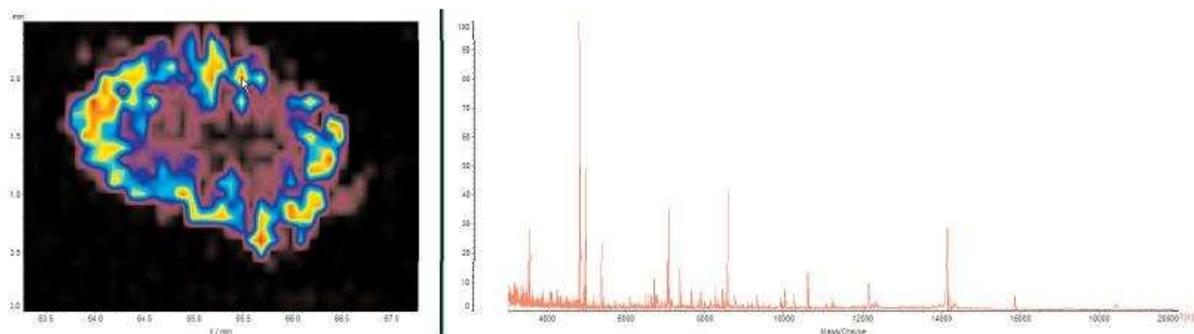


Figure 23 : Image d'une coupe de chaîne nerveuse de souris (2.5 x 4 mm). Le spectre montre les pics présents à un endroit donné sur la carte, déterminé par l'utilisateur.  
(<http://www.shimadzu.fr/en/products/biotech/maldi/ChipAssurance/default.aspx>)

### 3.3.2 Axima2Analyze

Sur le site MALDI-msi.org (<http://www.maldi-msi.org/>), se trouve également un logiciel développé par l'équipe de M. Setou appelé Axima2Analyze et dédié à l'imagerie sur les appareils Shimadzu AXIMA QIT. Ce logiciel permet la conversion des données issues des instruments Shimadzu pour leur relecture sous Biomap afin de permettre la reconstruction d'images.

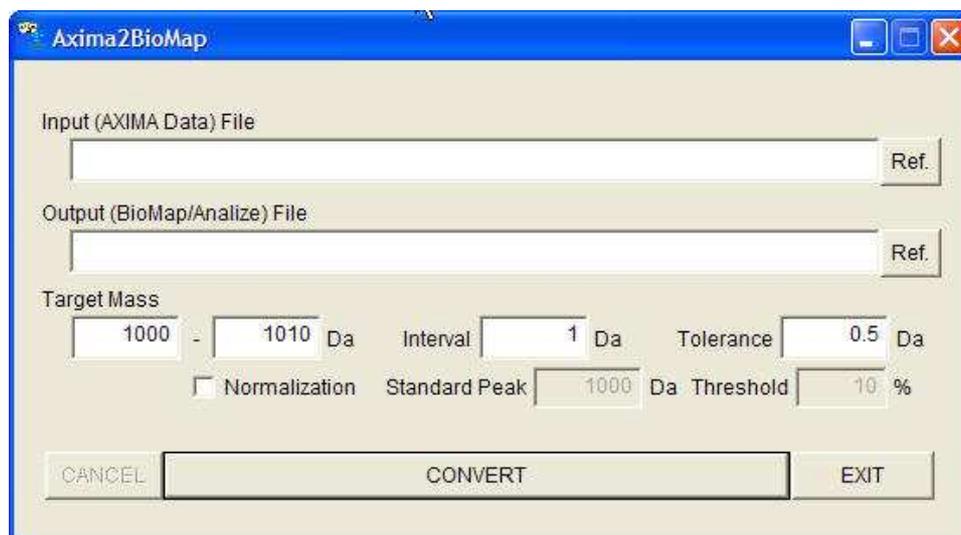


Figure 24: Interface du logiciel Axima2Biomap (<http://www.maldi-msi.org/>)

### 3.3.3 IMT

Le logiciel IMT développé par Alain Brunelle et David Touboul (<http://www.fist.fr/article385.html>) peut être utilisé avec tout type de spectromètre de masse réalisant des analyses MALDI-TOF. Il gère les données provenant d'un spectromètre de masse pour leur acquisition et leur traitement. Les spectres sont recueillis sous forme d'une matrice permettant de reconstruire des images en couleur représentant la localisation sur une coupe de peptides d'après leur rapport masse sur charge. Des spectres peuvent être reconstitués pour des zones choisies et différentes fonctionnalités sont disponibles pour réaliser des projections, soustractions, additions voire divisions d'images ou encore images en 3D. Ce logiciel a été enregistré à l'agence pour la protection des programmes en vue d'être intégré et/ou commercialisé dans le cadre d'une licence exclusive.

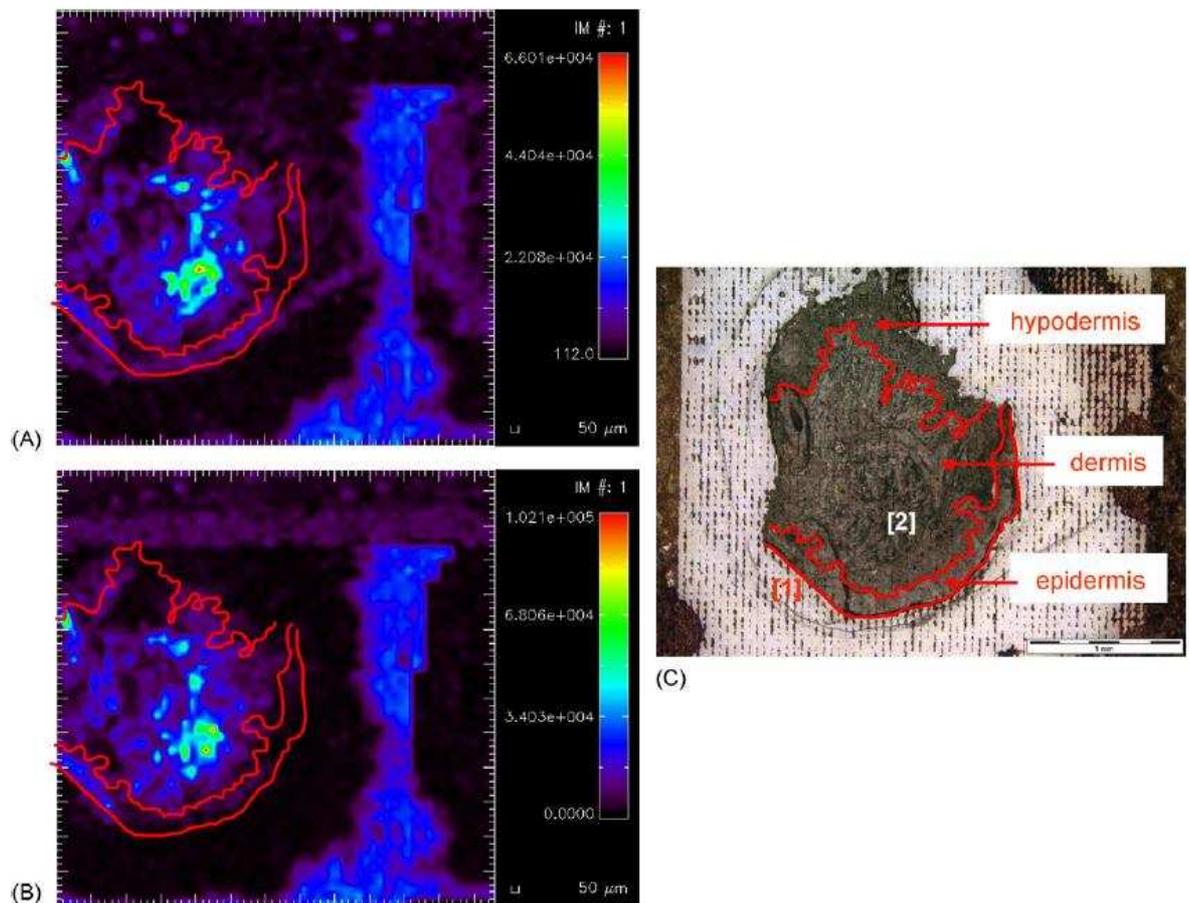


Figure 25 : Images reconstruites avec IMT<sup>96</sup> : MALDI-TOF images de (A) l'ion de  $m/z$  1010.8 ( $Ga_2$ ) et (B) l'ion de  $m/z$  1158.9 ( $Gb_3$ ) acquises à partir d'une coupe de biopsie de peau d'un patient de la maladie de Fabry. Image de dimensions 3.5 mm x 3.5 mm. (C) Image optique après l'acquisition des images MALDI. Les impacts du laser sont clairement visibles en dehors de la coupe. La barre d'échelle représente 1mm.

## 3.4 Comparaison des différents logiciels dédiés à l'IMS

Le Erreur ! Source du renvoi introuvable. résume les caractéristiques des différents logiciels d'imagerie vus précédemment.



---

### Résultats & Discussions

Comme présenté précédemment, l'IMS MALDI est une technologie récente qui nécessite l'utilisation d'outils bioinformatiques efficaces, performants et adaptés. La rapide progression des développements et les applications réalisées dans le domaine académique ont souvent laissé en retrait les sociétés fournissant les instruments de spectrométrie de masse qui sont, pour la plupart, loin de proposer à l'heure actuelle des solutions efficaces. Ainsi, les solutions dédiées à l'acquisition des données et au traitement des images sont assez disparates et proviennent en grande majorité de développements à façon pour les utilisations en laboratoire de recherche. Les logiciels ne regroupent en général pas l'ensemble des fonctionnalités souhaitées par un utilisateur et ne sont que rarement compatibles avec les données émanant de différents instruments. De plus, peu d'informations sont disponibles sur le traitement des données et ces logiciels restent quelque peu des boîtes noires. C'est dans ce contexte que nous avons souhaité développer au laboratoire notre propre logiciel d'acquisition puis de traitement des données et reconstruction d'images. Le logiciel peut s'adapter à tous types de données. Nous pouvons contrôler le traitement des données et définir les algorithmes à utiliser tout en ajoutant progressivement des fonctions au gré des nécessités rencontrées lors des applications. Ceci offre une grande souplesse et un contrôle du traitement des données.

#### 4.1 Le logiciel MITICS

Le logiciel est constitué de deux parties. La première sert à paramétrer l'acquisition des données de l'analyse par spectrométrie de masse et s'utilise donc avant l'acquisition proprement dite. La seconde sert à traiter les données et à reconstruire l'image, elle nécessite le traitement des spectres produits par l'analyse par spectrométrie de masse et sera donc employée après l'acquisition des données. La Figure 26 présente les grandes étapes réalisées par le logiciel dépendamment des instruments utilisés soit pour la série Ultraflex de chez Bruker Daltonics (flèches bleues) soit pour la série Voyager de chez Applied Biosystems (flèches rouges).

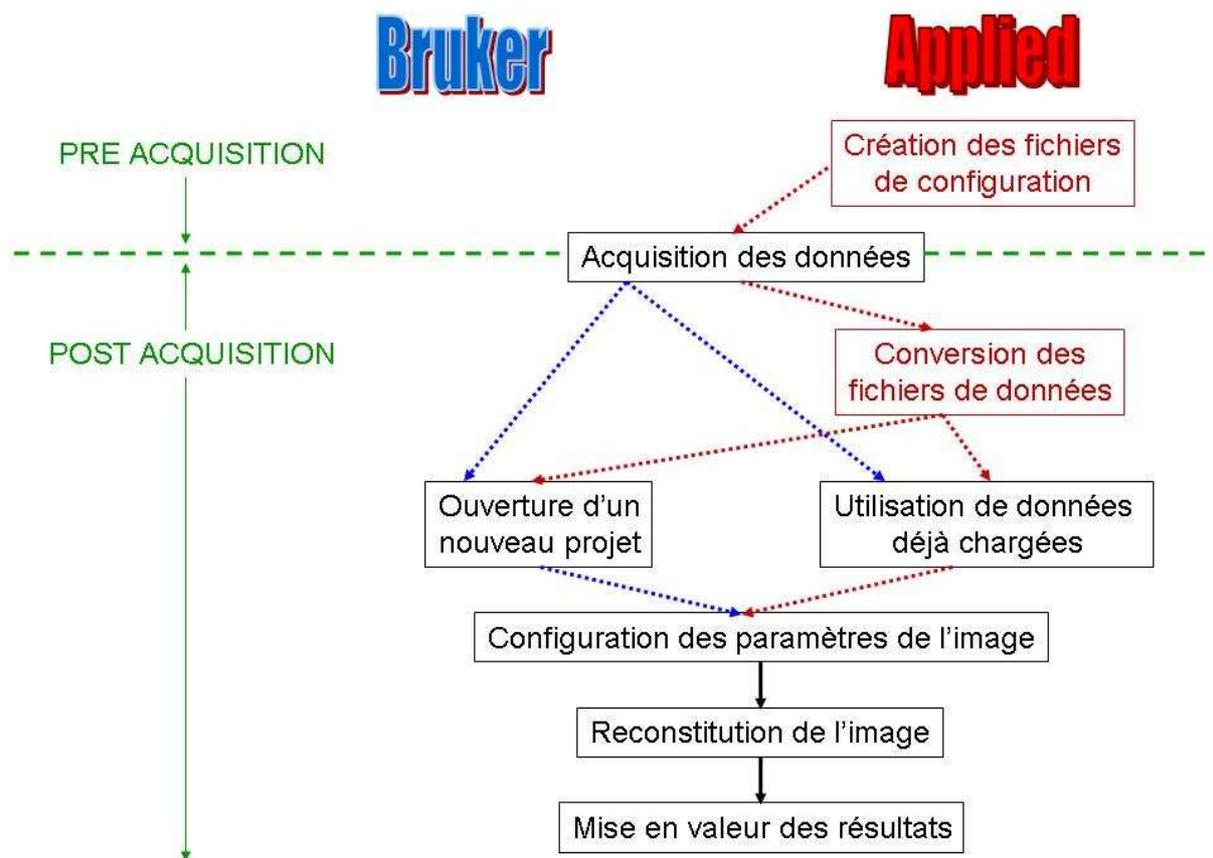


Figure 26: Schéma du fonctionnement du logiciel MITICS pour deux séries d'instruments distinctes

#### 4.1.1 Pré-acquisition : MITICS Control

MITICS Control a pour vocation de permettre l'acquisition des données sur le spectromètre en présentant une interface facile à utiliser dialoguant avec les logiciels habituels de l'instrument.

La première partie du logiciel consiste à paramétrer le balayage qui sera effectué sur la plaque du spectromètre. Pour cela, il faut créer trois fichiers : un fichier de format .BIC, un fichier de format .PLT et un fichier de format .SEQ qui seront fournis à la machine pour l'acquisition des données. Elle permet de s'affranchir de l'utilisation des logiciels Voyager Instrument Control Panel et Sequence Control Panel gouvernant respectivement l'acquisition et le mode automatique du spectromètre de masse Voyager Applied Biosystems. C'est dans l'interface de Sequence Control Panel que sont définis notamment les points qui seront balayés par le faisceau laser en mode automatique. La création du fichier .bic par le logiciel se substitue à la définition des paramètres de l'acquisition par le programme Voyager Instrument Control Panel, tandis que les interfaces permettant de créer les fichiers .plt et .seq se substituent au logiciel Sequence Control Panel pour le réglage des valeurs nécessaires au fonctionnement en mode automatique.

Cette partie a été programmée en C<sup>++</sup>. Ce langage a été préféré à d'autres compte tenu de sa rapidité d'exécution, mais il s'agit également d'un langage complet, performant, généraliste, puissant et dont le code source peut être compilé sous différents systèmes d'exploitation.

Un diagramme et une description des objets utilisés dans le code de ce programme se trouvent en annexe.

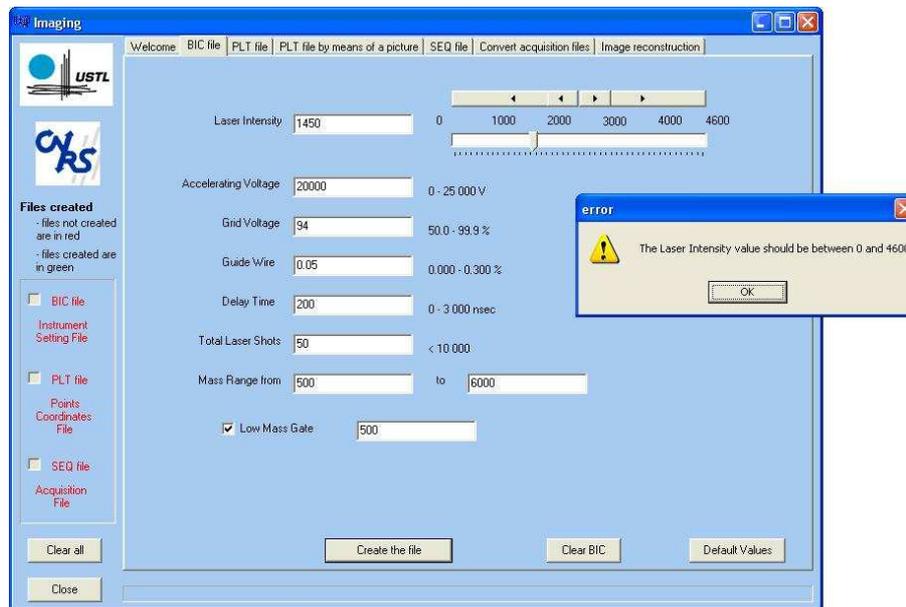


Figure 27: Interface pour la création du fichier .bic

Pour paramétrer le balayage qui va être réalisé sur la plaque, l'utilisateur doit, dans un premier temps, définir la configuration utilisée pour l'analyse. Ainsi, il doit d'abord créer le fichier .bic où le spectromètre de masse ira chercher les différentes valeurs nécessaires pour effectuer l'expérimentation voulue, soit les paramètres de l'analyse *i.e.* l'intensité laser, le nombre de tirs du laser, la tension d'accélération, la tension appliquée à la grille, la tension sur le guide d'ions, le temps de retard à l'extraction, la gamme de masse ou encore le seuil minimum de masse. L'ensemble de ces valeurs sera enregistré dans le fichier .bic grâce à l'onglet « BIC file » de l'interface graphique (GUI). Chacune de ces valeurs ne doit pas dépasser certaines limites spécifiques, celles-ci sont donc précisées à côté de chaque zone de texte à remplir. De plus, pour éviter d'éventuelles erreurs, les valeurs entrées par l'utilisateur sont vérifiées et un message d'erreur apparaît si elles sont en dehors des limites raisonnables comme montré sur la Figure 27. Pour l'intensité laser, trois possibilités s'offrent à l'utilisateur: selon ses préférences, il peut simplement entrer la valeur voulue à l'aide du clavier dans la zone prévue à cet effet, ou bien faire varier le curseur en cliquant dessus puis en le faisant glisser, ou encore cliquer sur les flèches situées sous la barre graduée (une simple flèche pour une faible variation, une double flèche pour une variation plus importante).

La Figure 28 schématise sur le spectromètre les différents paramètres présentés auparavant et dont l'utilisateur peut régler les valeurs.

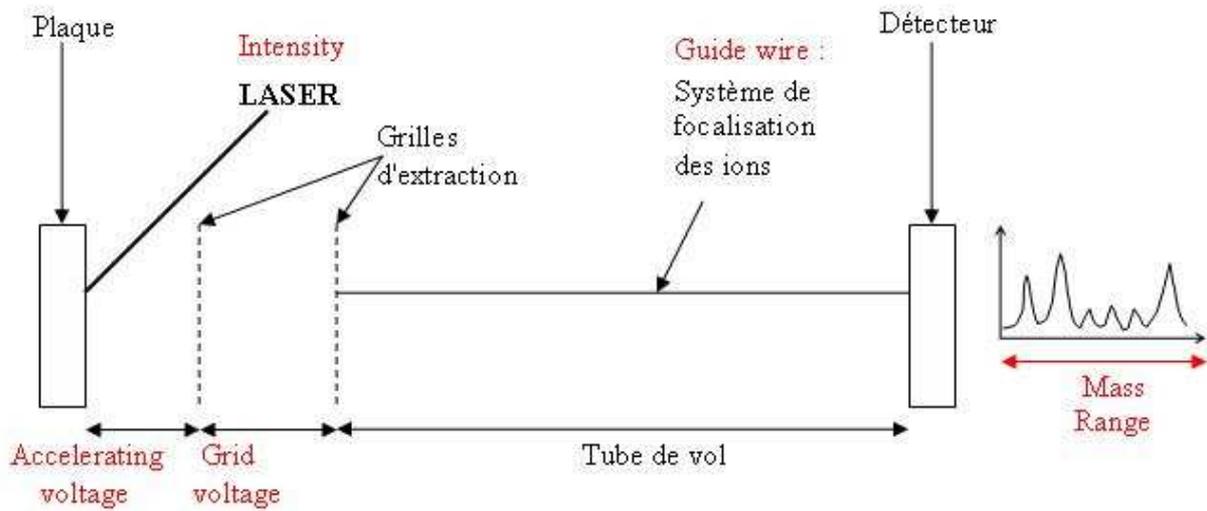


Figure 28: Paramètres réglables par l'utilisateur (en rouge)

Dans la partie gauche de la fenêtre se trouvent trois cases à cocher liées respectivement au fichier .bic, au fichier .plt et au fichier .seq. Afin de faciliter le suivi de la manipulation en cours, quand l'utilisateur a créé l'un de ces fichiers, son nom apparaît alors en vert au lieu de rouge et la case correspondante est cochée.

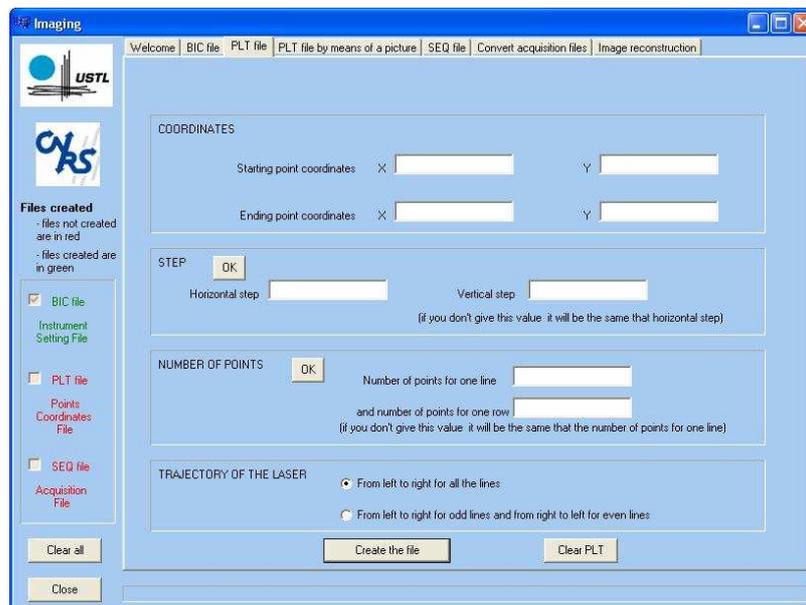


Figure 29: Interface pour la création du fichier .plt

L'étape suivante consiste en la création du fichier .plt (Figure 29) afin de se libérer du logiciel « Sequence Control Panel » qui configure le balayage réalisé par le laser sur la plaque. Les coordonnées de chaque point où le laser va frapper la plaque sont sauvegardées dans le fichier .plt. Elles sont calculées par le logiciel en fonction des informations fournies par l'utilisateur : les coordonnées du point de départ, du point d'arrivée et, soit le pas entre deux points, soit le nombre de points sur une ligne et sur une colonne. Ainsi l'utilisateur peut choisir le nombre de points exact qu'il souhaite ou bien le pas précis éloignant deux points l'un de l'autre en X et en Y, sachant que ce nombre ou ce pas ne sont pas forcément identiques horizontalement et verticalement pour une plus grande flexibilité. Une fois ces données fournies par l'utilisateur, le programme va calculer les renseignements qui lui manquent, c'est-à-dire le nombre de points si le pas lui a été indiqué, ou le pas si c'est le nombre de points qui lui a été donné. La zone à balayer par le laser peut aussi être définie par un « cliquer-glisser » de la souris sur une image de la plaque fournie par l'utilisateur comme sur la Figure 30. Dans ce cas, il s'avère nécessaire d'avoir trois points de repère pour assurer la concordance entre les coordonnées sur l'image et les coordonnées réelles sur la plaque. Les trois points de référence seront placés correctement à l'aide d'un « cliquer-glisser » avec le bouton droit de la souris tandis que la zone d'intérêt sera définie en dessinant un rectangle sur l'image de la coupe par un « cliquer-glisser » avec le bouton gauche de la souris.

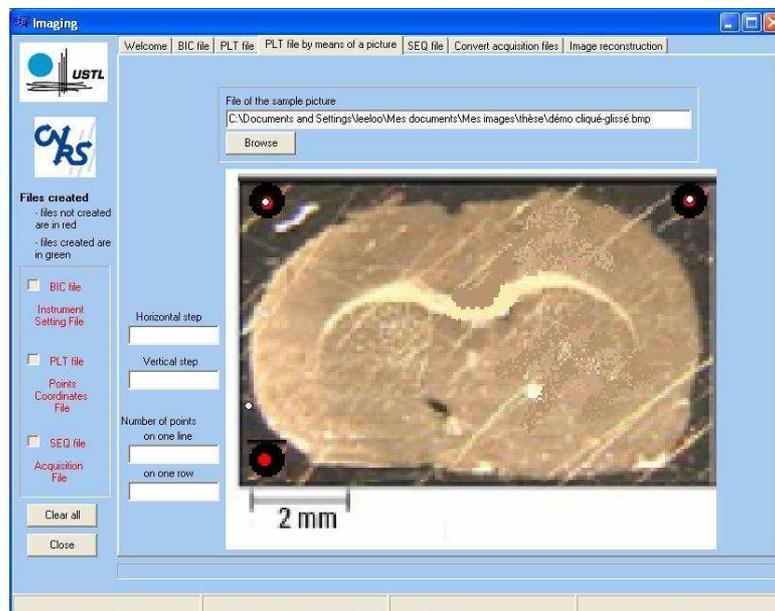


Figure 30: Interface pour la cration du fichier .plt à l'aide d'une vue de la plaque

La troisième étape, quant à elle, permet à l'utilisateur de configurer l'automatisation de l'acquisition des spectres (Figure 31). Ainsi le cinquième onglet de l'interface graphique appelé « SEQ file » va remplacer lui aussi le programme « Sequence Control Panel ». Le troisième fichier contient des informations concernant les fichiers utilisés par le logiciel pour

paramétrer l'acquisition des données : le fichier .bic, le fichier .plt et le répertoire où les fichiers résultats seront enregistrés. Les fichiers créés précédemment seront automatiquement inscrits pour gagner du temps. Cette étape permet également l'application éventuelle d'une macro sur les fichiers résultats. Par exemple si l'utilisateur veut automatiquement corriger la ligne de base de chaque spectre ou réduire le bruit de fond, il peut entrer le nom de la fonction adéquate dans la zone de texte adaptée et la manipulation demandée sera effectuée juste avant l'enregistrement des spectres acquis.

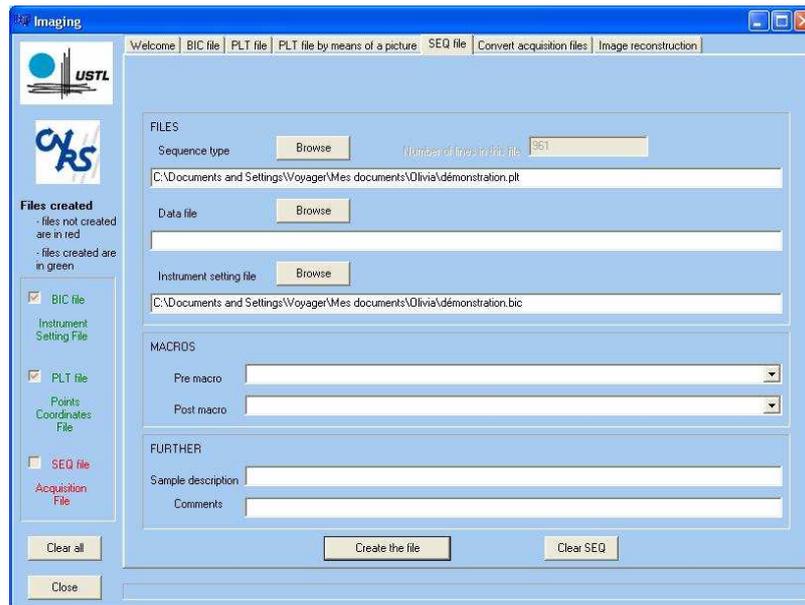


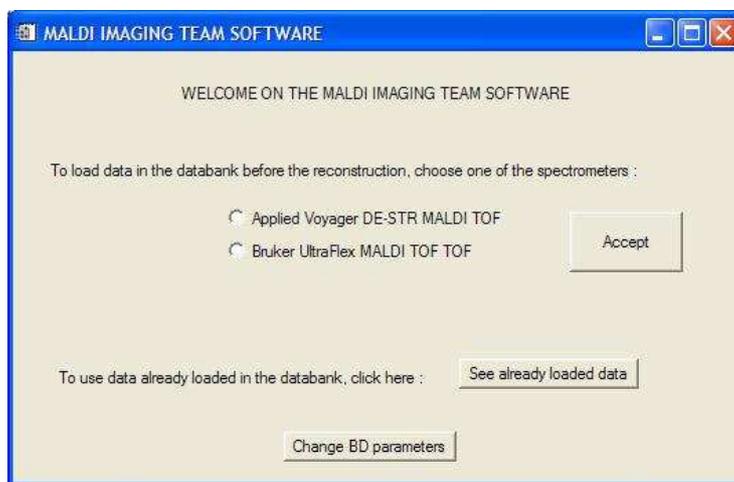
Figure 31: Interface pour la création du fichier .seq

### 4.1.2 Post-acquisition : MITICS Image

La deuxième partie du logiciel gère le traitement des données et la création de l'image proprement dite. L'analyse réalisée par le spectromètre va aboutir à un nombre considérable de fichiers contenant chacun un spectre sous la forme d'une liste de pics. Le programme va donc devoir parcourir tous ces fichiers afin d'accéder aux données des spectres pour reconstruire l'image pour un pic d'intérêt.

Cette partie a été programmée en IDL (Interactive Data Language) car ce langage intégré offre un grand nombre de possibilités en matière d'analyse et de visualisation des données. Il supporte beaucoup de formats, types et tailles de données et contient des routines natives de traitement d'images, entre autres. Ainsi il permet de construire facilement des visualisations graphiques évoluées et performantes. De plus, il dispose d'un environnement de développement fournissant une interface pratique pour écrire du code, le compiler et le déboguer, et intégrant une interface graphique utilisateur (GUI) qui va faciliter la conception d'une interface puis générer automatiquement le code correspondant.

Le principal avantage de MITICS Image est qu'il permet de reconstruire une image soit à partir de spectres acquis sur un instrument Applied Biosystems, soit à partir de spectres acquis sur un instrument Bruker Daltonics. Cependant, le format des données enregistrées après une analyse sur ces spectromètres de masse n'étant pas identique pour les deux constructeurs, le logiciel présente obligatoirement deux interfaces séparées pour le chargement des spectres dans la base de données. De plus, le programme offre la possibilité de réutiliser des spectres chargés précédemment dans la base de données pour reconstruire une nouvelle image, sans nécessiter de reprendre l'étude au début par l'enregistrement des données. Enfin s'il existe plusieurs bases de données reliées, elles doivent être correctement paramétrées. La **Figure 32** présente la première boîte de dialogue visualisée par l'utilisateur lors de l'utilisation de MITICS Image.



**Figure 32: Page d'accueil de MITICS pour la reconstruction d'image**

Sur cette page d'accueil, l'utilisateur retrouve les trois possibilités évoquées antérieurement :

- 1 - Une analyse du début à la fin, du parcours des fichiers contenant les spectres jusqu'à la reconstruction de l'image proprement dite, à partir soit d'un spectromètre Voyager Applied Biosystems, soit d'un spectromètre UltraFlex Bruker Daltonics
- 2 - La reconstruction d'une image à partir de données précédemment chargées dans la base de données lors d'une analyse complète
- 3 - La configuration des paramètres de la base de données où les données sont chargées.

#### **4.1.2.1 Premier choix : Nouveau projet**

Afin d'améliorer le traitement des données collectées, une base de données a été ajoutée au logiciel dans ses dernières versions. Cela permet à l'utilisateur de stocker les données pour pouvoir reconstruire des images ultérieurement. De plus, pour accélérer les calculs et optimiser le parcours des spectres, la collecte des données dans les fichiers est effectuée à

l'aide d'un parseur XML (Extensible Markup Language). Ainsi, si l'utilisateur choisit d'ouvrir un nouveau projet, il doit charger les nouvelles données dans la base de données. Deux possibilités s'offrent à lui : il peut soit cocher la case Applied Biosystems Voyager DE-STR MALDI TOF, soit cocher la case Bruker Daltonics UltraFlex MALDI TOF TOF avant de cliquer sur le bouton « Accept ».

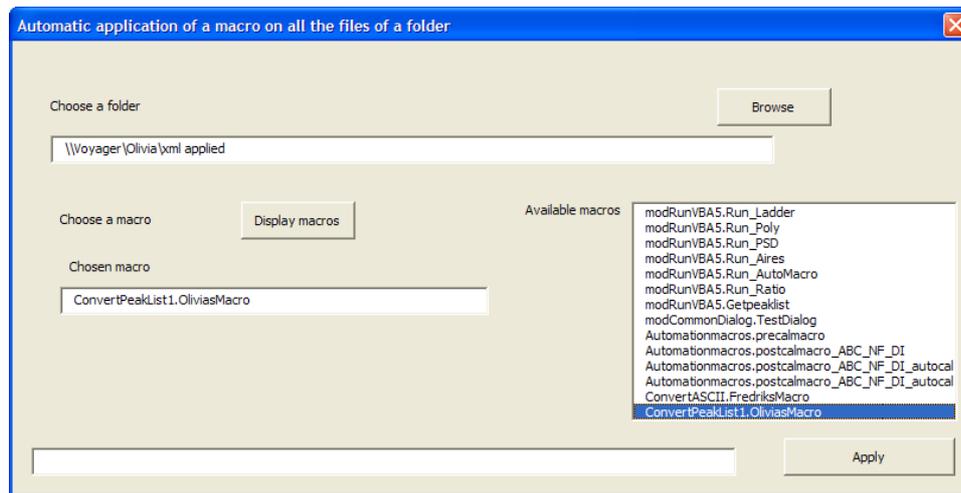
XML (*eXtensible Markup Language* c'est-à-dire *Langage à balises étendu*, ou *Langage à balises extensible en français*), mis au point par le World Wide Web Consortium (W3C) dès 1996 est libre de droits, indépendant des plates-formes. C'est en quelque sorte un langage HTML amélioré permettant de définir de nouvelles balises. Il s'agit effectivement d'un langage permettant de mettre en forme des documents grâce à des balises (markup). Il peut être considéré comme un métalangage permettant de définir d'autres langages, c'est-à-dire définir de nouvelles balises permettant de décrire la présentation d'un texte. La force de XML réside dans sa capacité à pouvoir décrire n'importe quel domaine de données grâce à son extensibilité. Il va permettre de structurer, poser le vocabulaire et la syntaxe des données qu'il va contenir. Les balises XML décrivent plus le contenu des données, que leur présentation, séparant ainsi les deux. XML permet donc de définir un format d'échange selon les besoins de l'utilisateur et offre des mécanismes pour vérifier la validité du document produit. Il est donc essentiel pour le receveur d'un document XML de pouvoir extraire les données du document. Cette opération est possible à l'aide d'un outil appelé analyseur (en anglais parser, parfois francisé en *parseur*). Le parseur permet d'une part d'extraire les données d'un document XML (on parle d'*analyse* du document ou de *parsing*) ainsi que de vérifier éventuellement la validité du document. Les principaux atouts d'XML résident tout d'abord dans son caractère autodéscriptif et extensible. De plus, il possède une structure arborescente permettant de modéliser la majorité des problèmes informatiques. Son universalité et sa portabilité proviennent des différents jeux de caractères pris en compte. Il peut être facilement distribué par n'importe quels protocoles à même de transporter du texte, comme http, ce qui le rend déployable. Un document XML étant utilisable par toute application pourvue d'un parseur (c'est-à-dire un logiciel permettant d'analyser un code XML), ce langage est intégrable. Ainsi, XML est particulièrement adapté à l'échange de données et de documents.

Pour la reconstruction d'images à partir de spectres provenant d'un spectromètre de masse Bruker Daltonics, les seuls fichiers lisibles disponibles sont les fichiers XML (Bruker ne permettant pas l'accès à ses codes sources); d'où l'écriture d'un parseur pour analyser les données des spectres. Ensuite il a été décidé d'utiliser la même méthode pour gérer les spectres provenant d'un appareil Applied Biosystems en raison de la réduction du temps nécessaire pour récupérer les données avec cette technique par rapport à l'ouverture de chaque fichier texte réalisée auparavant. Cependant lors d'une analyse le spectromètre Voyager d'Applied Biosystems ne fournit pas de fichier XML, il a donc fallu traduire les fichiers de spectres de format .dat en fichiers XML avant de pouvoir traiter les données.

## **Dans le cas d'une analyse avec un spectromètre de masse Applied Biosystems :**

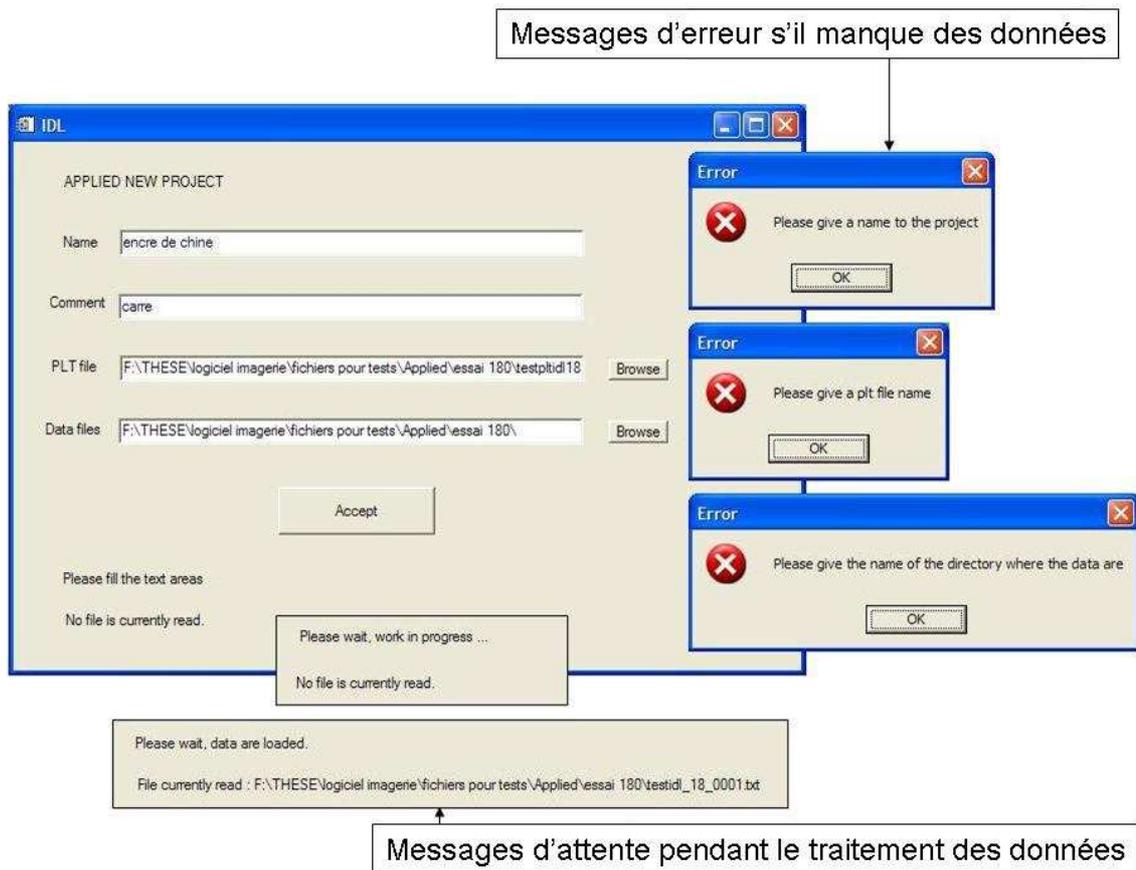
Dans le cas d'une analyse réalisée sur un spectromètre de masse Applied Biosystems, l'instrument produit des spectres au format .dat. Les fichiers .dat sont des fichiers d'un format propriétaire qui ne sont lisibles que par l'intermédiaire du programme de traitement des spectres Data Explorer fourni avec l'instrument. Sachant qu'il n'a pas été possible d'obtenir les codes sources auprès de la société pour relire directement les données, celles-ci ne peuvent donc pas être utilisées pour la reconstruction de l'image. Cependant, il est possible d'exporter les données pour recréer des fichiers .txt présentant les données sous la forme de deux colonnes correspondant à chaque valeur de  $m/z$  et l'intensité de mesure correspondante. Les fichiers .txt constituent un excellent moyen de pallier au problème de la relecture des données. Cependant, la lecture dans un fichier de ce type répétée un très grand nombre de fois, comme nécessaire dans une analyse pour recréer une image par spectrométrie de masse, consomme énormément de temps. De plus les fichiers .txt présentent une compression bien inférieure à celle des fichiers .dat et génèrent une collection de données d'une taille importante (environ 900 Ko par spectre). Un parseur pour extraire les données des spectres de l'instrument Bruker Daltonics ayant déjà été écrit auparavant, son utilisation afin de recueillir les données des spectres de l'instrument Applied Biosystems a été décidée. Ainsi, l'utilisateur doit, en premier lieu, transformer les fichiers de spectres .dat en fichier .xml gérés par le parseur XML. Cette étape supplémentaire permet de réduire le temps nécessaire à la récupération des données dans les spectres. Une autre solution envisagée au début de la conception du logiciel consistait à regrouper tous les fichiers textes en un seul, plus facilement et rapidement accessible ensuite. Cependant la création de ce fichier posait problème à cause du grand nombre de données. En effet, certains formats d'organisation de fichiers comme le format fat 32 (*file allocation table*) comportent des limites au niveau de la taille des fichiers. Ce dernier est utilisé généralement dans les anciennes versions du système d'exploitation Windows, comme Windows NT qui est installé sur l'ordinateur gérant le spectromètre MALDI-TOF Applied Biosystems dans notre laboratoire. Dans ce format, la taille d'un fichier ne peut pas excéder 4 Giga octets. Or les dernières évolutions apportées aux traitements réalisés avant l'acquisition des données pour l'imagerie par spectrométrie de masse ont permis d'améliorer la résolution des images reconstruites. Autrement dit, les analyses produisent de plus en plus de fichiers à gérer lors de la reconstruction de l'image. Ainsi cette première idée de regroupement des données a été abandonnée pour éviter d'avoir à gérer un fichier d'une telle taille, au profit d'un parcours de tous les fichiers textes afin de récolter les données. Mais, comme nous l'avons déjà précisé, cette méthode est très gourmande en temps, ainsi pour accélérer le processus de lecture des spectres, le parseur XML a été adopté. Le spectromètre Bruker Daltonics produisant déjà des fichiers XML possédant une grammaire associée, le plus simple consistait à générer des fichiers XML respectant les règles de la même grammaire, cela permettant ensuite d'utiliser le même algorithme de traitement des données.

Dans cette optique, nous avons réalisé une fonction codée en Visual Basic afin d’être utilisable facilement avec Data Explorer (**Figure 33**). Ce petit programme permet de choisir une macro et de l’appliquer à tous les fichiers contenus dans un répertoire donné. Ainsi une autre fonction a été écrite, qui transforme un fichier .dat en un fichier .xml avec les mêmes caractéristiques que les fichiers .xml obtenus à partir d’une analyse effectuée sur un spectromètre de masse Bruker Daltonics. Ces fichiers .xml seront utilisés par MITICS pour recréer les images.



**Figure 33: Interface permettant de convertir les fichiers .dat au format .xml**

Une fois que ces fichiers sont créés, l’utilisateur peut lancer le logiciel d’imagerie et ouvrir un nouveau projet à l’aide de l’interface présentée sur la **Figure 34**.



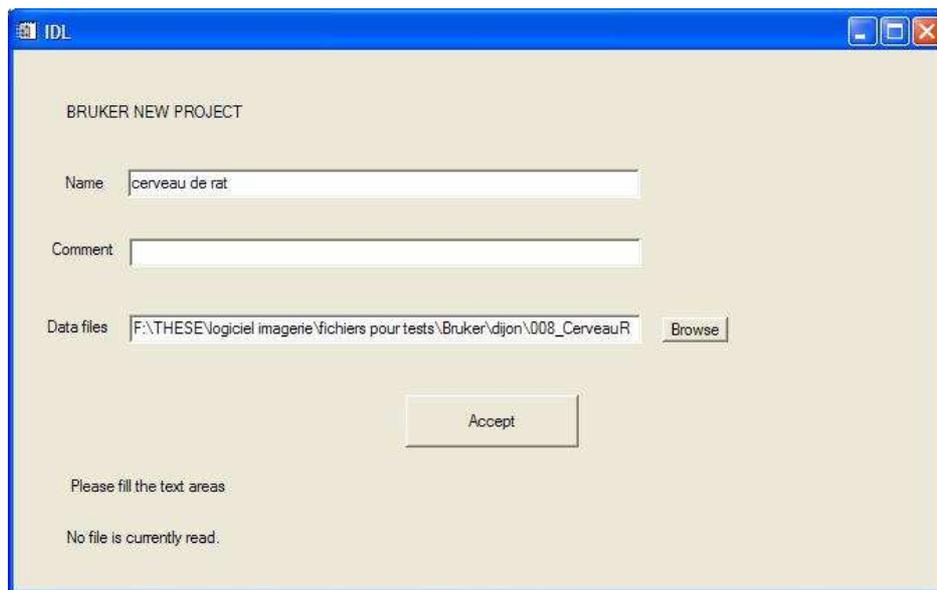
**Figure 34: Interface pour un projet sur un appareil Applied Biosystems**

Si l'utilisateur oublie de renseigner un de trois champs nécessaires, un message d'erreur apparaît demandant d'entrer un nom pour ce paramètre. Il fournit un nom, un commentaire si besoin, un fichier .plt (contenant les coordonnées des pixels de l'image, comme vu précédemment) et enfin le répertoire où les fichiers .xml se trouvent. Il peut parcourir les disques de l'ordinateur pour trouver les fichiers d'intérêt. La collecte des données est ensuite lancée en cliquant sur le bouton "Accept". Pendant le traitement des données, les messages en bas de la fenêtre varient pour informer l'utilisateur de l'avancement de l'analyse. Deux exemples sont présentés sur la **Figure 34**.

A la fin du traitement, une nouvelle fenêtre s'ouvre demandant à l'utilisateur de fournir quelques paramètres concernant l'image à reconstruire. Cette interface sera présentée après l'explication concernant l'ouverture d'un nouveau projet concernant un instrument Bruker Daltonics.

## Dans le cas d'une analyse avec un spectromètre Bruker Daltonics :

En cochant “Bruker Daltonics UltraFlex MALDI TOF TOF” et en cliquant sur le bouton “Accept” dans la fenêtre d'accueil, l'utilisateur ouvre un nouveau projet Bruker Daltonics comme montré sur la **Figure 35**. Il doit fournir un nom, un commentaire s'il le désire et l'emplacement des fichiers de données à analyser. Pour cette étape, il peut cliquer sur le bouton « Browse » pour chercher les fichiers dont il a besoin sur les disques durs de l'ordinateur, comme déjà vu précédemment. Pour lancer la collecte des données, il clique sur le bouton « Accept ».



**Figure 35: Interface pour un projet sur un appareil Bruker Daltonics**

Comme énoncé antérieurement, si l'utilisateur oublie de renseigner un champ indispensable, un message d'erreur le prévient. De même, pendant le traitement des données, les messages en bas de la fenêtre changent pour informer l'utilisateur de l'avancement de l'analyse. A la fin de l'analyse, une nouvelle fenêtre s'ouvre, demandant à l'utilisateur d'entrer les paramètres de configuration de l'image. Celle-ci sera présentée sur la **Figure 38**, après l'interface permettant de reconstruire une image à partir de données déjà chargées dans la base de données.

Parmi les différents tests réalisés, un chargement de 1754 fichiers a duré environ 25 minutes, 41 minutes ont été nécessaires pour enregistrer 6972 spectres dans la base de données, ou encore 45 mn pour 16 000 points pour un ordinateur fonctionnant sous Windows XP avec un processeur Intel Pentium 4 de 3.2GHz et 1.00 GO de RAM (toutes les durées précisées par la suite ont été mesurées sur ce même ordinateur).

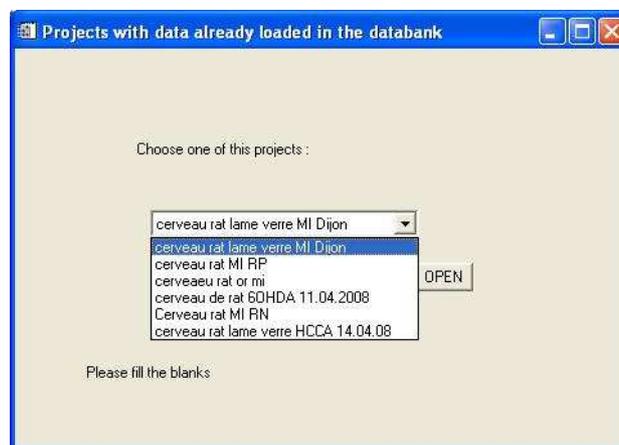
#### 4.1.2.2 Deuxième choix : Utiliser des données déjà chargées dans la base de données

Si l'utilisateur souhaite revenir sur un ancien projet afin de créer une nouvelle image à partir d'une valeur m/z différente, par exemple, il n'est pas nécessaire de reprendre toute l'expérimentation dès le début. En effet, il est possible de créer une image à partir de données déjà chargées dans la base de données. Pour cela, il suffit d'utiliser la fonction proposée par le bouton "See already loaded data" qui ouvre la fenêtre présentée sur la **Figure 36**.



**Figure 36: Interface pour accéder aux projets déjà chargés dans la base de données**

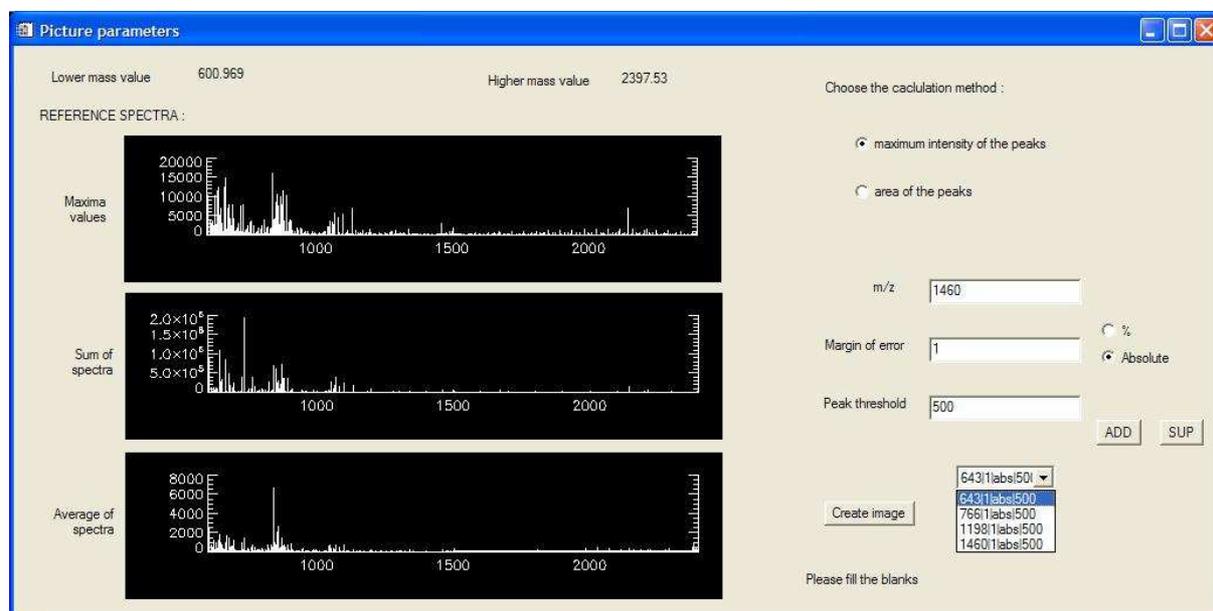
La liste de tous les projets disponibles dans la base de données s'affiche quand l'utilisateur clique sur la flèche du menu déroulant comme le montre la **Figure 37**.



**Figure 37: Affichage des projets présents**

Quand il a choisi, en le sélectionnant, l'un des noms présentés dans la liste, il clique sur le bouton « OPEN » pour atteindre l'étape suivante. Et la fenêtre de la **Figure 38** apparaît environ 7 minutes après dans le cas d'un projet de 12 000 points.

L'étape suivante est la même pour les trois cas décrits ci-dessus (nouveau projet Applied Biosystems, nouveau projet Bruker Daltonics ou réutilisation d'un projet déjà chargé). Il s'agit de définir la configuration des paramètres selon lesquels l'image sera reconstruite.

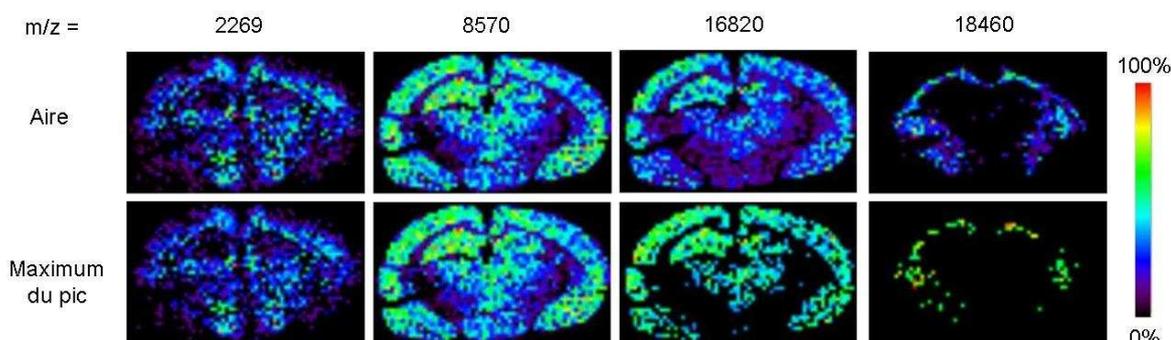


**Figure 38: Interface résumant les données et permettant de configurer les paramètres de l'image**

MITICS fonctionne avec des spectromètres dont l'analyseur est de type TOF. Or les spectres obtenus avec ces appareils montrent parfois une baisse de la résolution liée à l'augmentation du rapport masse-sur-charge. Ainsi une différence d'intensité entre deux pics ne correspond pas forcément à une différence du nombre d'ions détectés. En effet, si l'un des pics est moins haut, mais plus large, les aires seront équivalentes, contrairement à l'intensité mesurée, et traduiront donc mieux le nombre relatif de molécules. Dans ce cas, l'utilisation de l'aire pour les traitements permettra de prendre en compte le fait que la résolution dans un spectre peut ne pas être constante sur l'ensemble de la gamme de masse.

La Figure 39 montre bien que plus le rapport  $m/z$  augmente, plus la différence entre le signal observé pour l'aire du pic se distingue du signal observé pour le maximum du pic. En effet, pour les rapports  $m/z$  les plus faibles présentés ici (2269 et 8570), les images recréées avec le maximum du pic sont équivalentes à celles obtenues à partir de l'intensité maximale du pic, tandis que pour un rapport  $m/z$  plus élevé (16820), la différence devient notable. Enfin, pour le rapport  $m/z$  18460, la perte de signal lorsque le calcul est réalisé avec l'intensité maximale du pic au lieu de l'aire du pic se révèle encore plus flagrante. Dans ce cas, l'utilisation de l'aire sera préférable à l'utilisation de l'intensité maximale du pic pour conserver une meilleure cohérence entre les images reconstruites pour les  $m/z$  plus faibles par rapport aux  $m/z$  plus élevés. Si l'étude porte sur une gamme plus restreinte de masses moins

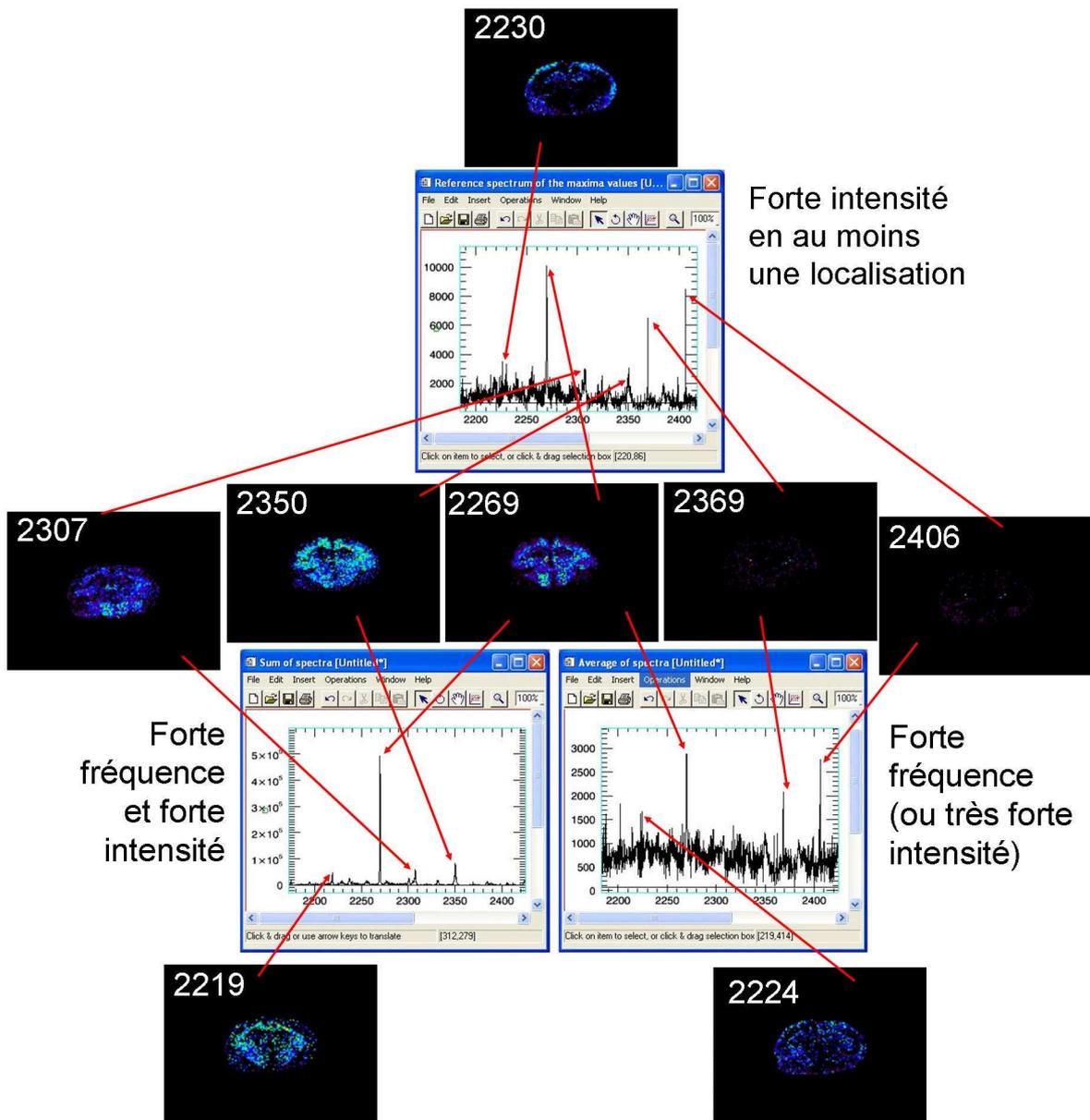
hautes, l'utilisation de l'intensité maximale du pic permet de limiter la marge d'erreur autorisée et de rendre l'analyse plus spécifique à un pic précis.



**Figure 39 : Comparaison entre les images recrées avec MITICS à l'aide de l'aire et celles recrées à l'aide du maximum du pic sur une coupe de cerveau de rat réalisée au laboratoire**

Afin de conserver une meilleure cohérence entre différentes images reconstruites pour une même analyse mais pour des masses différentes et éloignées, le logiciel permet de choisir entre deux types de mesures à utiliser: soit la valeur de l'intensité maximale d'un pic, soit l'aire du pic.

Puis l'utilisateur choisit la valeur du rapport masse-sur-charge d'un peptide d'intérêt dont il veut connaître la localisation sur le tissu. Ensuite il doit fournir une marge d'erreur en pourcentage pour pallier aux éventuels légers décalages des pics d'une même molécule d'un spectre à l'autre. Enfin la dernière valeur à renseigner concerne le seuil en dessous duquel un pic sera considéré comme du bruit de fond et donc ne sera pas pris en compte dans les calculs. Alors l'utilisateur n'a plus qu'à cliquer sur le bouton « Create image » pour que l'image soit recrée. Afin d'aider l'utilisateur à définir ces valeurs, un résumé des données est affiché sur la partie gauche de la fenêtre : la valeur la plus faible et la plus élevée de la gamme de masse et trois spectres de référence. Le premier spectre montre les valeurs maximales des intensités trouvées dans les spectres, ainsi si un pic n'est présent que dans très peu de spectres, il apparaît quand même dans ce spectre de référence sans être masqué par des pics plus fréquemment répartis sur la plaque. Le deuxième spectre affiche la somme de toutes les intensités et le troisième la moyenne des intensités. Dans les premières versions du logiciel, ces spectres étaient recalculés pour chaque lancement du programme, même dans le cas d'une reconstruction d'image à partir de données déjà chargées dans la base de données. Afin d'éviter la perte de temps occasionnée, les spectres de référence sont dorénavant enregistrés dans la base de données pour s'afficher plus rapidement. De plus, cliquer dans l'un des graphiques permet d'ouvrir une fenêtre dans laquelle de nombreux outils sont disponibles pour agrandir le spectre. Ainsi l'évaluation de la marge d'erreur nécessaire est facilitée.



**Figure 40 : Exploitations distinctes des trois spectres de référence dans le cas d'images reconstruites d'après une coupe de cerveau de rat réalisée au laboratoire.**

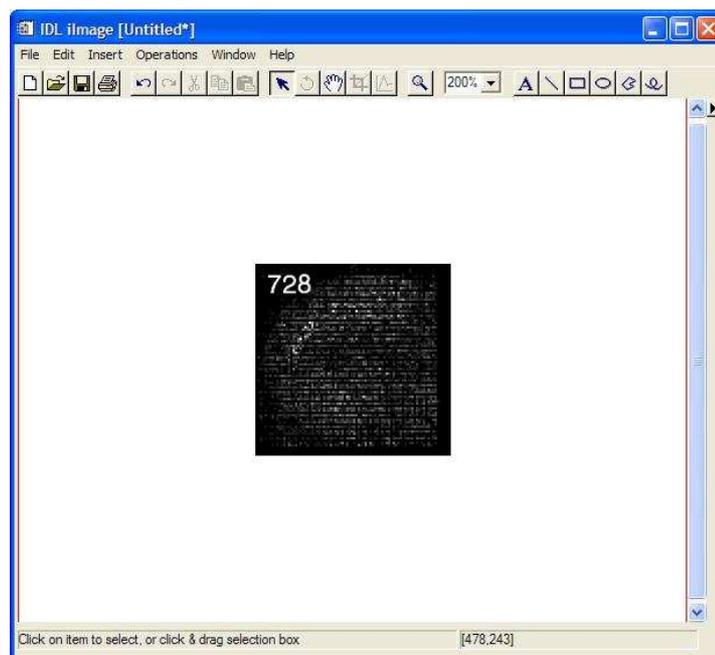
La Figure 40 montre clairement que les pics trouvés dans les trois spectres de référence ont des types de localisation différents. Le pic de rapport  $m/z$  2230 par exemple, qui ressort bien sur le spectre contenant les valeurs maximales des pics (en haut sur la figure), est très localisé, principalement sur les bords extérieurs de la coupe dans ce cas. Quant aux pics de rapports  $m/z$  2369 et 2406, apparaissant aussi sur ce spectre des maxima, ils sont situés en tellement peu d'endroits sur la coupe que seuls quelques points apparaissent sur l'image. Cependant ces pics sont visibles aussi sur le spectre résumant les valeurs moyennes des pics (à droite sur la figure) parce que, bien qu'ils soient très rares, leurs intensités sont très élevées. Un pic dont l'intensité reste modeste, mais présent sur une zone importante de la coupe, comme celui de rapport  $m/z$  2224, figurera lui aussi dans ce spectre moyen de par sa fréquence. Le pic de rapport  $m/z$  2269 est lui aussi visible sur le spectre moyen, car il est

présent sur une aire assez large de la coupe. Son intensité étant forte pour certaines localisations, il apparaît dans le spectre des maxima. La combinaison de ces deux conditions entraîne sa présence dans le troisième spectre de référence, celui calculé en additionnant tous les spectres acquis (à gauche sur la figure). Le pic de rapport  $m/z$  2219, bien que présent sur une zone un peu plus restreinte, figure lui aussi dans le spectre de la somme en raison de la forte intensité qu'il possède pour cette zone. Les pics de rapports  $m/z$  2350 et 2307 se trouvent dans les spectres des maxima et de la somme parce qu'ils bénéficient à la fois d'une intensité relativement élevée et d'une localisation assez fréquente.

Il est remarquable facilement que le spectre moyen contient le plus de pics. De par sa méthode de calcul, il va contenir tous les rapports  $m/z$  présents en quelque endroit que ce soit de la coupe. Sur la Figure 40 tous les pics pour lesquels une image a été reconstruite se trouvent dans ce spectre moyen, même si cela n'a pas été précisé précédemment.

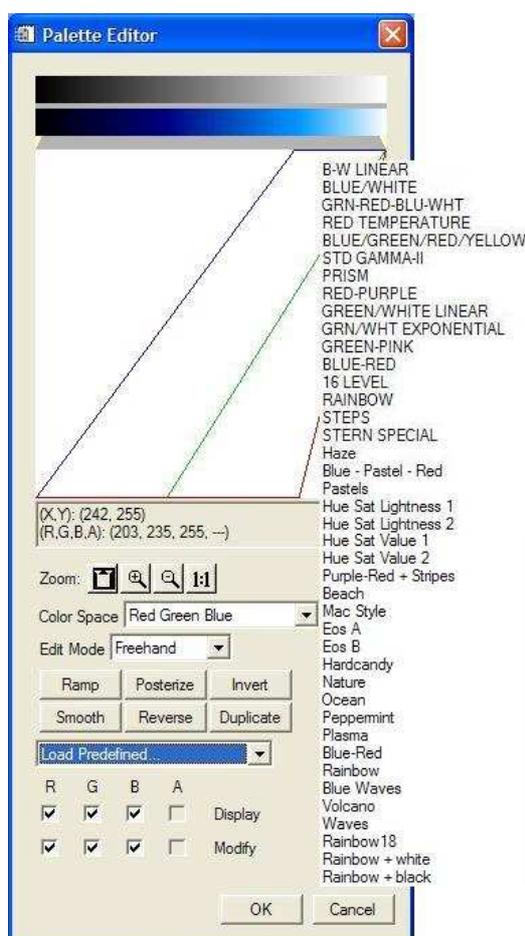
Une fois les champs nécessaires renseignés, un message d'attente apparaît en bas de la fenêtre pour indiquer où en est le traitement des données. Quand l'image est reconstruite, le message change afin de prévenir l'utilisateur qu'il peut entrer de nouvelles valeurs pour recréer une autre image. En effet, il est possible de créer plusieurs images à la suite, et de les garder à disposition pour les comparer afin d'examiner les différences de localisation de deux peptides par exemple. Si, malgré les indications affichées dans la fenêtre, les valeurs entrées par l'utilisateur amènent à une requête sans résultats, par exemple dans le cas de paramètres trop exclusifs, trop restreints, un message prévient et demande de redéfinir les données. Cette étape dure approximativement 6 minutes pour un jeu de données de 7000 fichiers environ, ou 1mn pour une image d'un peu moins de 16000 points.

Voici sur la **Figure 41** la fenêtre qui est présentée à l'utilisateur ensuite, affichant sur cet exemple une image reconstruite à partir d'une analyse sur une coupe de cerveau de rat.



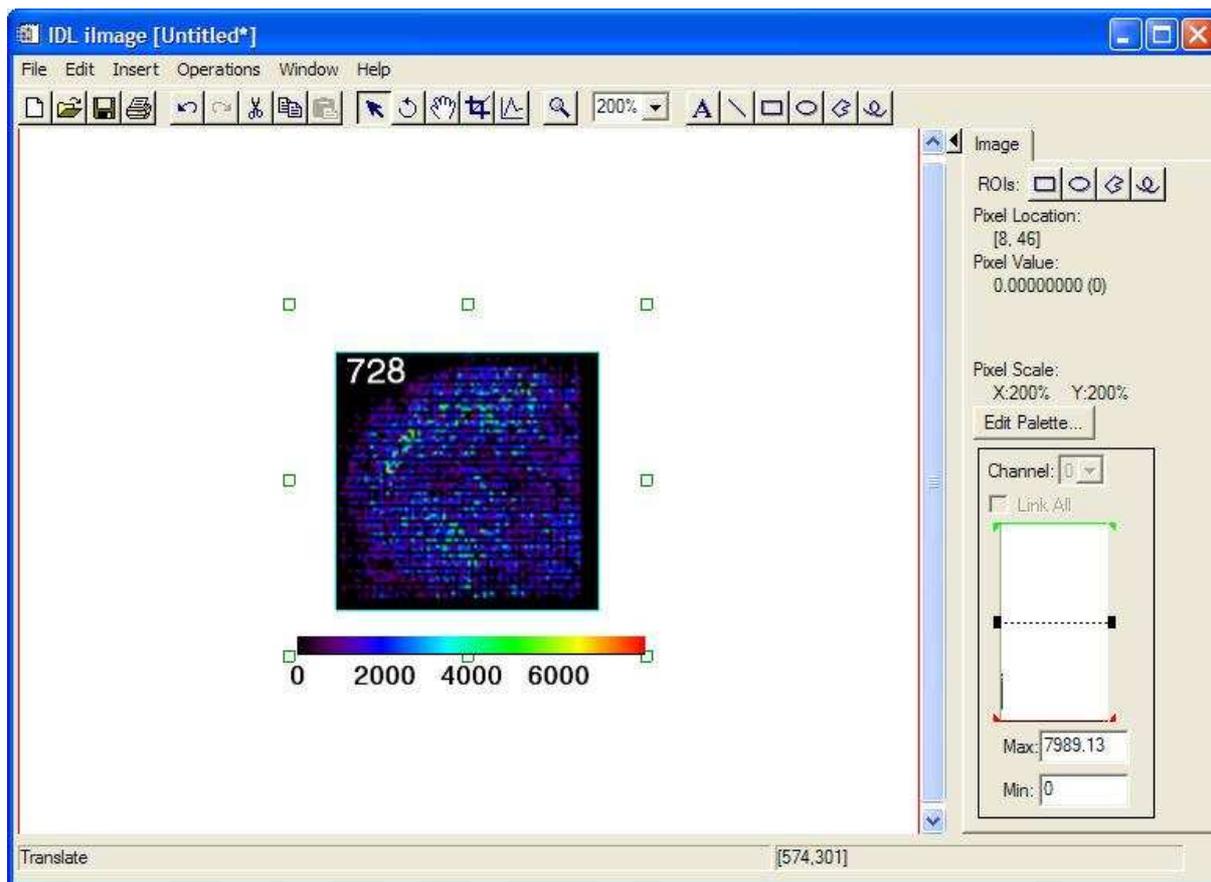
**Figure 41: Interface présentant l'image reconstruite**

Dans un premier temps l'image présentée à l'utilisateur est en noir et blanc. Cependant cette interface propose un grand nombre d'outils pour travailler l'image. Le principal est obtenu en cliquant sur « Edit palette » dans la partie droite. La Figure 42 présente une vue de l'interface apparaissant dans la fonction « Edit Palette ». Celle-ci permet de choisir parmi un large choix de couleurs celle qui mettra le mieux en valeur les résultats présentés sur l'image. L'utilisateur peut créer lui-même la gamme de coloris qu'il désire ou bien adopter l'un des dégradés dont il dispose dans la liste des « Load Predifined ». Pour l'exemple montré sur la **Figure 43**, la palette « Rainbow » a été sélectionnée.



**Figure 42: Palette des jeux de couleurs prédéfinis**

Les autres outils proposés permettent entre autres de définir une région d'intérêt, de déplacer l'image, de la redimensionner, de la tourner, ou encore de dessiner des formes géométriques dessus. Sauvegarder l'image finale, dans différents formats, fait bien sûr partie des possibilités offertes par le logiciel, ainsi que son impression, ou l'ouverture d'une nouvelle image soit vide soit déjà enregistrée préalablement. Les outils habituels (couper, copier, coller, zoom...) sont évidemment également disponibles aussi.



**Figure 43: Image reconstruite selon une échelle monochromatique**

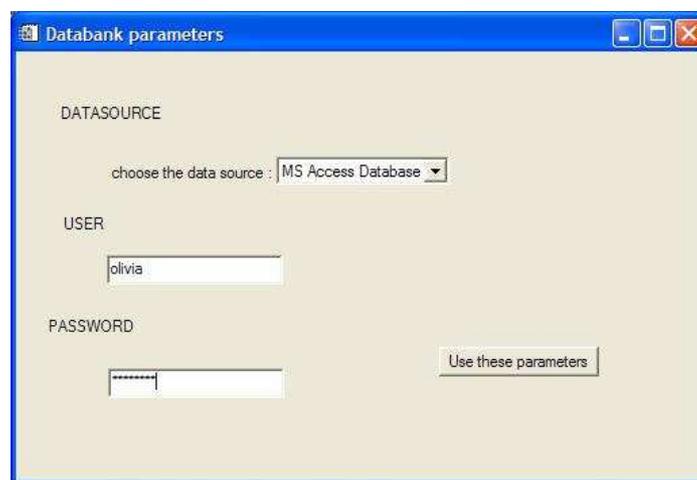
Le large panel de couleurs et de dégradés de teintes basiques disponibles, ainsi que les jeux plurichromatiques préconçus permettent de mettre en valeur les résultats par leur adaptation aux intensités relevées. L'image peut donc être retravaillée après son affichage sans nécessiter de nouveau calcul.

La toute dernière version du logiciel comprend un outil supplémentaire rendant possible, par un clic de la souris sur un point de l'image, l'affichage du spectre correspondant enregistré à ces coordonnées. L'utilisateur peut ainsi agrandir la zone du pic d'intérêt et perfectionner sa requête de reconstruction d'image en fonction des paramètres qu'il visualise sur le spectre en adaptant le seuil du bruit de fond, la marge d'erreur ou même la valeur du rapport m/z choisi de la façon la plus adéquate, ou encore en changeant la méthode de calcul. En effet, cette fonctionnalité lui offre l'opportunité de vérifier si la valeur de l'aire du pic paraît plus pertinente que celle du maximum du pic en vue de la reconstruction de l'image.

Ultérieurement les objectifs sont de rendre possible la superposition de l'image créée et de l'image réelle de la plaque MALDI avec l'échantillon ainsi que de visualiser la présence de plusieurs peptides à la fois sur une seule image grâce aux jeux de couleurs.

### 4.1.2.3 Troisième choix : Configuration des paramètres de la base de données

La dernière partie du logiciel concerne la configuration de la base de données utilisée pour charger les données. En cliquant sur le bouton « Change BD parameters » l'utilisateur ouvre une nouvelle fenêtre où il doit choisir parmi différentes options proposées (**Figure 44**). Le logiciel va chercher automatiquement sur l'ordinateur où il est installé les différentes sources de données ODBC (Open DataBase Connectivity) disponibles dans les outils d'administration du Panneau de configuration. Ces sources de données ODBC peuvent être liées à un serveur de bases de données situé sur l'ordinateur, mais aussi à un serveur de bases de données situé sur une autre machine sur le réseau. Ainsi il est possible de délocaliser la base de données sur un ordinateur différent de celui sur lequel le logiciel est installé, ou d'utiliser plusieurs bases de données placées sur des machines liées.

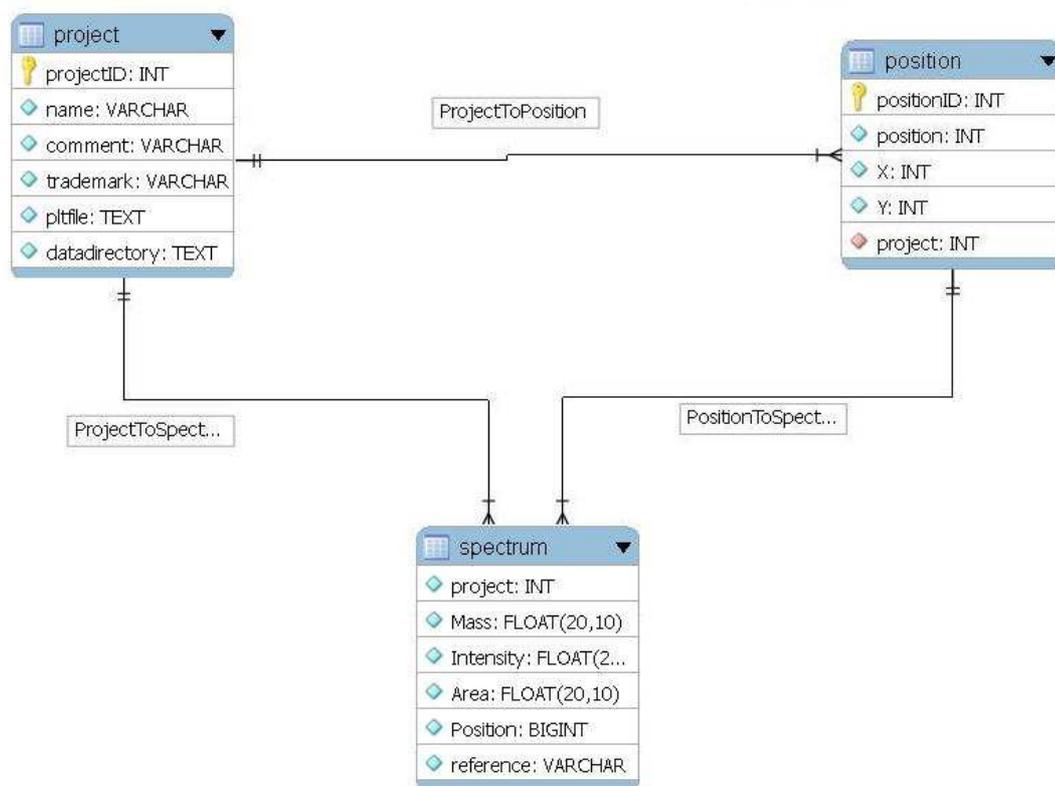


**Figure 44:** Interface permettant de configurer les paramètres de la base de données

Après avoir entré le nom de la source de données adéquate, le nom d'utilisateur adapté et le mot de passe correspondant, l'utilisateur peut cliquer sur le bouton « Use these parameters » pour appliquer cette configuration. Ainsi lors de l'ouverture d'un nouveau projet, les données des spectres récoltées seront enregistrées dans la base de données indiquée qui devra être composée de trois tables comme on peut le voir sur la **Figure 45**.

MySQL, le système de gestion de bases de données le plus populaire au monde, a été choisi dans cette application pour sa fiabilité. En effet, ce logiciel open source est très sûr. Il dispose d'un système d'autorisations très souple qui permet des accès à différents niveaux de privilèges. Par exemple, la création ou la suppression d'une base de données peuvent être limitées à certains utilisateurs ou groupes d'utilisateurs. Et les mots de passe qui circulent sur internet sont cryptés. De plus, de par l'attention portée par les développeurs à la vitesse de traitement lors de sa conception, MySQL est rapide. Il est aussi simple d'utilisation et fonctionne sur plus de 20 plateformes (Linux, Windows, Mac OS...) offrant ainsi une grande flexibilité. L'un de ses autres avantages réside dans son coût puisqu'il est gratuit dans le cadre de la licence GPL. Le grand nombre de développeurs utilisant MySQL garantit une assistance

efficace par le biais des listes de diffusion spécialisées. D'autre part il permet la création et la manipulation de bases de données de grande taille, ce qui est à prendre en considération dans le cas de notre application gérant un nombre élevé de données. Le nombre de lignes des bases de données sous MySQL peut atteindre cinquante millions. Par défaut, la taille d'une table est limitée à 4 Go, mais il est possible d'aller jusqu'à 8 Go si le système d'exploitation de l'installation sur laquelle se trouve la base de données le permet. Par ailleurs, MySQL est configurable afin de s'adapter au mieux à des besoins spécifiques grâce à la licence open source GPL qui autorise les programmeurs à le modifier. Enfin ce système de gestion de bases de données a été écrit et rigoureusement testé pour éviter toute fuite de mémoire ce qui le rend efficace et sûr.



**Figure 45: Structure de la base de données**

### 4.1.3 Installation de MITICS

Un programme d'installation est disponible. Celui-ci permet d'installer facilement le logiciel en lui-même, mais aussi le logiciel de gestion de bases de données MySQL nécessaire à l'utilisation du logiciel d'imagerie, ainsi que l'ODBC (Open DataBase Connectivity) qui les lie tous les deux. Cependant, le programme d'installation vérifie avant de l'installer si le logiciel de gestion de bases de données MySQL est déjà présent ou non. Ainsi, dans le cas où il est déjà présent sur l'ordinateur où l'utilisateur veut installer le logiciel d'imagerie, l'installeur se contente d'installer le logiciel d'imagerie proprement dit et l'ODBC qui va permettre de le relier à MySQL.

Afin de réaliser ce programme d'installation, le logiciel Inno Setup 5 a été utilisé car cet installeur entièrement gratuit crée des exécutables permettant d'installer les logiciels à façon. De plus, l'installation d'un logiciel packagé avec InnoSetup est d'une grande simplicité et assure également la prise en charge de la désinstallation. Les **Figure 46**, **Figure 47** et **Figure 48** présentent quelques unes des différentes fenêtres affichées lors de l'installation du logiciel sur l'ordinateur.



Figure 46: Première fenêtre de l'interface d'installation du logiciel

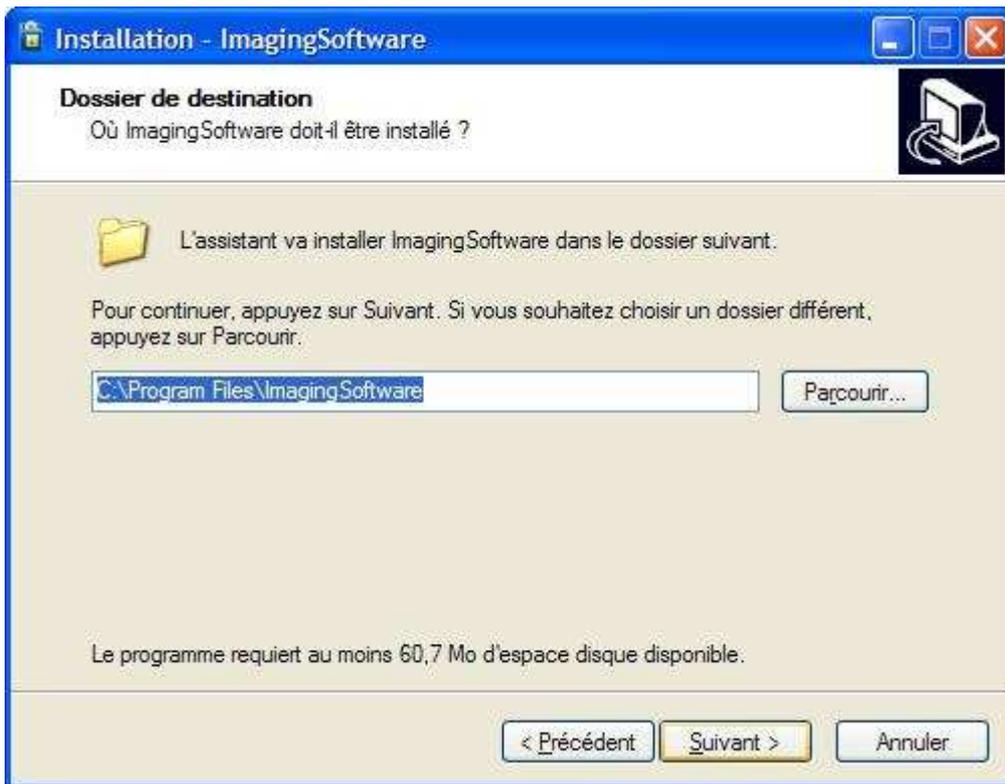


Figure 47: Choix du répertoire d'installation du logiciel

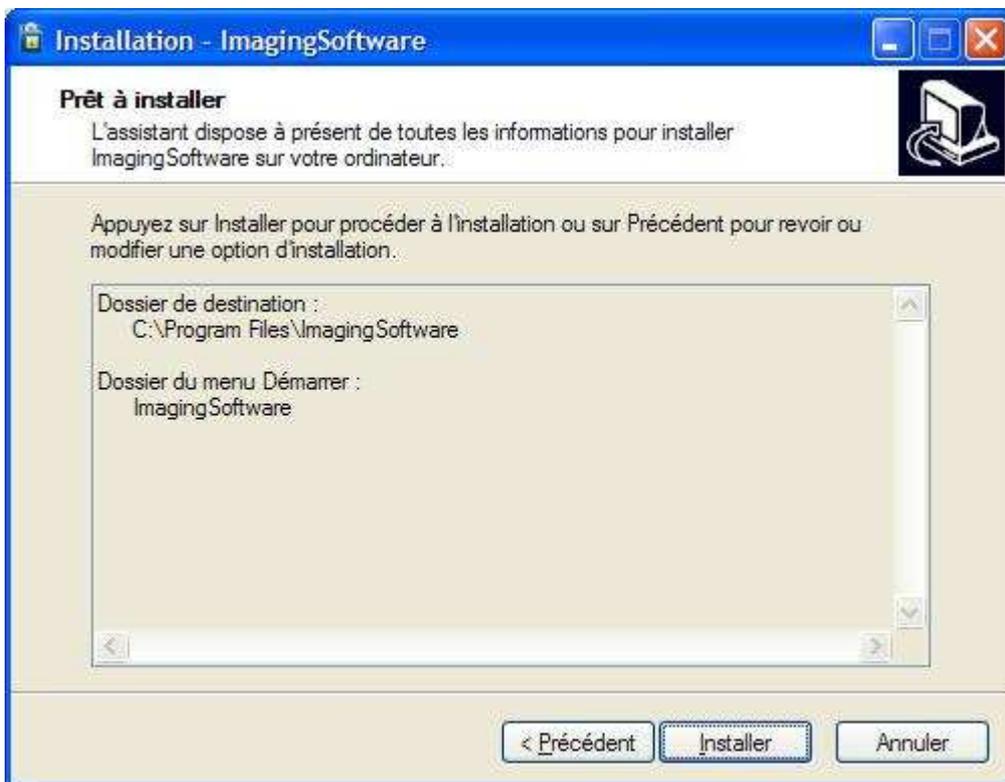
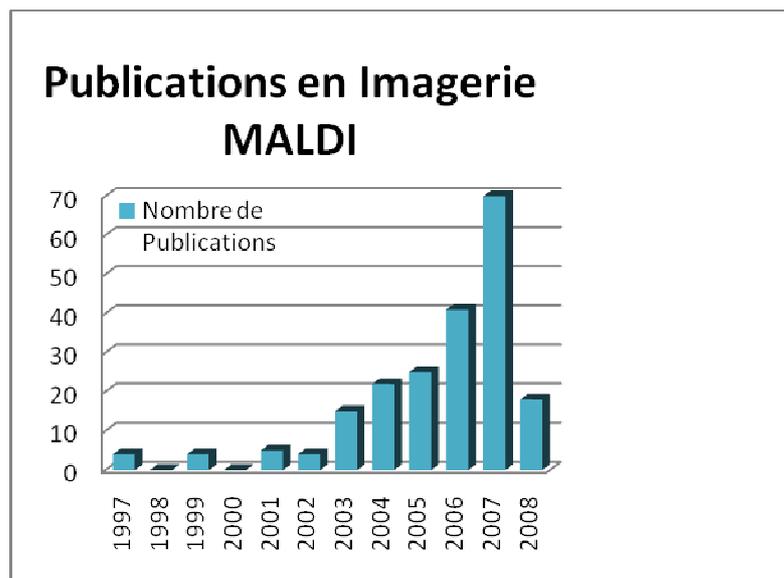


Figure 48: Bilan des choix d'installation

La réalisation d'un nouveau logiciel d'imagerie peut paraître à première vue saugrenue étant donné le fait qu'il existe déjà des logiciels commerciaux qui semblent donner de bons résultats, si on se réfère à l'explosion du nombre de publications ces 5 dernières années (Figure 49). Cette technologie est encore en développement mais devrait d'ici 5 ans entrer dans une phase d'expansion exponentielle après avoir convaincu le domaine de la clinique, de la pharmaceutique, de l'agrochimie et de l'environnement. Cependant, il semble dommage de dépendre des constructeurs pour réaliser cette technologie qui, outre la prouesse technologique des instruments, s'appuie sur un simple logiciel de conversion des données permettant la création d'images qui seront le support d'hypothèses biologiques ou de diagnostics. L'autre point fort dommageable est le côté « Black Box » des logiciels commerciaux du point de vue du traitement des données, du signal, des calculs statistiques, de la normalisation et de l'interpolation des données des logiciels commerciaux. C'est dans ce contexte qu'il nous semblait important de créer un logiciel, universel, open source et transparent. Chaque paramètre est contrôlé par l'utilisateur et l'image représente réellement ce qui est présent sur la coupe sans être passé sous un coup de « Photoshop ».



**Figure 49 : Evolution du nombre de publications en imagerie MALDI depuis son introduction en 1997 jusqu'au mois de mai 2008**

De plus étant basé sur de l'open source, il peut être accaparé par les utilisateurs pour lui introduire d'autres fonctions et ainsi l'améliorer. Publié en cette année 2008, quarantième anniversaire du symbole de libération, MITICS pourra peut-être briser les chaînes liant les chercheurs à leurs machines et à leur constructeurs.

Dans le prochain chapitre, nous allons voir les performances de MITICS et le comparer à ses concurrents.

## 4.2 Applications biologiques des développements bioinformatiques

### 4.2.1 Applications à la sangsue

La singularité de la sangsue réside dans sa faculté à régénérer sa chaîne nerveuse suite à une lésion <sup>97</sup>. L'étude de cette restauration totale et fonctionnelle du système nerveux ouvre donc des perspectives thérapeutiques fortes pour les maladies neurodégénératives ou encore dans les cas d'altération du système nerveux. Ainsi nous nous intéressons, dans le cadre d'une collaboration avec le Professeur Eduardo Macagno (UCSD, USA), aux facteurs embryonnaires se ré-exprimant chez l'animal adulte dans le cadre de la régénération et pouvant par la suite trouver des applications chez les vertébrés supérieurs. Plus précisément, la compréhension des mécanismes mis en place chez ce modèle invertébré se rapprochent fortement de l'homme tant du point de vue génétique <sup>98</sup> que du point de vue de l'organisation de son système nerveux (*i.e.* symétrie bilatérale et centralisation du système nerveux central<sup>99</sup>). Ces analogies, ainsi que l'existence d'une barrière hématoencéphalique<sup>100-102</sup> et la conservation des neuropeptides <sup>103</sup> ou encore la réponse innée très proche<sup>104-107</sup> devrait trouver des applications pour la compréhension des mécanismes de régénération/dégénération du système nerveux central chez l'homme. Le but de notre projet est donc l'identification de facteurs et de mécanismes présents chez la sangsue et inhibés chez l'homme. L'imagerie MALDI offre un outil puissant dans ce contexte en permettant de visualiser de manière dynamique l'évolution des biomolécules et notamment celles du protéome au cours du temps.

#### 4.2.1.1 Présentation du modèle sangsue *Hirudo medicinalis*

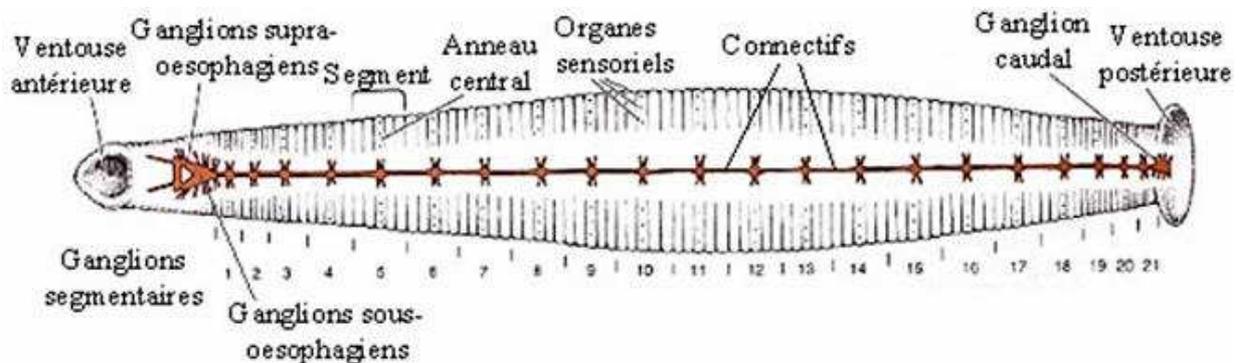
La sangsue *Hirudo medicinalis* (**Figure 50**), est un invertébré hermaphrodite vivant en eau douce. Cet ectoparasite se nourrit du sang de vertébrés comme de poissons, de reptiles ou de mammifères.

	<p><b>Règne:</b> Animalia <b>Embranchement :</b> Annelida <b>Classe :</b> Hirudinea <b>Ordre :</b> Arhynchobdellida <b>Sous-ordre :</b> Hirudiniformes <b>Famille :</b> Hirudinidae <b>Genre :</b> Hirudo <b>Espèce :</b> medicinalis</p>
-------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**Figure 50:** *Hirudo medicinalis* et sa classification phylogénétique

Dans l'embranchement des Annélides, la sangsue médicinale se situe parmi les représentants les mieux étudiés. En effet, différents groupes ont mené de nombreuses

expériences au sujet de son système nerveux, portant plus particulièrement sur le comportement, les connexions synaptiques et les fonctions des canaux ioniques <sup>108, 109</sup>. Ces études ont abouti à une connaissance très précise de la structure du système nerveux tel que schématisé sur **Figure 51**.



**Figure 51: Le système nerveux d'*Hirudo medicinalis*** <sup>110</sup>

Le choix de ce modèle repose en partie sur la simplicité de son système nerveux. En effet, le système nerveux central d'*Hirudo medicinalis* est constitué de 21 ganglions réunis par des connectifs inter-ganglionnaires, un ganglion cérébroïde et un ganglion caudal, tous deux formés de la fusion de plusieurs ganglions (**Figure 51**). 400 neurones forment chaque ganglion <sup>111</sup>. Leur architecture est très bien caractérisée morphologiquement et embryologiquement <sup>103</sup>, et demeure extrêmement conservée d'un ganglion à l'autre. Seuls les ganglions situés en cinquième et sixième positions, « les ganglions sexuels », diffèrent : ils possèdent environ 700 neurones.

La partie antérieure de l'animal est innervée par deux masses formées de quatre ganglions fusionnés. Ces deux masses sont reliées entre elles et forment un collier nerveux péri-œsophagien. Le rôle des ganglions supra-œsophagiens se limite à la neurosécrétion, tandis que les ganglions sous-œsophagiens jouent un rôle sensoriel important par l'innervation de la ventouse antérieure et des trois mâchoires. Ensuite, un ganglion segmentaire innerve chaque métamère du corps de l'animal. Quant à la ventouse postérieure de la sangsue, elle est innervée par sept ganglions fusionnés en un seul ganglion caudal volumineux. Une capsule fibreuse contenant, au niveau des connectifs, des fibres musculaires contractiles couvre l'ensemble de la chaîne nerveuse qui est ventrale, et enfermée dans un sinus sanguin.

La section d'un connectif entraîne une perturbation dans les mouvements de natation de l'animal. Cependant, il a été prouvé par de nombreuses expériences réalisées *in vivo* et *in vitro*, que la régénération est à la fois fonctionnelle et structurale car après quatre semaines la nage de l'animal redevient fonctionnelle <sup>112-114</sup>. En effet, à l'inverse des neurones mammaliens, les neurones de sangsue possèdent la faculté, suite à une lésion, de régénérer des synapses et surtout de restaurer leurs fonctions <sup>97</sup>. De plus, ils présentent le rare avantage d'être facilement accessibles <sup>115-118</sup>. La taille importante des neurones et de leur position inchangée

au cours du développement du système nerveux imposent l'embryon de sangsue comme un excellent modèle pour l'étude, au niveau moléculaire et cellulaire, des mécanismes de croissance neuronale. Des recherches sur les processus d'arborisation et les connections synaptiques des neurones de ganglion de sangsue ont généré de nombreuses données<sup>119-121</sup>. Les interactions cellule-cellule au cours du développement ont été étudiées à l'aide d'expériences de délétion facilitées par le temps de développement relativement lent (30 jours) de l'embryon de sangsue<sup>122-126, 127, 128</sup>.



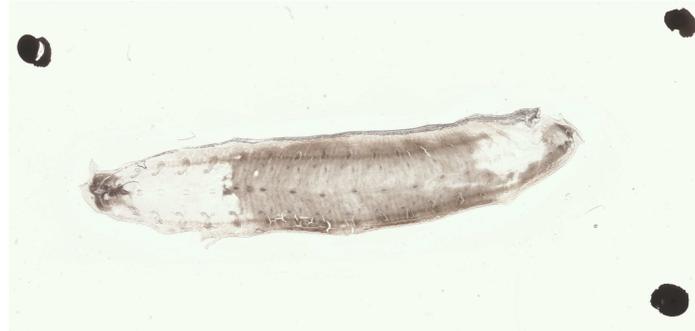
**Figure 52: Ganglions d'*Hirudo medicinalis* (Laboratoire de Neuroimmunologie des Annélides)**

Le système nerveux de sangsue semble donc idéal pour analyser les caractéristiques spatio-temporelles du développement neuronal et de la régénération nerveuse chez l'adulte. L'analyse directe par spectrométrie de masse apparaît comme l'une des techniques adéquates pour l'étude des molécules mises en jeu dans ces processus.

#### **4.2.1.2 IMS des embryons de sangsue**

En partenariat avec le Laboratoire du Professeur Eduardo Macagno, des études sont réalisées en vue de développer la technique d'imagerie MALDI à la fois sur le modèle sangsue et sur les maladies neurodégénératives, dans notre cas des animaux traités au 6-hydroxydopamine (6-OHDA) pour mimer la maladie de Parkinson. Le but de ces travaux réside dans l'identification des facteurs régulés et impliqués dans ces maladies.

Les images présentées ici ont été réalisées à partir d'embryons de sangsue au stade E12. Ils ont été ouverts le long de la ligne médiane dorsale et le vitellus a été retiré. Puis ils ont été épinglés à plat et exposés pendant environ 2 minutes au méthanol pour fixer les tissus. Enfin ils ont été placés sur les lames de verre ITO recouvertes d'une couche fine et transparente de métal (Indium Tin Oxyde) et immédiatement séchés. L'embryon a été monté de façon à ce que la surface (l'endoderme) se trouve face au faisceau laser, la région antérieure de l'animal étant positionnée à droite. La **Figure 53** ci-dessous représente l'image histologique de l'animal enregistrée avant que la matrice ne soit déposée sur les tissus.



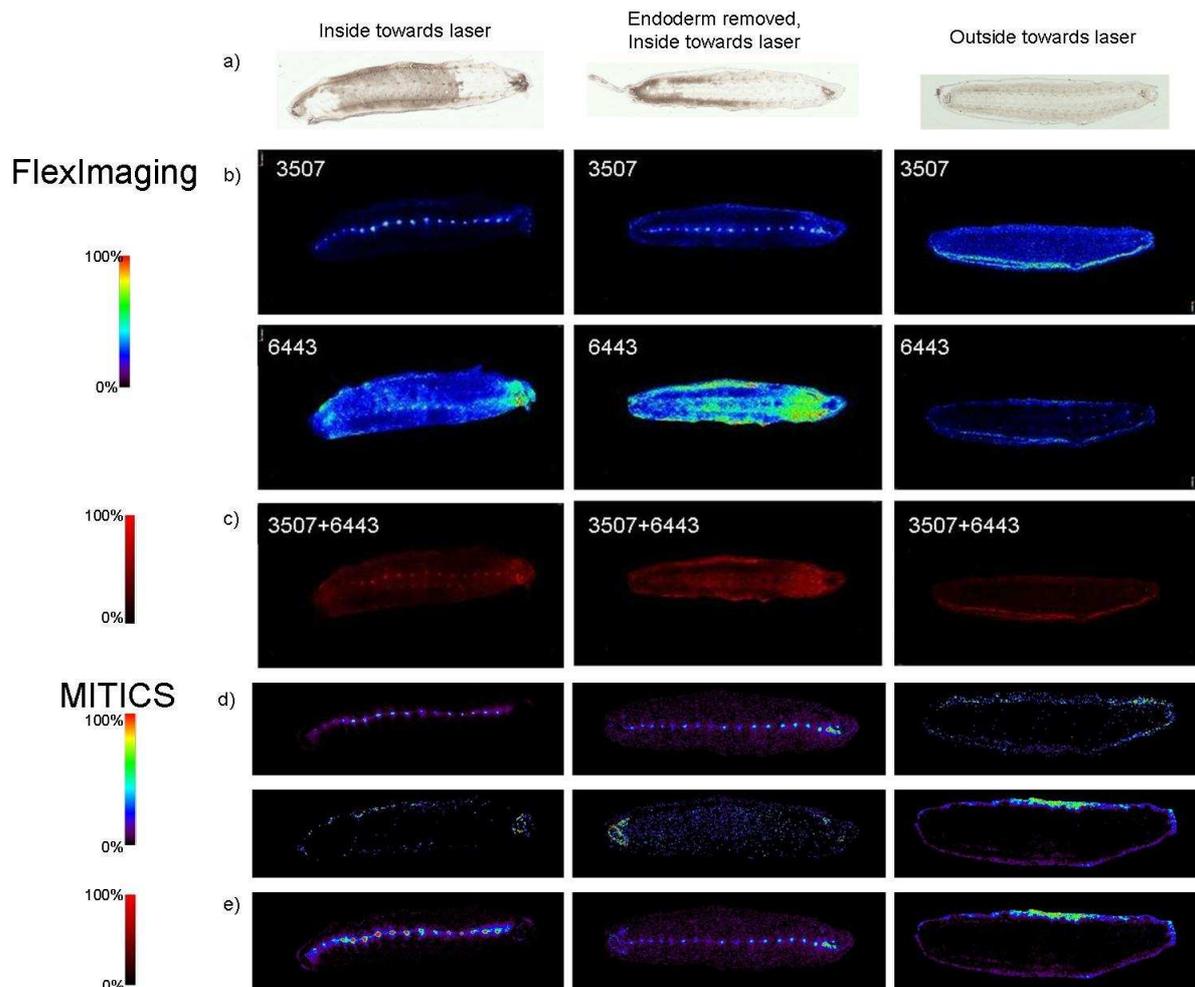
**Figure 53: Image de l'embryon avant dépôt de la matrice**

Les études réalisées sur des embryons entiers avec ou sans l'endoderme permettent des analyses du côté dorsal ou ventral. 20 000 points espacés de 60  $\mu\text{m}$  les uns des autres ont été balayés et 20 000 spectres correspondants ont été enregistrés pour reconstruire les images, définissant une zone rectangulaire couvrant entièrement les embryons. Les images réalisées montrent une répartition précise du peptidome chez l'embryon de sangsue dépendante du niveau de l'analyse (dorsal ou ventral). Certains peptides sont spécifiques à la chaîne nerveuse tant chez l'embryon que chez l'adulte, comme nous pouvons l'observer pour celui de  $m/z$  3507 (Figure 54). Sa caractérisation est en cours par analyse MS/MS directe.

En ce qui concerne, le peptide à  $m/z$  6443, celui-ci présente une localisation au niveau de la chaîne nerveuse, les ganglions antérieurs et postérieurs ainsi que les néphridies. Ce peptide présente une masse et une répartition correspondant à un peptide antimicrobien, la Lumbricine, précédemment isolé au laboratoire par le Dr. A. Tasiemski chez la sangsue adulte<sup>129</sup>. Ceci met donc en évidence la présence de ce peptide antimicrobien chez l'embryon, dans le système nerveux, très tôt lors du développement. Les travaux du groupe de A. Tasiemski viennent de montrer que chez la sangsue adulte ces peptides antimicrobiens *e.g.* la theromacine et la lumbricine, seraient également des facteurs neurotrophiques<sup>129</sup>. En effet, lors d'une lésion ces peptides antimicrobiens sont produits en grande quantité par les neurones et les cellules microgliales et permettent la repousse très rapide des neurites. L'utilisation d'anticorps spécifiques inhibe totalement cette repousse. Leur présence au sein du système nerveux chez l'embryon laisse suggérer également un rôle dans les processus de neurogénèse avec une réexpression chez l'adulte. Cette hypothèse est en cours de validation par des études d'inactivation génique chez l'embryon.

Les images reconstruites à la fois avec le logiciel MITICS et avec le logiciel FlexImaging sont présentées sur la Figure 54. Pour l'ion de  $m/z$  3507, les deux logiciels donnent des résultats similaires. Les différences visibles entre les images reconstruites par FlexImaging et celles reconstruites par MITICS sont probablement dues à l'utilisation d'échelles de couleurs légèrement différentes et à la normalisation appliquée par FlexImaging. Ce traitement des données permet d'obtenir des images plus séduisantes mais son

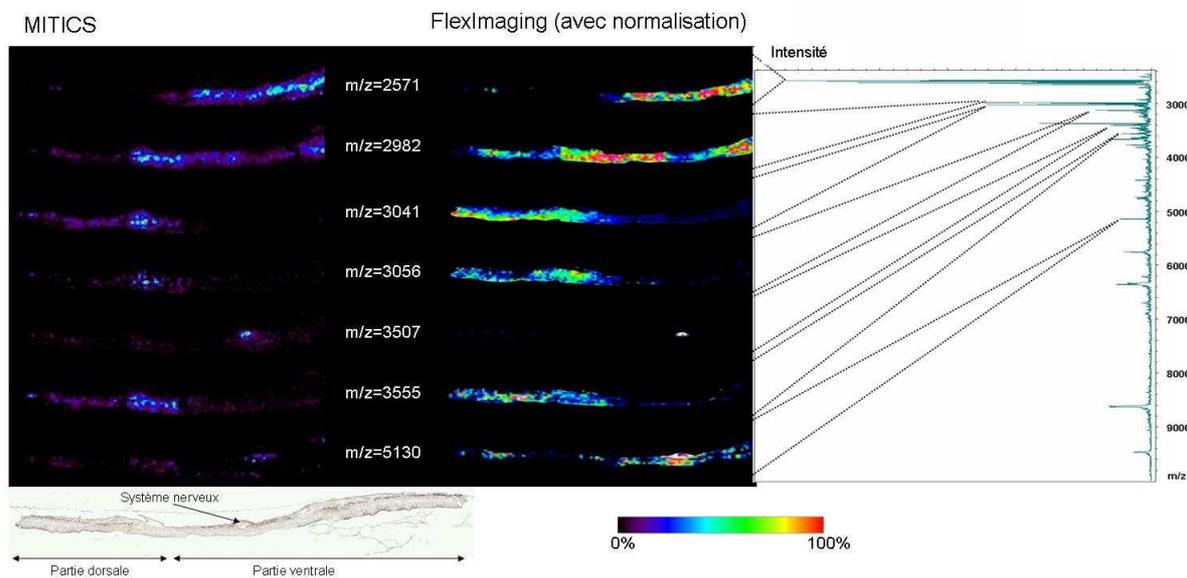
fonctionnement reste très mal connu et rend son utilisation hasardeuse. Pour l'ion de  $m/z$  6443, les différences sont plus notables. En effet, le signal de cet ion est assez faible et sa localisation difficilement observable dans les néphridies de l'animal. Cela est un peu plus clair sur l'image moléculaire reconstruite à l'aide de FlexImaging grâce à la normalisation des données. Ici cette normalisation trouve un intérêt en rehaussant le signal faible mais elle a tendance à augmenter l'intensité du bruit de fond sur les images.



**Figure 54 : Images d'embryon de sangsue reconstruites pour  $m/z=3507$  et  $6443$  avec FlexImaging (b et c) puis avec MITICS (d et e) dans trois conditions différentes**

L'étude de la répartition dorso-ventrale du peptidome chez l'adulte au niveau d'un métamère confirme la présence du peptide  $m/z$  3507 strictement au sein du système nerveux (Figure 55). Les peptides  $m/z$  3041, 3056 et 3555 ont une répartition totalement dorsale alors que celui de  $m/z$  2571 est ventral, tandis que ceux de  $m/z$  2982 et 5130 sont aussi bien dorsaux que ventraux. Ces résultats reflètent bien une localisation spécifique du protéome dépendamment de l'axe dorso-ventral au niveau de chaque métamère. Le travail désormais est

de pouvoir attribuer à ces masses et ces cartes de répartitions la caractérisation structurale de ces peptides. C'est dans cette optique qu'a été entreprise la caractérisation du transcriptome du système nerveux par le séquençage d'une banque d'EST à l'annotation de laquelle j'ai participé (Voir chapitre suivant).



**Figure 55 : Images moléculaires reconstruites à partir d'une tranche d'un métamère d'une sangsue adulte**

Bien qu'utilisant la même échelle d'intensité, les couleurs affichées par MITICS apparaissent moins intenses qu'avec FlexImaging en raison de la normalisation appliquée par ce dernier. Cependant, même si les intensités paraissent plus faibles avec MITICS, les résultats au niveau de la localisation restent équivalents.

## 4.2.2 Imagerie dans le cadre de la maladie de Parkinson

Dans un but de compréhension des mécanismes et facteurs impliqués dans la régénération nerveuse chez la sangsue adulte, nous avons étudié par comparaison la maladie de Parkinson chez un modèle animal, le rat traité au 6-OHDA.

Des études protéomiques à partir de blocs fixés et paraffinés de cerveau de rat traité au 6-OHDA ont été réalisées dans l'unité précédemment<sup>31, 77</sup>. A partir des protéines identifiées comme marqueurs de la pathologie, j'ai réalisé la cartographie de la répartition des fragments de digestion trypsique *in silico* de ces protéines. Dans ce contexte, je me suis basée sur une liste de 9 protéines identifiées comme biomarqueurs et j'ai recherché dans les spectres MALDI de digestion trypsique sur coupe si je pouvais retrouver les ions identifiés lors de la digestion *in silico* (Table 1). Sur la base de ces ions caractéristiques de chacune des protéines, j'ai entrepris la réalisation des images. L'idée sous-jacente était de voir si chacun des ions constituant la protéine avait une localisation spécifique et si la somme de ces ions pouvait définitivement donner la réelle répartition de la protéine marqueur. La première protéine a été la Collapsin Response Mediator Protein 2 (CRMP2), considérée comme un nouveau marqueur de la pathologie<sup>31, 77</sup>. Les localisations des ions fragments de cette protéine sont parfaitement corrélées, et leur superposition (Figure 56) donne une répartition au niveau du corps calleux, siège des démences dans le cadre des maladies neurodégénératives<sup>31, 77</sup>. CRMP2 est normalement localisée, comme le montrent les résultats d'hybridation *in situ* (Figure 56), au niveau des cellules pyramidales de l'hippocampe et des cellules de purkinje<sup>130</sup>. Sa localisation spécifique traduit donc un rôle dans la pathologie. De plus, CRMP2 n'est normalement pas exprimée chez l'adulte mais uniquement au cours de la neurogénèse et permet l'orientation des axones lors de la pousse neuritique. Sa réexpression chez l'adulte est donc liée à la pathologie<sup>130</sup>. Les neurones reçoivent des signaux de repousse, mais anarchiquement au sein du corps calleux participant aux troubles neurologiques.

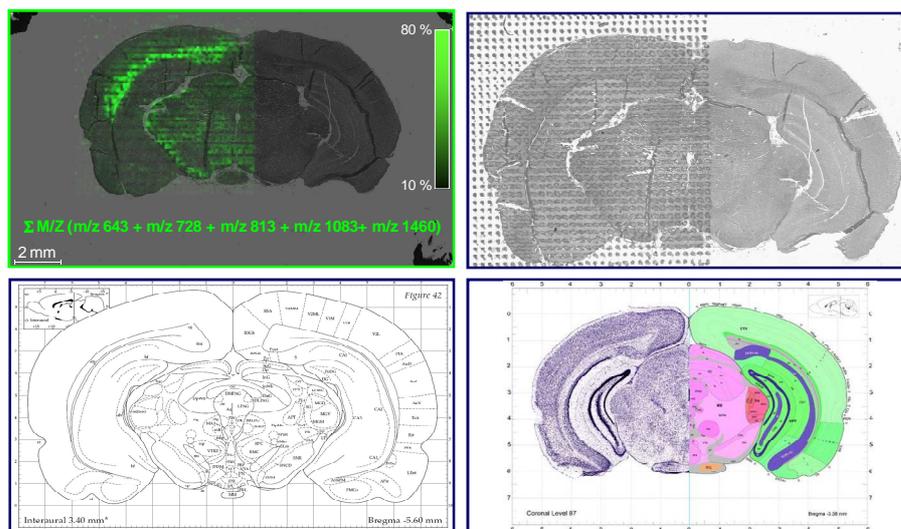
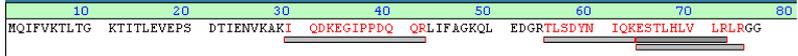
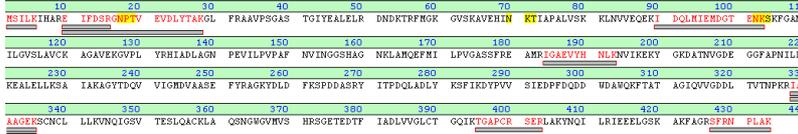
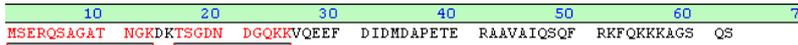
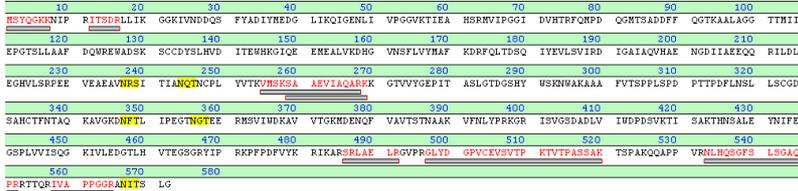
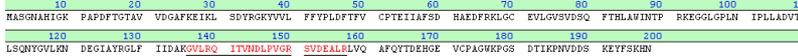
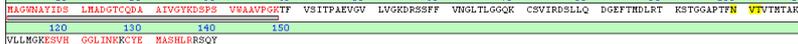
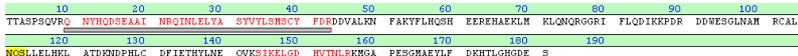


Figure 56 : répartition des ions trypsiques de CRMP2 (somme des m/z 643, 728, 813, 1083 et 1460), scan de la coupe avant et après dépôt de la trypsine, image du plan de coupe selon l'index Bregma et image de la répartition des ARNm par hybridation *in situ* selon le Allan Brain Atlas (<http://www.brain-map.org>)

Nom	Couverture de séquence	Masse (Da)	Peptides retrouvés
Ubiquitin	43.4%	8 559.617	1067.621 ESTLHLVLR 1081.556 TLSDYNIQK 1336.670 ESTLHLVLRRLR 1523.840 IQDKEGIPPDQQR  
Alpha Enolase	16.8%	47 098.245	591.080 MSILK 766.460 EIFDSR 786.473 IAKAAGEK 932.532 SFRNPLAK 976.490 TGAPCRSER 1143.644 IGAEVYHNLK 1636.854 IDQLMIEMDG TENK 2153.164 EIFDSRGNPTVEVDLYTAK  
PEP 19	37.1%	6 803.290	1049.526 TSGDNDGQKK 1336.670 MSERQSAGATNGK  
CRMP2	15.0%	62 238.591	591.080 ITSDR 766.460 IVAPPGGR 841.517 MSYQGKK 844.477 SRLAELR 1143.644 SAAEVIAQARK 1460.791 VMSKSAAEVIAQAR 2169.166 NLHQSGFSLSGAQIDDNIPR 2406.335 GLYDGPVCEVSVTPKTVTPASSAK  
Peroxidoxin	11.1%	21 770.064	788.474 SVDEALR 1636.854 GVLRQITVNDLPVGR  
Profilin	40.7%	14 977.521	1052.621 ESVHGGLINK 1109.700 CYEMASHLR 3959.791 MAGWNAIDSLMADGTCQDAAIVGYKDSPSVWAAVPGK  
Ferritin Heavy Chain	26.0%	20 982.271	1481.856 SIKELGDHVTNLR 4105.412 QNYHQDSEAAINRQINLELYASYVYLSMSCYFDR  

Nom	Couverture de séquence	Masse (Da)	Peptides retrouvés
Apolipoprotein E	25.3%	35 731.331	726.233 LSTHLR 888.441 GVSAIRER 968.532 LGPLVEQGR 1162.139 MEEQTQQIR 1183.796 LAKEVQAAQAR 1238.700 TANLGAGAAQPLR 1505.863 GWFEPLVEDMQR 1523.840 QRTANLGAGAAQPLR
HSP27	20.5%	22 807.573	591.080 QTADR 932.532 QTADRWR 988.549 RVPFLLR 1358.716 AQIGGPESEQSGAK 1481.856 EGVVEITGKHEER

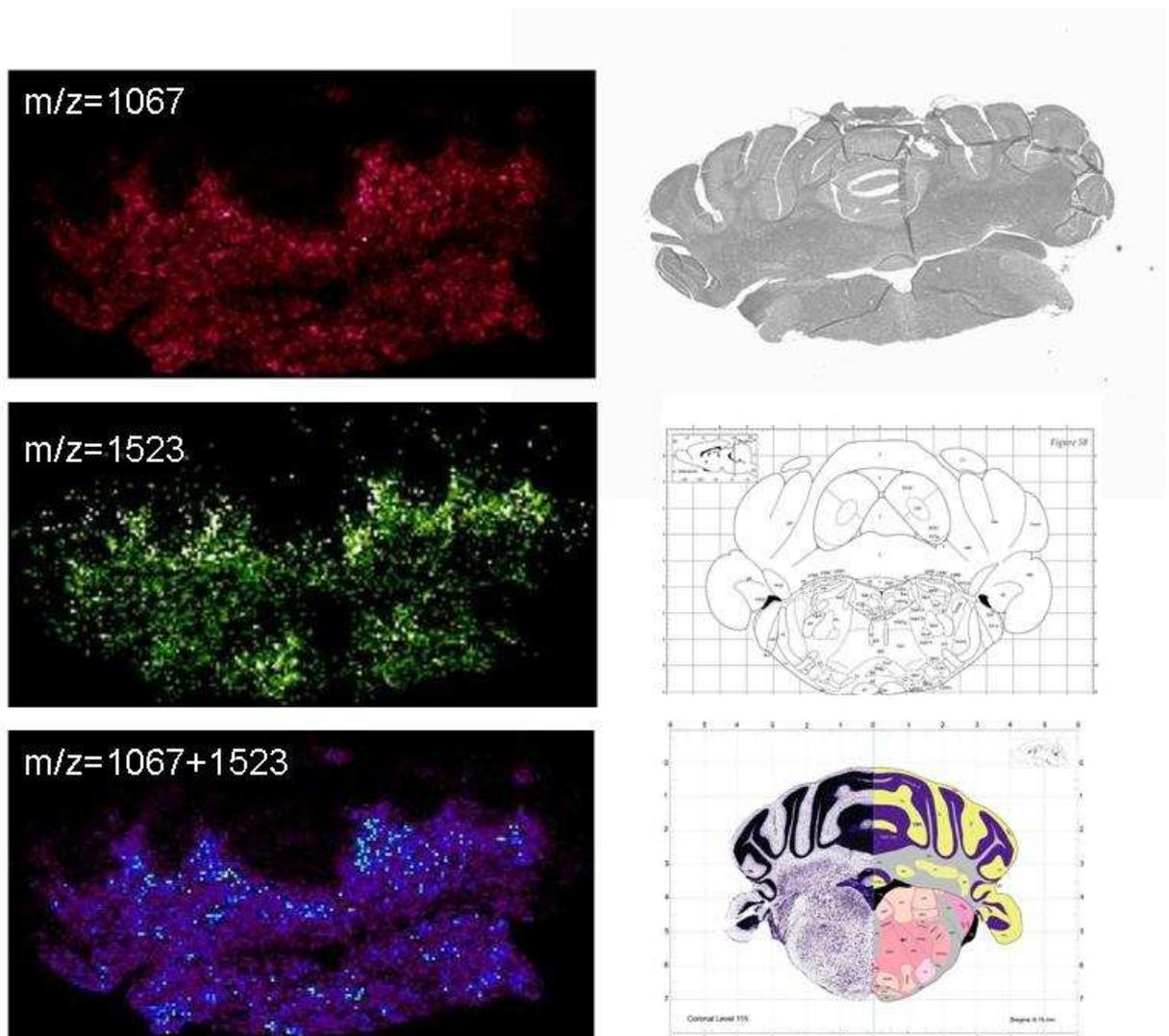
**Tableau 2: Différents biomarqueurs identifiés par Imagerie MALDI dans le cadre de l'analyse de rats injectés 6-OHDA et par comparaison avec des rats témoins, ainsi que leur séquence et les peptides issus de la digestion enzymatique par la trypsine in silico**

Puis nous avons réalisé des images de la répartition d'autres biomarqueurs : l'ubiquitine, l'apolipoprotéine E et la vimentine (**Figure 57-61**).

L'ubiquitine intervient au niveau du protéasome. Elle est surexprimée dans le cadre des maladies neurodégénératives<sup>131</sup>. Quant à l'apolipoprotéine E, bien qu'un certain nombre de controverses existent, nos résultats confortent les nouvelles données cliniques à ce sujet qui la placent comme marqueur de la pathologie<sup>132, 133</sup>. L'ubiquitine et l'apolipoprotéine E s'accumulent dans la substance grise lors de la maladie de Parkinson alors que, dans le cas de la maladie d'Alzheimer, ces molécules ont une localisation corticale<sup>134</sup>. Les images moléculaires reconstruites confirment ces résultats : les peptides formant ces protéines se retrouvent dans les mêmes types cellulaires (**Figure 57-59-60**).

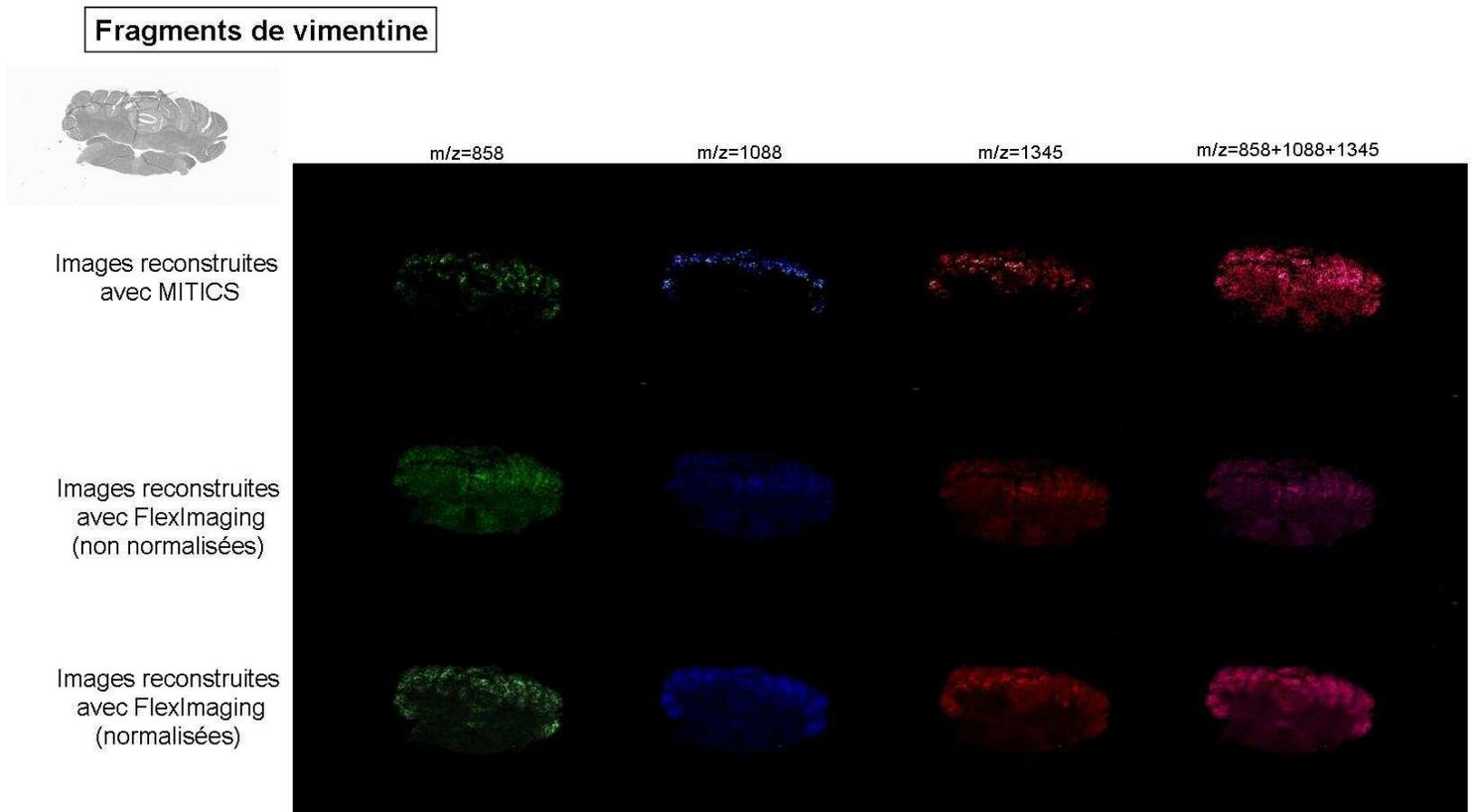
La comparaison des images reconstruites à celles de l'expression de l'ubiquitine chez des animaux contrôles, par exemple, montre que l'expression est corticale chez les témoins, alors que chez les animaux traités, celle-ci est à l'opposé (**Figure 57**). La figure 57 représente les images reconstruites pour les ions de m/z 1067 et 1523, fragments d'ubiquitine, et montre que ces deux fragments sont bien présents à la même localisation dans le bulbe rachidien (medulla oblongata), tandis qu'aucun signal n'est retrouvé dans la partie du cervelet (cerebellum) de la coupe en comparaison avec la photographie de la coupe prise avant l'expérience où la morphologie est bien apparente. L'image composite de la somme de ces deux fragments visible sur la dernière vignette de la figure 57. La possibilité de sommer ainsi

différents ions correspondants à une même molécule mène à une image plus contrastée avec une intensité de signal plus élevée et moins de variations entre les points. L'image moléculaire pour l'ion de m/z 1523 présente une localisation plus diffuse que celle de l'ion 1067. Ceci n'est pas dû au traitement par le logiciel, mais probablement à la présence d'un second pic plus petit sous l'enveloppe du pic correspondant au fragment de digestion de l'ubiquitine et ne présentant pas exactement la même distribution que notre fragment. Cet effet est dissipé par la visualisation de l'image composite, qui montre une fois de plus l'intérêt d'utiliser des images basées sur le signal de plusieurs ions.



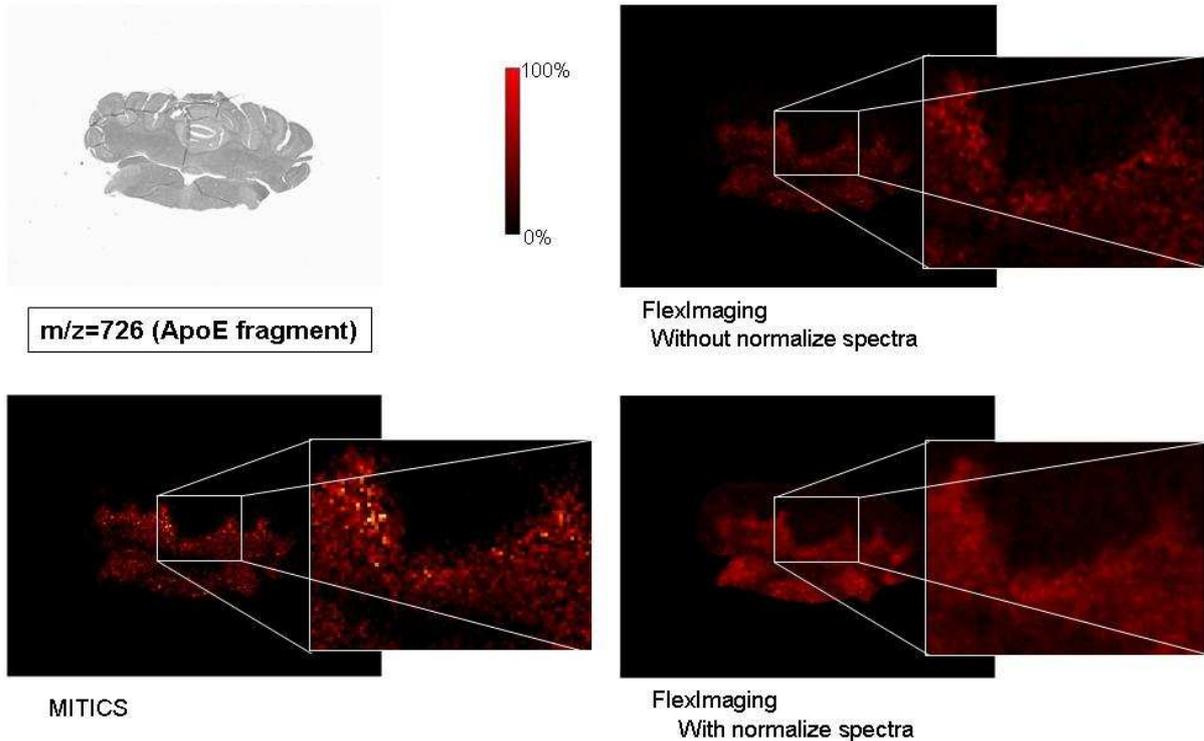
**Figure 57 : Images de la répartition des ions de l'ubiquitine Sont inclus : le scan de la coupe, l'image du plan de coupe selon l'index Bregma, image de la répartition des ARNm par hybridation in situ selon Allen Brain Atlas (<http://www.brain-map.org>)**

Dans un deuxième temps, j'ai comparé les images réalisées par MITICS et FlexImaging en ce qui concerne la répartition des ions qui composent la vimentine, qui est également un marqueur de la pathologie <sup>134</sup> (**Figure 58**). La comparaison des deux logiciels (MITICS et FlexImaging) montre la même répartition de ces ions mais une meilleure définition des images dans le cas de MITICS par rapport à FlexImaging sans normalisation. Dans le cas de l'utilisation de la normalisation avec FlexImaging, les résultats sont équivalents à ceux obtenus avec MITICS.



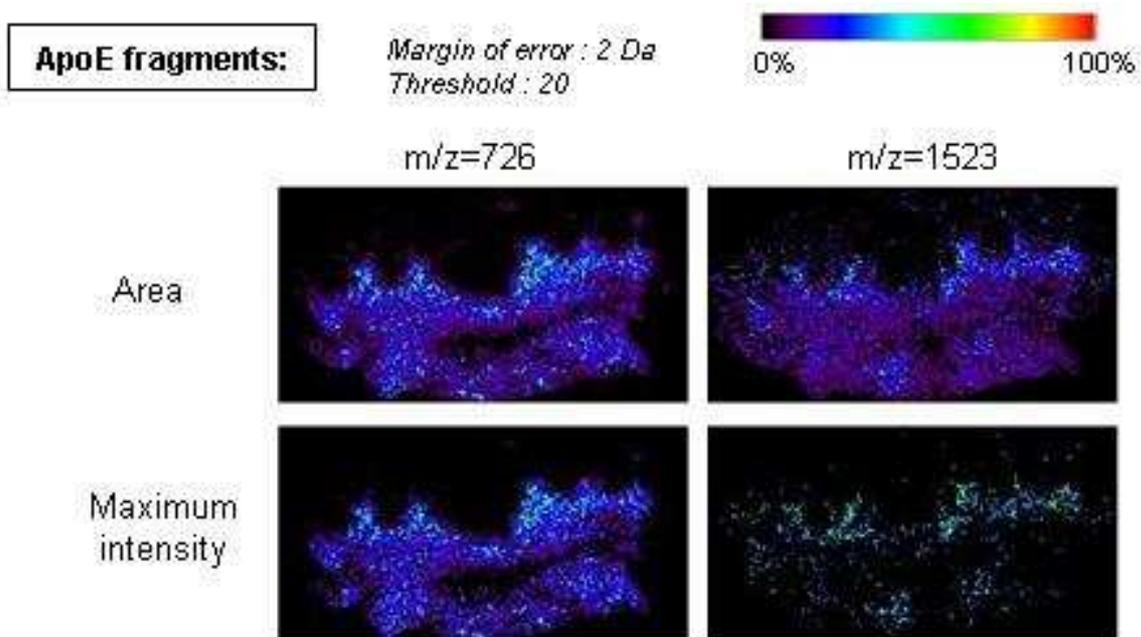
**Figure 58 : Comparaison de la répartition des ions de la vimentine entre Fleximaging avec ou sans normalisation et MITICS et scan de la coupe.**

La comparaison des deux logiciels a été poursuivie par une analyse plus fine grâce à un grossissement d'une zone de même aire en prenant comme exemple la répartition d'un ion provenant de l'apolipoprotéine E, également marqueur de la pathologie (**Figure 59**). L'agrandissement montre que l'analyse statistique et le traitement de données réalisés par FlexImaging conduisent à une image moins pixélisée que celle de MITICS, mais, par là même, moins définie pour donner un aspect plus lissé de l'image. Certes MITICS donne une image plus pixellisée puisque sans interpolation et traitement de l'image, mais cette image est très nette avec une grande sensibilité.



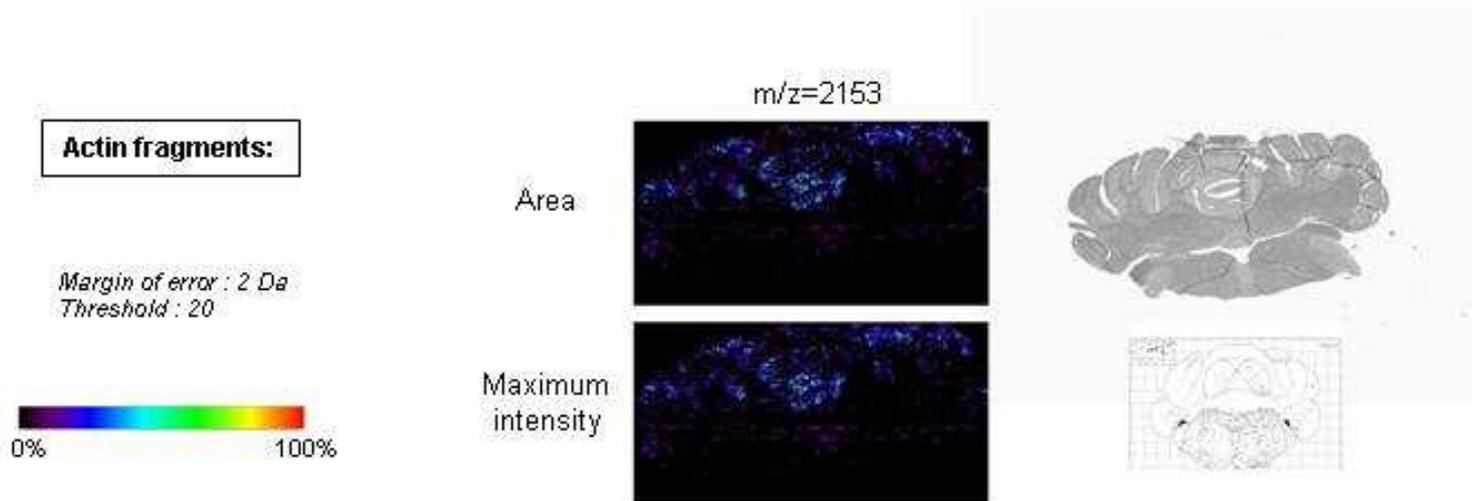
**Figure 59 : Comparaison de la répartition d'un ion de l'apolipoprotéine E entre Fleximaging avec ou sans normalisation et MITICS. Sont inclus le scan de la coupe et des grossissements des aires de répartition du peptide**

L'analyse de ces images montre que les logiciels commerciaux apportent une solution mais ils se basent uniquement sur l'intensité maximale du pic. Grâce à MITICS, il est possible de réaliser des images basées soit sur l'intensité maximale du pic d'intérêt, soit sur l'aire du pic d'intérêt (**Figure 60**). Les résultats montrent une réelle différence dans le cadre de la réalisation d'une image par l'aire du pic. Des images ont été reconstruites sur les ions de m/z 726 et 1523, fragments de digestion de l'Apolipoprotéine E. L'ion de m/z 1523 n'est pas détecté de la même façon en fonction de la méthode de reconstruction de l'image. L'image la plus correcte est bien par l'aire du pic car cette méthode de calcul permet une meilleure cohérence avec les autres images reconstruites pour la même analyse mais pour des peptides de m/z différents. En effet, la reconstruction d'une image à partir de l'aire d'un pic pallie la baisse de résolution qui peut parfois apparaître avec l'augmentation du rapport masse-surcharge comme cela a été discuté dans le chapitre 4.1. Pour le pic de rapport m/z 726, la différence entre l'image reconstruite à l'aide du maximum du pic et celle reconstruite à l'aide de l'aire du pic est moindre car la résolution du spectre est meilleure dans les masses plus basses : les pics sont plus intenses et moins larges, donc les valeurs du maximum et de l'aire sont moins éloignées l'une de l'autre.



**Figure 60 :** Images moléculaires avec MITICS de la répartition de deux ions de l'apolipoprotéine E en prenant en compte soit l'aire, soit l'intensité maximale du pic

Par ailleurs, nous avons remarqué que les ions provenant de l'actine présentent une répartition différente voire antinomique de celles des biomarqueurs précédemment cartographiés, comme on peut le voir sur la **Figure 61**.



**Figure 61 :** Images moléculaires avec MITICS de la répartition d'un ion de l'actine en prenant en compte soit l'aire, soit l'intensité maximale du pic

L'ensemble de ces données met en évidence le fait que l'imagerie MALDI dépend du traitement bioinformatique effectué pour reconstruire les images. Cependant, notre exemple montre bien qu'il faut être très prudent lors de l'utilisation d'un logiciel dont on ne connaît pas les méthodes d'interpolation ou de traitement statistique du signal. En effet, les divergences visibles sur les images reconstruites précédemment témoignent de l'importance des différents logiciels utilisés, ainsi que leurs méthodes de calcul, leurs traitements informatiques et statistiques ou encore les paramètres définis lors de la création de l'image. Les cartes de localisation obtenues restent fortement dépendantes de toutes ces caractéristiques ; en effet, de nombreux traitements internes mal connus peuvent induire un biais et l'interprétation biologique de l'image résultante qui en découle peut ainsi s'avérer fautive. Il apparaît également clairement que la mesure de l'aire du pic semble mieux convenir aux données MALDI-TOF et en particulier pour les plus hautes masses comme attendu par la dégradation de la résolution avec les rapports  $m/z$  croissants sur les instruments TOF. Ceci souligne également que le traitement des données doit être ajusté selon l'analyseur équipant l'instrument.

## 4.3 La Banque EST de Sangsue

### 4.3.1 Présentation du projet

Dans le cadre du Consortium de génomique de la sangsue médicinale, une collaboration internationale a été mise en place entre le Laboratoire de Neuroimmunologie des Annélides, l'University of Iowa Center for Bioinformatics and Computational Biology, le Robert H. Lurie Comprehensive Cancer Center of Northwestern University du Professeur Bento Soares, l'International School for Advanced Studies of Trieste et l'University of California, San Diego. Le but de ce projet est de trouver la corrélation entre la génétique de la sangsue médicinale et ses fonctions biologiques.

A l'heure actuelle, peu de données moléculaires sont disponibles chez les Lophotrochozoaires comme les Mollusques, ou les Annélides, comparativement à celles obtenues chez les Ecdysozoaires tels que la Drosophile, *C. elegans* ou les Anophèles. Les études phylogénétiques réalisées montrent que les Lophotrochozoaires sont plus proches des vertébrés, du point de vue de la structuration de leur génome (introns), que des Ecdysozoaires qui semblent avoir évolué différemment suite à une pression évolutive plus forte. Cette annotation constitue donc un travail réellement fondamental en vue d'obtenir des modèles invertébrés plus proches de l'homme pour des études sur les pathologies comme les maladies neurodégénératives.

L'amorce de ce projet d'annotation de cent mille clones d'ESTs de ce lophotrochozoaire réside dans le séquençage des ESTs du système nerveux de la sangsue réalisé par le centre national de séquençage du Génoscope d'Evry, ainsi que dans la construction d'une banque normalisée d'ESTs enrichie en transcrits à partir d'embryons de sangsue. Les séquences proviennent de deux sources : le Génoscope et l'Université d'Iowa.

Un clustering (classement) préalable des données a été effectué par l'équipe du Professeur Terry Gaasterland (USCD, USA) pour éliminer les séquences redondantes et optimiser la fiabilité des séquences comme expliqué dans le paragraphe suivant.

### 4.3.2 Le clustering des ESTs

Le clustering des ESTs a été réalisé par l'équipe de Terry Gaasterland du Laboratory for Computational Genomics afin de réduire le nombre de séquences, d'éviter les redondances et d'obtenir des séquences plus longues. Il a été effectué à l'aide du programme Phrap (<http://www.phrap.org/>).

Phrap fonctionne avec phred et consed. C'est un ensemble de logiciels utilisé pour le traitement des données de séquençage de génome en vue d'un assemblage.

### 4.3.2.1 Phred

Phred est un programme capable de lire les chromatogrammes issus des séquenceurs d'ADN et de les transformer en séquences nucléotidiques. Il génère en sortie un fichier contenant la séquence d'ADN correspondant au chromatogramme et la valeur de la qualité associée à cette transformation. Ces fichiers ont le plus souvent une extension «.phr».

### 4.3.2.2 Phrap

Phrap (Phragment Assembly Program) est un programme qui permet l'assemblage d'un génome à partir des résultats du phred. Il prend en entrée les fichiers «.phr» générés par le programme phred et génère un fichier «.ace». Il est couplé avec d'autres programmes tels que phr2fasta, qui convertit les fichiers «.phr» au format fasta pris en charge par le programme phred, et cross\_match et swat qui masquent les séquences appartenant aux vecteurs et permettent ainsi d'éviter les erreurs d'interprétation.

### 4.3.2.3 Consed

Consed est un programme qui permet l'édition et la visualisation de l'assemblage effectué par le programme phrap. Il prend en entrée le fichier «.ace» généré par le programme phrap et génère à son tour la séquence unique et finale du génome. Cette étape est suivie par l'annotation de la séquence obtenue c'est à dire la recherche de CDS, de protéines, de domaines, de répétitions, d'ARNr, d'ARNt, .... Cette étape est extrêmement variable selon le type d'organisme étudié (eucaryotes, procaryotes) et selon les objectifs des annotateurs.

Malheureusement le premier classement réalisé par l'équipe de Terry Gaasterland n'a pas été convaincant. Lors de l'alignement de séquences entre elles, quand le logiciel Phrap a détecté une ressemblance entre le début d'une séquence et la fin d'une autre, il a correctement regroupé les deux en un « contig » commençant par la première séquence et se finissant par les bases de la seconde. Cependant, dans le cas où la correspondance entre les deux séquences se situait au début des deux séquences, comme schématisé sur la Figure 62, le programme les a bien réunies dans le même « cluster » mais en inversant le sens de l'une des séquences pour créer un « contig » où la partie commune aux deux séquences se retrouvait au milieu de la séquence finale.

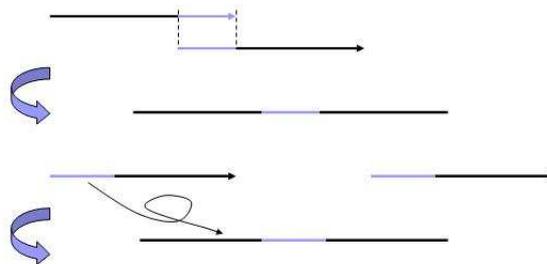


Figure 62 : Représentation schématique du fonctionnement du logiciel Phred

### 4.3.3 Annotation

En l'espace de quelques années, la pléthore de nouvelles données biologiques produites par les grands projets génomes et, par la suite, le développement d'applications informatiques pour l'analyse des données ont influencé tous les aspects de la recherche en biologie.

Les ordinateurs assurent désormais la collecte et la gestion des données dans les laboratoires. La puissance des bases de données permet un accès plus aisé aux données stockées sous un format informatique, ce qui facilite grandement la détection d'une correspondance entre deux séquences. Il y a 30 ans environ, rechercher des similarités entre deux séquences d'ADN prenait un temps considérable car les comparer signifiait les aligner à la main l'une au dessous de l'autre avec un logiciel de traitement de texte et chercher des identités caractère par caractère. Ce type de recherche qui valait une publication de cinq pages est maintenant devenu routinier. En effet, à la fin des années 80, des programmes informatiques rapides ont été conçus pour comparer des séquences d'ADN et ont bouleversé le domaine de la biologie moléculaire en en devenant l'un des outils informatiques les plus utiles. Actuellement, un biologiste peut trouver en quelques secondes des dizaines de correspondances entre des séquences en utilisant des programmes d'alignement de séquences tels que BLAST et FASTA. Ces programmes couramment employés permettent ainsi de développer une hypothèse sur la fonction d'un gène nouvellement découvert ou d'identifier des gènes en comparant une séquence d'ADN inconnue à la totalité des séquences d'une banque de données publique. Pour améliorer encore la méthode, certains groupes ont mis en place des techniques et des outils pour faciliter, accélérer la recherche d'une séquence dans une base de données. Par exemple le partitionnement de la base de données fouillée en sous-groupes de taille inférieure autorise à ne parcourir qu'une petite partie de cette base de données<sup>135</sup>.

#### 4.3.3.1 Vue d'ensemble

L'annotation des ESTs de sangsue a été réalisée à l'UCSD en partenariat avec le bioinformaticien Richard Rouse. Les bases de données utilisées dans ce projet ont été mises en place à l'aide de MySQL puis reliées au site web à l'aide de codes écrits en langage Perl et PHP. Les alignements nécessaires ont été exécutés grâce aux programmes BLAST et CLUSTALW installés en local sur les serveurs du Laboratoire du Professeur Macagno.

Dans le cadre de ce projet, j'ai réalisé un stage dans le laboratoire du Professeur Macagno à l'Université de San Diego et ainsi participé à l'annotation de la banque de données de transcrits de sangsue grâce à un travail en commun avec Richard Rouse. Tout d'abord, les 100 000 séquences ont été alignées avec celles de la banque de données Uniprot disponible sur internet, à l'aide du logiciel BLAST. Cela a permis d'annoter environ 70000 séquences. Ensuite nous avons classé les ESTs (afin de supprimer les redondances et pour optimiser la fiabilité des séquences) puis nous les avons à nouveau comparés aux séquences d'Uniprot. Enfin un BLAST supplémentaire a été exécuté pour rechercher les homologies entre les

transcrits de sangsue et les séquences de la banque de données Gene Ontology disponible elle aussi sur internet. Le choix de cette banque de données est dû à sa gestion intelligente des termes et synonymes biologiques. A partir de cette banque et grâce à ces annotations, il a été possible d'identifier de très nombreux gènes et surtout d'obtenir la représentation la plus complète possible du transcriptome de la sangsue. Cependant de nouvelles annotations ont été effectuées depuis, et ont remplacé les données obtenues.

Parmi les cent mille clones d'EST qui ont été séquencés et clusterisés, 40 413 contigs ont ainsi été obtenus. Ces contigs ont été comparés à différentes banques de données (NCBI NR Database, EBI SwissProt, TrEMBL database) par utilisation du logiciel BLAST et ont généré 2 450 986 annotations.

A l'heure actuelle, sur la base de ces données plusieurs biopuces à ADN gènes sont en cours de réalisation au laboratoire (FRE CNRS 2933) en collaboration avec le laboratoire des biopuces du Pr. Y. Lemoine (UMR CNRS 8117). Par exemple, une analyse phylogénique couplée aux résultats obtenus d'après l'annotation a conduit à l'extraction de la banque d'ESTs de sangsue d'un transcrit d'innexine dont la séquence n'était pas encore connue entièrement. Après une RACE PCR, il a été possible de reconstruire sa séquence complète et l'arbre phylogénique lié.

#### 4.3.3.2 Leechmaster

Afin d'accéder aux séquences dans la base de données d'ESTs de sangsue, un site internet a été créé : leechmaster.ucsd.edu. Le HTML (HyperText Markup Language) utilisé pour réaliser ce site est un langage de balisage et de description de document. Avec l'HTTP (HyperText Transfer Protocol) et les adresses web, il constitue l'une des trois inventions à la base du World Wide Web. Ce langage codant les pages web permet à l'aide de balises l'intégration de scripts en PHP grâce auxquels l'accès à la base de données est géré par une CGI (Common Gateway Interface). PHP (officiellement, ce sigle est un acronyme récursif pour *PHP: Hypertext Preprocessor*) est un langage de scripts Open Source, spécialement conçu pour le développement d'applications web et principalement de pages web dynamiques via un serveur HTTP. Sa syntaxe est empruntée aux langages C, Java et Perl. Une des grandes forces de PHP réside dans le fait qu'il supporte de nombreuses bases de données et rend extrêmement simple l'écriture d'une page web exploitant une base de données.

Plutôt que de créer une nouvelle base de données il a été décidé d'utiliser BioSQL et bioseqdb (<http://cvs.open-bio.org/cgi-bin/viewcvs/viewcvs.cgi/?cvsroot=biosql>), qui est un schéma de base de données créé justement dans le but de stocker des séquences. Ce projet a été initié par the Open Bioinformatics Foundation sous le nom OBDA ([obda.open-bio.org](http://obda.open-bio.org)) pour standardiser les méthodes permettant aux projets d'accéder aux ressources de données de séquences.

Lors du stage réalisé à l'université de Californie de San Diego, j'ai effectué deux annotations par comparaison aux bases de données vues ci-dessus et leurs résultats ont été mis

en ligne sur internet pour en permettre la consultation par les biologistes des laboratoires partenaires (Figure 63).

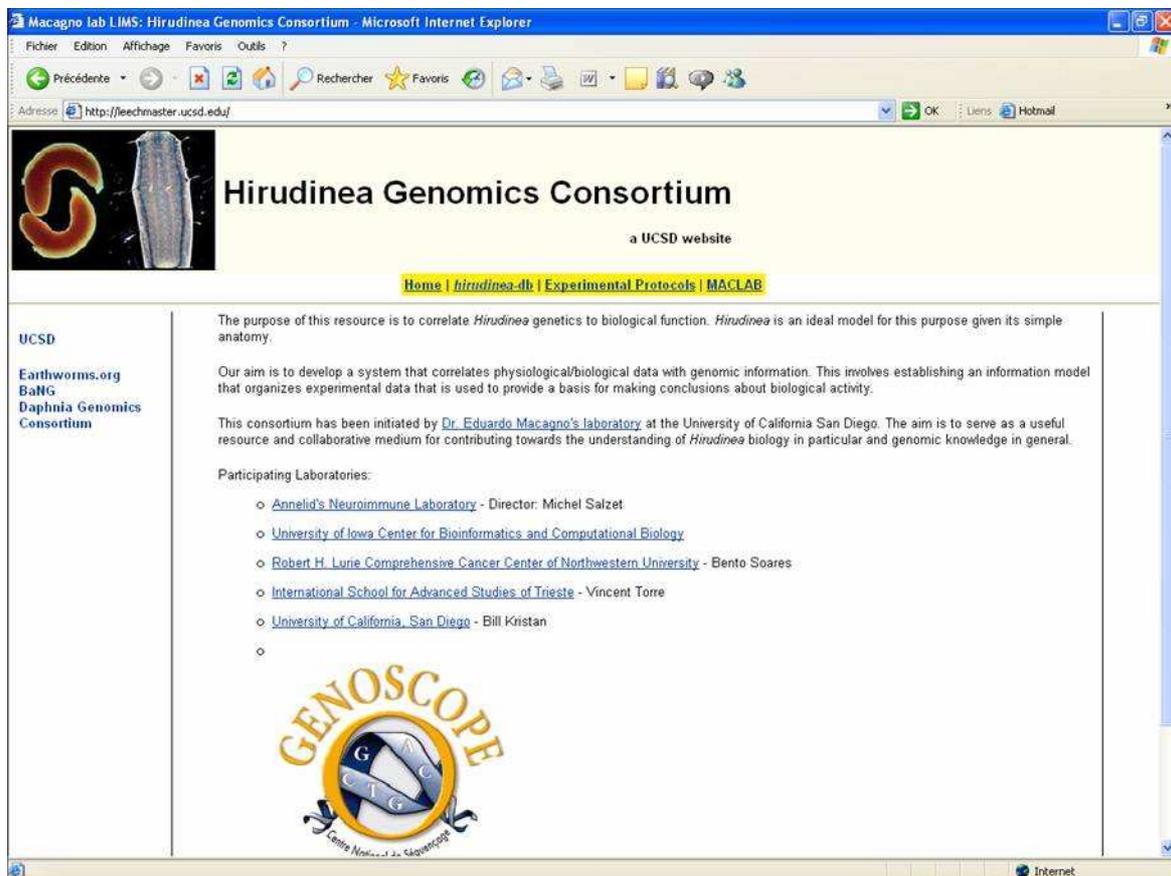


Figure 63: Page d'accueil du site leechmaster.ucsd.edu

Mon travail lors du stage à San Diego a consisté à mettre en place ces annotations, par la création de fichiers adéquats contenant les séquences pour qu'ils soient acceptés par le programme blast tournant en local sur le serveur du laboratoire du Professeur Macagno. Puis la comparaison aux bases de données a été effectuée en lignes de commandes sous linux. Enfin nous avons parsé les fichiers résultats de format xml grâce à un programme en Perl pour en récupérer les données et les charger dans la base de données. Une fois ce travail effectué, les résultats ont été affichés sur le site internet mis en place à l'aide du langage php permettant le lien avec les bases de données MySQL et donc la consultation de l'annotation.

Grace à cette banque annotée, il est désormais possible de la traduire en protéine et de réaliser une digestion tryptique *in silico* afin de pouvoir l'interroger lors des analyses par spectrométrie de masse en utilisant les logiciels Mascot, Phenyx, InsPecT en vue de corréler imagerie MALDI et caractérisation structurale des peptides et protéines d'intérêt.

### Conclusion

Les protéines sont des éléments biologiques essentiels d'une grande diversité en raison des mécanismes comme l'épissage alternatif et les modifications post-traductionnelles. La protéomique qui décrit leur analyse se révèle donc d'une importance majeure, d'où son essor ces 10 dernières années. Parmi les outils disponibles en vue de la détection de peptides et de protéines se trouve la spectrométrie de masse, devenue incontournable grâce aux récents développements qu'elle a connus. Il en résulte des applications promettant de belles perspectives comme l'imagerie par spectrométrie de masse. En effet, en plus de la détection et de la caractérisation des peptides et protéines, l'imagerie permet leur localisation sur tissus et peut ainsi contribuer à approfondir les connaissances des processus biologiques en comprenant la dynamique du protéome lié à ces processus.

L'intérêt de l'élaboration d'un nouveau logiciel pour l'acquisition et la reconstruction d'image résidait dans l'absence d'un programme libre et universel en ce qui concerne les fabricants des appareils, mais aussi dans la volonté de mieux connaître le traitement du signal et des processus de normalisation. Par conséquent mon travail durant cette thèse a consisté à écrire un logiciel indépendant des formats propriétaires des fichiers utilisés par les sociétés mettant au point les spectromètres de masse. Afin de se libérer au maximum des contraintes liées aux marques, le paramétrage de l'acquisition des données peut être effectué à l'aide de la première partie.

Ce nouvel outil possède la caractéristique supplémentaire d'offrir à l'utilisateur un choix entre deux méthodes pour la sélection des signaux à considérer. Ainsi il est possible de recourir à la valeur maximale du pic correspondant au peptide étudié pour reconstruire l'image des zones de l'échantillon où il est présent, mais l'utilisateur dispose aussi d'une autre alternative représentée par l'usage de l'aire du pic. Ces éléments sont loin d'être anodins car ils définissent les paramètres en vue de la création de l'image et de ce que le biologiste pourra en retirer pour son interprétation des mécanismes qu'il étudie. L'application de ce logiciel aux études sur le modèle animal Parkinson a permis de mettre en évidence, une distribution différente des ions en fonction de l'aire ou de l'intensité maximale du pic.

Pour renseigner plus précisément l'utilisateur sur les données, le logiciel présente trois graphiques de référence résumant la totalité des spectres du projet. Chacun compile les données de façon différente pour mieux détailler la répartition des pics. Ainsi la première montre les valeurs maximales des intensités trouvées dans les spectres, la deuxième affiche la somme de toutes les intensités et la troisième la moyenne des intensités. La plupart des

logiciels existant ne fournissent qu'un seul spectre moyen dont la méthode de conception n'est pas connue, tandis que dans le cas de MITICS, les calculs effectués sont transparents.

Les applications biologiques sur la sangsue et le rat traité au 6-OHDA ont montré la force de la technologie et l'intérêt de l'imagerie moléculaire. Nous avons pu mettre en évidence des répartitions spécifiques de certains peptides au niveau du système nerveux de l'embryon et de la sangsue adulte, comme le peptide antimicrobien la lumbricine. Ce peptide exerce chez l'adulte une fonction de neurotrophine et est présent chez l'embryon où il participe aux processus de neurogénèse. La comparaison avec la maladie de Parkinson a permis de constater l'expression très forte d'une molécule de la famille de la sémaphorine au sein du corps calleux dans le cadre de la pathologie. Or cette molécule est connue pour s'exprimer normalement spécifiquement chez l'embryon de sangsue, mais non chez l'adulte. L'équipe a récemment mis en évidence dans les EST de système nerveux de sangsue adulte, tout comme dans les EST d'embryon de sangsue, la présence de ce facteur.

Ces applications biologiques ont également fait partie intégrante de ma thèse et m'ont permis lors de mon stage à l'USCD de participer à l'annotation des EST de la sangsue médicale. Ces EST sont désormais le lien direct entre l'imagerie MALDI et l'identification des molécules participant au protéome chez cet animal. L'identification moléculaire des peptides détectés chez l'embryon comme chez l'adulte permettra, lors de la réalisation des images moléculaires, de caractériser les facteurs impliqués dans le processus de régénération nerveuse chez la sangsue adulte. La stratégie réalisée dans le cadre de l'étude de la maladie de Parkinson *i.e.* digestion trypsique de l'ensemble des protéines sur coupes, analyses par nanoLC-ion trappe, caractérisation automatique des peptides digérés, identification des protéines marqueurs après interrogation des banques Mascot, Phenyx, puis réalisation d'images moléculaires des ions provenant de la digestion trypsique des protéines d'intérêt, conduira quant à elle, à l'identification de biomarqueurs impliqués dans la dégénérescence neuronale. La comparaison entre les résultats de ces deux études sera d'un intérêt majeur en vue de la compréhension des phénomènes de dégénérescence et de régénération du système nerveux.

## Perspectives

Les perspectives qui se profilent pour MITICS se classent en plusieurs catégories. Tout d'abord, le programme pourrait être amélioré grâce à des modifications mineures telles que la possibilité de se déplacer dans les fenêtres à l'aide de la touche tabulation ou encore la création d'une aide. Ensuite le travail du chercheur serait facilité par quelques évolutions plus perfectionnées comme la visualisation de plusieurs pics différents à l'aide de l'utilisation de couleurs distinctes sur une même image. Une superposition de l'image recréée avec une photographie de l'échantillon sur la plaque est envisageable aussi afin de gagner en efficacité grâce à une meilleure corrélation entre les images histologiques et moléculaires.

Enfin au vu de la progression régulière de l'imagerie par spectrométrie de masse, il n'est plus exagéré d'imaginer pouvoir reconstruire, à l'aide de coupes successives, des images non plus planes mais en trois dimensions qui permettraient de visualiser des molécules au sein d'organes ou de petits animaux entiers<sup>136, 137</sup>.

L'analyse statistique en chaque point d'une coupe liée à l'aire des pics de chaque molécule d'intérêt permettra de reconstruire des images de répartition statistique sur la présence d'une molécule en un endroit précis de la coupe. Cette image moléculaire statistique est importante si on veut comparer plusieurs coupes les unes aux autres. En effet, la concentration d'un même peptide, d'une coupe à une autre, est différente, de même le plan de coupe est différent. Si on veut réaliser des analyses différentielles entre deux animaux à des états physiologiques différents, il est indispensable de pouvoir comparer la répartition statistique sur l'ensemble de coupe. Ceci constituera également par la suite un pas vers la quantification directe des peptides sur coupes, qui est un enjeu très fort pour le devenir de cette technologie.

Néanmoins la modification majeure qui amplifierait fortement le potentiel de ce programme concerne la caractérisation des peptides au sein des tissus. En effet la capacité de les identifier automatiquement ouvrirait de nombreuses possibilités pour des projets de détection de marqueurs dans le domaine de la biologie et en particulier pour l'étude de certaines pathologies ou la recherche d'un diagnostic plus précoce.

A coté des développements chimiques réalisés au laboratoire pour simplifier le spectre de fragmentation, il est nécessaire de relier les données de spectrométrie de masse aux données génomiques, afin de traduire les listes de pics détectés à la liste des peptides putatifs issus de la banque EST traduite et digérée *in silico* par la trypsine. Dans cette optique, une plateforme informatique est mise en place en partenariat entre le laboratoire, avec le Laboratory for Computational Genomics du Professeur Terry Gaasterland et le laboratoire de Vineet Bafna (UCSD) à San Diego. L'idée est de développer un système qui organise les données expérimentales et les conjugue à l'information génomique pour servir de base aux conclusions sur l'activité biologique. Cet outil permettra de rattacher la base de données d'ESTs de sangsue annotées au logiciel afin de faciliter la reconnaissance de molécules lors d'analyses sur des embryons ou des chaînes nerveuses de sangsue.

Les perspectives à long terme résident dans le lien entre l'imagerie MALDI et l'imagerie tomographique par émission de positons (PET), l'instrumentation de type X-ray CT instrumentation et l'imagerie par Résonance Magnétique (IRM) dans le cadre d'études préclinique et clinique. La complémentarité des techniques non invasives et moléculaires obtenues par imagerie MALDI permettra une meilleure précision pour un meilleur diagnostic.

On peut penser qu'à court terme le diagnostic moléculaire pourra être réalisé par imagerie par spectrométrie de masse à partir de cœlioscopie ou de frottis couplé et que le pronostic sera réalisé à l'aide de biopuces sur tissu en utilisant des aptamères, anticorps, lectines taggés<sup>94</sup>. L'imagerie MALDI deviendra une technologie de routine au niveau clinique et un outil important en complément pour le pathologiste afin d'offrir du diagnostic histo-pathologique moléculaire.

---

### Bilan

#### Publications

---

- **Jardin-Mathé O**, Bonnel D., Franck J., Elayed M., Wisztorski M., Macagno E.R., Fournier I., Salzet M. MALDI IMAGING TEAM IMAGING COMPUTING SYSTEM (MITICS): Application to Biomarkers found in FFPE tissue from 6OHDA treated rats. *J. Proteomics*
- Macagno E.R, Soares B., Scheetz T., Gasterland T, **Jardin-Mathé O.**, Salzet M. Large-scale gene discovery in *Hirudo medicinalis*, a key model organism for studies of the Nervous System (en preparation)
- Wisztorski M., Lemaire R., Stauber J., **Jardin-Mathé O**, Day R., Salzet M., Fournier I. New developments in MALDI imaging for Pathological proteomic studies. *Curr Pharm Des.* 2007;13(32):3317-24
- Wisztorski M., Stauber J., Lemaire R., Ait Menguelet S, **Jardin-Mathé O.**, Salzet M., Fournier I. Imagerie MALDI : Une nouvelle technologie pour découvrir et valider de nouveaux biomarqueurs. *Med. Sci.* 23 Spec No 1:31-6

#### Brevet

---

- **Jardin-Mathé O.**, Wisztorski M., Salzet M., Fournier I. MALDI IMAGING TEAM IMAGING COMPUTING SYSTEM (MITICS). Enveloppe Solleau (2008) INPI N° 316023 010408

#### Participation à des congrès

---

- J. Franck, .M; El-Ayed, J. Stauber, M. Wisztorski, **O. Jardin-Mathé**, I. Fournier; M. Salzet, Séquençage de novo sur Tissus par Dérivations N-terminales: vers l'Identification in situ des Protéines, SMAP 2007 : 24èmes Journées Françaises de Spectrométrie de Masse-Congrès de la Société Française d'Electrophorèse et d'Analyse Protéomique, Pau, France, 17-20, Septembre, 2007
- **O. Jardin-Mathé**, M. Wisztorski, R. Lemaire, J. Stauber, M. Salzet, I. Fournier, Développement d'un Logiciel de Reconstruction d'images en Vue d'une Application à l'Imagerie par Spectrométrie de Masse, 23<sup>èmes</sup> Journées Française de Spectrométrie de Masse, Nantes, 11-14 Septembre, 2006
- M. Wisztorski, **O. Jardin-Mathé**, V. Thomy, R. Lemaire, P. Ducoroy, J.B. Hendra, C. Courcelle, M. Salzet, I. Fournier, L'imagerie par Spectrométrie de Masse MALDI-TOF: Innovations et Perspectives. 1<sup>st</sup> Symposium of Analytical Chemistry and Biology: From molecule to proteome, September 26 -29, Montpellier, France, 2005

- Fournier I. ; Lemaire, R. ; Wisztorski, M. ; Stauber, J. ; **Jardin-Mathé, O.** ; Day, R. ; Salzet, M. « New insights in sample preparation for MALDI imaging and new developments to approach specific MALDI imaging of the transcriptome “ 54rd American Society For Mass Spectrometry conference (ASMS), Seattle (USA), 2006

### Références

1. Chaurand, P.; Caprioli, R. M., Direct profiling and imaging of peptides and proteins from mammalian cells and tissue sections by mass spectrometry. *Electrophoresis* **2002**, 23, (18), 3125-35.
2. Chaurand, P.; Fouchecourt, S.; DaGue, B. B.; Xu, B. J.; Reyzer, M. L.; Orgebin-Crist, M. C.; Caprioli, R. M., Profiling and imaging proteins in the mouse epididymis by imaging mass spectrometry. *Proteomics* **2003**, 3, (11), 2221-39.
3. Fournier, I.; Day, R.; Salzet, M., Direct analysis of neuropeptides by in situ MALDI-TOF mass spectrometry in the rat brain. *Neuro Endocrinol Lett* **2003**, 24, (1-2), 9-14.
4. Karas, M.; Hillenkamp, F., Laser desorption ionization of proteins with molecular masses exceeding 10,000 daltons. *Anal Chem* **1988**, 60, (20), p. 2299-301.
5. Hillenkamp, F.; Karas, M., Mass spectrometry of peptides and proteins by matrix-assisted ultraviolet laser desorption/ionization. *Methods Enzymol* **1990**, 193, 280-95.
6. Tanaka, K.; Waki, H.; Ido, Y.; Akita, S.; Yoshida, Y.; Yoshida, T., Protein and polymer analysis up to m/z 100,000 by laser ionization time-of-flight mass spectrometry. *Rapid Commun. Mass Spectrom.* **1988**, 2, 151-153.
7. Cotter, R. J., Time-of-Flight Mass Spectrometry: Instrumentation and Applications in Biological Research. *American Chemical Society Publication* **1997**.
8. Wiley, W. C. M., I. H. , Time-of-Flight Mass Spectrometer with Improved Resolution. *Rev. Sci. Instrum.* **1955**, 26.
9. Mamyrin, B., Time-of-flight mass spectrometry (concepts, achievements, and prospects). *International Journal of Mass Spectrometry* **2001**, 206 (3), 251-266.
10. Mamyrin, B. A. K., V. I.; Shmikk, D. V.; Zagulin, V. A., The mass-reflectron, a new nonmagnetic time-of-flight mass spectrometer with high resolution. *Sov. Phys. JETP*, **1973**, 37, 45.
11. Aisemberg, G. O.; Macagno, E. R., Lox1, an Antennapedia-class homeobox gene, is expressed during leech gangliogenesis in both transient and stable central neurons. *Dev Biol* **1994**, 161, (2), 455-65.
12. Li, K. W.; Hoek, R. M.; Smith, F.; Jimenez, C. R.; van der Schors, R. C.; van Veelen, P. A.; Chen, S.; van der Greef, J.; Parish, D. C.; Benjamin, P. R.; et al., Direct peptide profiling by mass spectrometry of single identified neurons reveals complex neuropeptide-processing pattern. *J Biol Chem* **1994**, 269, (48), 30288-92.
13. Li, L.; Garden, R. W.; Sweedler, J. V., Single-cell MALDI: a new tool for direct peptide profiling. *Trends Biotechnol* **2000**, 18, (4), 151-60.

14. Jimenez, C. R.; Li, K. W.; Dreisewerd, K.; Spijker, S.; Kingston, R.; Bateman, R. H.; Burlingame, A. L.; Smit, A. B.; van Minnen, J.; Geraerts, W. P., Direct mass spectrometric peptide profiling and sequencing of single neurons reveals differential peptide patterns in a small neuronal network. *Biochemistry* **1998**, *37*, (7), 2070-6.
15. Sweedler, J. V.; Li, L.; Floyd, P.; Gilly, W., Mass spectrometric survey of peptides in cephalopods with an emphasis on the FMRFamide-related peptides. *J Exp Biol* **2000**, *203*, (Pt 23), 3565-73.
16. Li, K. W.; van Golen, F. A.; van Minnen, J.; van Veelen, P. A.; van der Greef, J.; Geraerts, W. P., Structural identification, neuronal synthesis, and role in male copulation of myomodulin-A of *Lymnaea*: a study involving direct peptide profiling of nervous tissue by mass spectrometry. *Brain Res Mol Brain Res* **1994**, *25*, (3-4), 355-8.
17. Li, K. W.; Jimenez, C. R.; Van Veelen, P. A.; Geraerts, W. P., Processing and targeting of a molluscan egg-laying peptide prohormone as revealed by mass spectrometric peptide fingerprinting and peptide sequencing. *Endocrinology* **1994**, *134*, (4), 1812-9.
18. Jimenez, C. R.; van Veelen, P. A.; Li, K. W.; Wildering, W. C.; Geraerts, W. P.; Tjaden, U. R.; van der Greef, J., Neuropeptide expression and processing as revealed by direct matrix-assisted laser desorption ionization mass spectrometry of single neurons. *J Neurochem* **1994**, *62*, (1), 404-7.
19. Li, L.; Romanova, E. V.; Rubakhin, S. S.; Alexeeva, V.; Weiss, K. R.; Vilim, F. S.; Sweedler, J. V., Peptide profiling of cells with multiple gene products: combining immunochemistry and MALDI mass spectrometry with on-plate microextraction. *Anal Chem* **2000**, *72*, (16), 3867-74.
20. Redeker, V.; Toullec, J. Y.; Vinh, J.; Rossier, J.; Soye, D., Combination of peptide profiling by matrix-assisted laser desorption/ionization time-of-flight mass spectrometry and immunodetection on single glands or cells. *Anal Chem* **1998**, *70*, (9), 1805-11.
21. Bulet, P.; Urge, L.; Ohresser, S.; Hetru, C.; Otvos, L., Jr., Enlarged scale chemical synthesis and range of activity of drosocin, an O-glycosylated antibacterial peptide of *Drosophila*. *Eur J Biochem* **1996**, *238*, (1), 64-9.
22. Caprioli, R. M.; Farmer, T. B.; Gile, J., Molecular imaging of biological samples: localization of peptides and proteins using MALDI-TOF MS. *Anal Chem* **1997**, *69*, (23), 4751-60.
23. Spengler, B., Ion Imaging and Confocal Microscopy with a new Scanning UV- Laser Microprobe. *42nd Annual Conference on Mass Spectrometry and Allied Topics, Chicago, Illinois* **1994**.
24. Chaurand, P.; Schwartz, S. A.; Capriolo, R. M., Profiling and imaging proteins in tissue sections by MS. *Anal Chem* **2004**, *76*, (5), 87A-93A.

25. Chaurand, P.; Sanders, M. E.; Jensen, R. A.; Caprioli, R. M., Proteomics in diagnostic pathology: profiling and imaging proteins directly in tissue sections. *Am J Pathol* **2004**, 165, (4), 1057-68.
26. Stoeckli, M. C., P.; Caprioli, R. M. , Applications of MALDI MS Imaging to Biological Samples. *Proceedings of the 47th Annual Conference on Mass Spectrometry and Allied Topics, Dallas, TX, June 13-17* **1999**.
27. Chaurand, P.; Schwartz, S. A.; Billheimer, D.; Xu, B. J.; Crecelius, A.; Caprioli, R. M., Integrating histology and imaging mass spectrometry. *Anal Chem* **2004**, 76, (4), 1145-55.
28. Lemaire, R.; Wisztorski, M.; Desmons, A.; Tabet, J. C.; Day, R.; Salzet, M.; Fournier, I., MALDI-MS direct tissue analysis of proteins: Improving signal sensitivity using organic treatments. *Anal Chem* **2006**, 78, (20), 7145-53.
29. Lemaire, R.; Desmons, A.; Ducoroy, P.; Tabet, J. C.; Salzet, M.; Fournier, I. In *Direct analysis and MALDI imaging on formalin-fixed paraffin-embedded (FFPE) tissues : Application to Parkinson disease*, 54th ASMS Conference on Mass Spectrometry, Seattle, Washington, 2006; Seattle, Washington, 2006; pp ThP18-333.
30. Groseclose, M. R.; Andersson, M.; Hardesty, W. M.; Caprioli, R. M., Identification of proteins directly from tissue: in situ tryptic digestions coupled with imaging mass spectrometry. *J Mass Spectrom* **2007**, 42, (2), 254-62.
31. Lemaire, R.; Desmons, A.; Tabet, J. C.; Day, R.; Salzet, M.; Fournier, I., Direct analysis and MALDI imaging of formalin-fixed, paraffin-embedded tissue sections. *J Proteome Res* **2007**, 6, (4), 1295-305.
32. Schwartz, S. A.; Reyzer, M. L.; Caprioli, R. M., Direct tissue analysis using matrix-assisted laser desorption/ionization mass spectrometry: practical aspects of sample preparation. *J Mass Spectrom* **2003**, 38, (7), 699-708.
33. Baluya, D. L.; Garrett, T. J.; Yost, R. A., Automated MALDI Matrix Deposition Method with Inkjet Printing for Imaging Mass Spectrometry. *Anal Chem* **2007**.
34. Spengler, B.; Hubert, M., Scanning microprobe matrix-assisted laser desorption ionization (SMALDI) mass spectrometry: instrumentation for sub-micrometer resolved LDI and MALDI surface analysis. *J Am Soc Mass Spectrom* **2002**, 13, (6), 735-48.
35. Dreisewerd, K.; Schiirenberg, M.; Karas, M.; Hillenkamp, F., Influence of the laser intensity and spot size on the desorption of molecules and ions in matrix-assisted laser desorption/ionization with a uniform beam profile. *International Journal of Mass Spectrometry and Ion Processes* **1995**, 141, 127-148.
36. Chaurand, P.; Schriver, K. E.; Caprioli, R. M., Instrument design and characterization for high resolution MALDI-MS imaging of tissue sections. *J Mass Spectrom* **2007**, 42, (4), 476-89.

37. Klerk, L. A. B., A.; Fletcher, I.W.; van Liere, R.; Heeren, R.M.A. , Extended data analysis strategies for high resolution imaging MS: New methods to deal with extremely large image hyperspectral datasets. *Int J Mass Spectrom* **2007**, 260, 222-236.
38. Norris, J. L.; Cornett, D. S.; Mobley, J. A.; Andersson, M.; Seeley, E. H.; Chaurand, P.; Caprioli, R. M., Processing MALDI Mass Spectra to Improve Mass Spectral Direct Tissue Analysis. *Int J Mass Spectrom* **2007**, 260, (2-3), 212-221.
39. Trim, P. J.; Atkinson, S. J.; Princivalle, A. P.; Marshall, P. S.; West, A.; Clench, M. R., Matrix-assisted laser desorption/ionisation mass spectrometry imaging of lipids in rat brain tissue with integrated unsupervised and supervised multivariate statistical analysis. *Rapid Commun Mass Spectrom* **2008**, 22, (10), 1503-9.
40. McCombie, G.; Staab, D.; Stoeckli, M.; Knochenmuss, R., Spatial and spectral correlations in MALDI mass spectrometry images by clustering and multivariate analysis. *Anal Chem* **2005**, 77, (19), 6118-24.
41. Cornett, D. S.; Reyzer, M. L.; Chaurand, P.; Caprioli, R. M., MALDI imaging mass spectrometry: molecular snapshots of biochemical systems. *Nat Methods* **2007**, 4, (10), 828-33.
42. Chaurand, P.; DaGue, B. B.; Pearsall, R. S.; Threadgill, D. W.; Caprioli, R. M., Profiling proteins from azoxymethane-induced colon tumors at the molecular level by matrix-assisted laser desorption/ionization mass spectrometry. *Proteomics* **2001**, 1, (10), 1320-6.
43. Hintersteiner, M.; Enz, A.; Frey, P.; Jaton, A. L.; Kinzy, W.; Kneuer, R.; Neumann, U.; Rudin, M.; Staufenbiel, M.; Stoeckli, M.; Wiederhold, K. H.; Gremlich, H. U., In vivo detection of amyloid-beta deposits by near-infrared imaging using an oxazine-derivative probe. *Nat Biotechnol* **2005**, 23, (5), 577-83.
44. Langstrom, B.; Andren, P. E.; Lindhe, O.; Svedberg, M.; Hall, H., In vitro imaging techniques in neurodegenerative diseases. *Mol Imaging Biol* **2007**, 9, (4), 161-75.
45. Laurent, C.; Levinson, D. F.; Schwartz, S. A.; Harrington, P. B.; Markey, S. P.; Caprioli, R. M.; Levitt, P., Direct profiling of the cerebellum by matrix-assisted laser desorption/ionization time-of-flight mass spectrometry: A methodological study in postnatal and adult mouse. *J Neurosci Res* **2005**, 81, (5), 613-21.
46. Lemaire, R.; Menguellet, S. A.; Stauber, J.; Marchaudon, V.; Lucot, J. P.; Collinet, P.; Farine, M. O.; Vinatier, D.; Day, R.; Ducoroy, P.; Salzet, M.; Fournier, I., Specific MALDI imaging and profiling for biomarker hunting and validation: fragment of the 11S proteasome activator complex, Reg alpha fragment, is a new potential ovary cancer biomarker. *J Proteome Res* **2007**, 6, (11), 4127-34.
47. Meistermann, H.; Norris, J. L.; Aerni, H. R.; Cornett, D. S.; Friedlein, A.; Erskine, A. R.; Augustin, A.; De Vera Mudry, M. C.; Ruepp, S.; Suter, L.; Langen, H.; Caprioli, R. M.; Ducret, A., Biomarker discovery by imaging mass spectrometry: transthyretin

- is a biomarker for gentamicin-induced nephrotoxicity in rat. *Mol Cell Proteomics* **2006**, 5, (10), 1876-86.
48. Pierson, J.; Svenningsson, P.; Caprioli, R. M.; Andren, P. E., Increased levels of ubiquitin in the 6-OHDA-lesioned striatum of rats. *J Proteome Res* **2005**, 4, (2), 223-6.
  49. Reyzer, M. L.; Caprioli, R. M., MALDI mass spectrometry for direct tissue analysis: a new tool for biomarker discovery. *J Proteome Res* **2005**, 4, (4), 1138-42.
  50. Schwartz, S. A.; Weil, R. J.; Thompson, R. C.; Shyr, Y.; Moore, J. H.; Toms, S. A.; Johnson, M. D.; Caprioli, R. M., Proteomic-based prognosis of brain tumor patients using direct-tissue matrix-assisted laser desorption ionization mass spectrometry. *Cancer Res* **2005**, 65, (17), 7674-81.
  51. Skold, K.; Svensson, M.; Nilsson, A.; Zhang, X.; Nydahl, K.; Caprioli, R. M.; Svenningsson, P.; Andren, P. E., Decreased striatal levels of PEP-19 following MPTP lesion in the mouse. *J Proteome Res* **2006**, 5, (2), 262-9.
  52. Cornett, D. S.; Mobley, J. A.; Dias, E. C.; Andersson, M.; Arteaga, C. L.; Sanders, M. E.; Caprioli, R. M., A novel histology-directed strategy for MALDI-MS tissue profiling that improves throughput and cellular specificity in human breast cancer. *Mol Cell Proteomics* **2006**, 5, (10), 1975-83.
  53. Chaurand, P.; Norris, J. L.; Cornett, D. S.; Mobley, J. A.; Caprioli, R. M., New developments in profiling and imaging of proteins from tissue sections by MALDI mass spectrometry. *J Proteome Res* **2006**, 5, (11), 2889-900.
  54. Schwamborn, K.; Krieg, R. C.; Reska, M.; Jakse, G.; Knuechel, R.; Wellmann, A., Identifying prostate carcinoma by MALDI-Imaging. *Int J Mol Med* **2007**, 20, (2), 155-9.
  55. Shimma, S.; Sugiura, Y.; Hayasaka, T.; Hoshikawa, Y.; Noda, T.; Setou, M., MALDI-based imaging mass spectrometry revealed abnormal distribution of phospholipids in colon cancer liver metastasis. *J Chromatogr B Analyt Technol Biomed Life Sci* **2007**, 855, (1), 98-103.
  56. Schwartz, S. A.; Weil, R. J.; Johnson, M. D.; Toms, S. A.; Caprioli, R. M., Protein profiling in brain tumors using mass spectrometry: feasibility of a new technique for the analysis of protein expression. *Clin Cancer Res* **2004**, 10, (3), 981-7.
  57. Wisztorski, M.; Croix, D.; Macagno, E.; Fournier, I.; Salzet, M., Molecular MALDI imaging: An emerging technology for neuroscience studies. *Dev Neurobiol* **2008**, 68, (6), 845-58.
  58. Jimenez, C. R.; Burlingame, A. L., Ultramicroanalysis of peptide profiles in biological samples using MALDI mass spectrometry. *Exp Nephrol* **1998**, 6, (5), 421-8.
  59. Jimenez, C. R.; Li, K. W.; Dreisewerd, K.; Mansvelder, H. D.; Brussaard, A. B.; Reinhold, B. B.; Van der Schors, R. C.; Karas, M.; Hillenkamp, F.; Burbach, J. P.;

- Costello, C. E.; Geraerts, W. P., Pattern changes of pituitary peptides in rat after salt-loading as detected by means of direct, semiquantitative mass spectrometric profiling. *Proc Natl Acad Sci U S A* **1997**, 94, (17), 9481-6.
60. Kruse, R.; Sweedler, J. V., Spatial profiling invertebrate ganglia using MALDI MS. *J Am Soc Mass Spectrom* **2003**, 14, (7), 752-9.
61. Hummon, A. B.; Amare, A.; Sweedler, J. V., Discovering new invertebrate neuropeptides using mass spectrometry. *Mass Spectrom Rev* **2006**, 25, (1), 77-98.
62. Rubakhin, S. S.; Greenough, W. T.; Sweedler, J. V., Spatial profiling with MALDI MS: distribution of neuropeptides within single neurons. *Anal Chem* **2003**, 75, (20), 5374-80.
63. Altelaar, A. F.; van Minnen, J.; Jimenez, C. R.; Heeren, R. M.; Piersma, S. R., Direct molecular imaging of *Lymnaea stagnalis* nervous tissue at subcellular spatial resolution by mass spectrometry. *Anal Chem* **2005**, 77, (3), 735-41.
64. Croll, R. P.; Van Minnen, J., Distribution of the peptide Ala-Pro-Gly-Trp-NH<sub>2</sub> (APGWamide) in the nervous system and periphery of the snail *Lymnaea stagnalis* as revealed by immunocytochemistry and in situ hybridization. *J Comp Neurol* **1992**, 324, (4), 567-74.
65. de Lange, R. P.; de Boer, P. A.; ter Maat, A.; Tensen, C. P.; van Minnen, J., Transmitter identification in neurons involved in male copulation behavior in *Lymnaea stagnalis*. *J Comp Neurol* **1998**, 395, (4), 440-9.
66. Dreisewerd, K.; Lemaire, R.; Pohlentz, G.; Salzet, M.; Wisztorski, M.; Berkenkamp, S.; Fournier, I., Molecular profiling of native and matrix-coated tissue slices from rat brain by infrared and ultraviolet laser desorption/ionization orthogonal time-of-flight mass spectrometry. *Anal Chem* **2007**, 79, (6), 2463-71.
67. Falth, M.; Savitski, M. M.; Nielsen, M. L.; Kjeldsen, F.; Andren, P. E.; Zubarev, R. A., SwedCAD, a database of annotated high-mass accuracy MS/MS spectra of tryptic peptides. *J Proteome Res* **2007**, 6, (10), 4063-7.
68. Falth, M.; Skold, K.; Norrman, M.; Svensson, M.; Fenyó, D.; Andren, P. E., SwePep, a database designed for endogenous peptides and mass spectrometry. *Mol Cell Proteomics* **2006**, 5, (6), 998-1005.
69. Jespersen, S.; Chaurand, P.; van Strien, F. J.; Spengler, B.; van der Greef, J., Direct sequencing of neuropeptides in biological tissue by MALDI-PSD mass spectrometry. *Anal Chem* **1999**, 71, (3), 660-6.
70. Verhaert, P. D. P. C., M.C.; Pekar, T.M.; Miller, K., Neuropeptide imaging on an LTQ with vMALDI source: The complete 'all-in-one' peptidome analysis. *Int J Mass Spectrom* **2007**, 260, 177-184.
71. DeKeyser, S. S.; Kutz-Naber, K. K.; Schmidt, J. J.; Barrett-Wilt, G. A.; Li, L., Imaging mass spectrometry of neuropeptides in decapod crustacean neuronal tissues. *J Proteome Res* **2007**, 6, (5), 1782-91.

72. Jackson, S. N.; Wang, H. Y.; Woods, A. S., Direct profiling of lipid distribution in brain tissue using MALDI-TOFMS. *Anal Chem* **2005**, *77*, (14), 4523-7.
73. Jackson, S. N.; Wang, H. Y.; Woods, A. S., In situ structural characterization of phosphatidylcholines in brain tissue using MALDI-MS/MS. *J Am Soc Mass Spectrom* **2005**, *16*, (12), 2052-6.
74. Jackson, S. N.; Wang, H. Y.; Woods, A. S., In situ structural characterization of glycerophospholipids and sulfatides in brain tissue using MALDI-MS/MS. *J Am Soc Mass Spectrom* **2007**, *18*, (1), 17-26.
75. Jackson, S. N.; Wang, H. Y.; Woods, A. S.; Ugarov, M.; Egan, T.; Schultz, J. A., Direct tissue analysis of phospholipids in rat brain using MALDI-TOFMS and MALDI-ion mobility-TOFMS. *J Am Soc Mass Spectrom* **2005**, *16*, (2), 133-8.
76. Pierson, J.; Norris, J. L.; Aerni, H. R.; Svenningsson, P.; Caprioli, R. M.; Andren, P. E., Molecular profiling of experimental Parkinson's disease: direct analysis of peptides and proteins on brain tissue sections by MALDI mass spectrometry. *J Proteome Res* **2004**, *3*, (2), 289-95.
77. Stauber, J.; Lemaire, R.; Franck, J.; Bonnel, D.; Croix, D.; Day, R.; Wisztorski, M.; Fournier, I.; Salzet, M., MALDI imaging of formalin-fixed paraffin-embedded tissues: application to model animals of Parkinson disease for biomarker hunting. *J Proteome Res* **2008**, *7*, (3), 969-78.
78. Stoeckli, M.; Knochenmuss, R.; McCombie, G.; Mueller, D.; Rohner, T.; Staab, D.; Wiederhold, K. H., MALDI MS imaging of amyloid. *Methods Enzymol* **2006**, *412*, 94-106.
79. Stoeckli, M.; Staab, D.; Staufenbiel, M.; Wiederhold, K. H.; Signor, L., Molecular imaging of amyloid beta peptides in mouse brain sections using mass spectrometry. *Anal Biochem* **2002**, *311*, (1), 33-9.
80. Hsieh, Y.; Chen, J.; Korfmacher, W. A., Mapping pharmaceuticals in tissues using MALDI imaging mass spectrometry. *J Pharmacol Toxicol Methods* **2007**, *55*, (2), 193-200.
81. Rubakhin, S. S.; Jurchen, J. C.; Monroe, E. B.; Sweedler, J. V., Imaging mass spectrometry: fundamentals and applications to drug discovery. *Drug Discov Today* **2005**, *10*, (12), 823-37.
82. Rudin, M.; Rausch, M.; Stoeckli, M., Molecular imaging in drug discovery and development: potential and limitations of nonnuclear methods. *Mol Imaging Biol* **2005**, *7*, (1), 5-13.
83. Reyzer, M. L.; Hsieh, Y.; Ng, K.; Korfmacher, W. A.; Caprioli, R. M., Direct analysis of drug candidates in tissue by matrix-assisted laser desorption/ionization mass spectrometry. *J Mass Spectrom* **2003**, *38*, (10), 1081-92.

84. Hsieh, Y.; Casale, R.; Fukuda, E.; Chen, J.; Knemeyer, I.; Wingate, J.; Morrison, R.; Korfmacher, W., Matrix-assisted laser desorption/ionization imaging mass spectrometry for direct measurement of clozapine in rat brain tissue. *Rapid Commun Mass Spectrom* **2006**, 20, (6), 965-72.
85. O'Brien, E.; Dedova, I.; Duffy, L.; Cordwell, S.; Karl, T.; Matsumoto, I., Effects of chronic risperidone treatment on the striatal protein profiles in rats. *Brain Res* **2006**, 1113, (1), 24-32.
86. Jones, J. J.; Borgmann, S.; Wilkins, C. L.; O'Brien, R. M., Characterizing the phospholipid profiles in mammalian tissues by MALDI FTMS. *Anal Chem* **2006**, 78, (9), 3062-71.
87. Khatib-Shahidi, S.; Andersson, M.; Herman, J. L.; Gillespie, T. A.; Caprioli, R. M., Direct molecular analysis of whole-body animal tissue sections by imaging MALDI mass spectrometry. *Anal Chem* **2006**, 78, (18), 6448-56.
88. Burrell, M.; Earnshaw, C.; Clench, M., Imaging Matrix Assisted Laser Desorption Ionization Mass Spectrometry: a technique to map plant metabolites within tissues at high spatial resolution. *J Exp Bot* **2007**, 58, (4), 757-63.
89. Mullen, A. K.; Clench, M. R.; Crosland, S.; Sharples, K. R., Determination of agrochemical compounds in soya plants by imaging matrix-assisted laser desorption/ionisation mass spectrometry. *Rapid Commun Mass Spectrom* **2005**, 19, (18), 2507-16.
90. Robinson, S.; Warburton, K.; Seymour, M.; Clench, M.; Thomas-Oates, J., Localization of water-soluble carbohydrates in wheat stems using imaging matrix-assisted laser desorption ionization mass spectrometry. *New Phytol* **2007**, 173, (2), 438-44.
91. Takats, Z.; Wiseman, J. M.; Cooks, R. G., Ambient mass spectrometry using desorption electrospray ionization (DESI): instrumentation, mechanisms and applications in forensics, chemistry, and biology. *J Mass Spectrom* **2005**, 40, (10), 1261-75.
92. Chen, H.; Talaty, N. N.; Takats, Z.; Cooks, R. G., Desorption electrospray ionization mass spectrometry for high-throughput analysis of pharmaceutical samples in the ambient environment. *Anal Chem* **2005**, 77, (21), 6915-27.
93. Kauppila, T. J.; Wiseman, J. M.; Ketola, R. A.; Kotiaho, T.; Cooks, R. G.; Kostianen, R., Desorption electrospray ionization mass spectrometry for the analysis of pharmaceuticals and metabolites. *Rapid Commun Mass Spectrom* **2006**, 20, (3), 387-92.
94. Lemaire, R.; Stauber, J.; Wisztorski, M.; Van Camp, C.; Desmons, A.; Deschamps, M.; Proess, G.; Rudlof, I.; Woods, A. S.; Day, R.; Salzet, M.; Fournier, I., Tag-mass: specific molecular imaging of transcriptome and proteome by mass spectrometry based on photocleavable tag. *J Proteome Res* **2007**, 6, (6), 2057-67.

95. Clerens, S.; Ceuppens, R.; Arckens, L., CreateTarget and Analyze This!: new software assisting imaging mass spectrometry on Bruker Reflex IV and Ultraflex II instruments. *Rapid Commun Mass Spectrom* **2006**, 20, (20), 3061-6.
96. Touboul, D., Roy, S., Germain, D.P., Chaminade, P., Brunelle, A., Lapr evote, O., MALDI-TOF and cluster-TOF-SIMS imaging of Fabry disease biomarkers. *International Journal of Mass Spectrometry* **2006**, 260, 158-165.
97. von Bernhardt, R.; Muller, K. J., Repair of the central nervous system: lessons from lesions in leeches. *J Neurobiol* **1995**, 27, (3), 353-66.
98. Raible, F.; Tessmar-Raible, K.; Osoegawa, K.; Wincker, P.; Jubin, C.; Balavoine, G.; Ferrier, D.; Benes, V.; de Jong, P.; Weissenbach, J.; Bork, P.; Arendt, D., Vertebrate-type intron-rich genes in the marine annelid *Platynereis dumerilii*. *Science* **2005**, 310, (5752), 1325-6.
99. Denes, A. S.; Jekely, G.; Steinmetz, P. R.; Raible, F.; Snyman, H.; Prud'homme, B.; Ferrier, D. E.; Balavoine, G.; Arendt, D., Molecular architecture of annelid nerve cord supports common origin of nervous system centralization in bilateria. *Cell* **2007**, 129, (2), 277-88.
100. Coggeshall, R. E.; Fawcett, D. W., The Fine Structure of the Central Nervous System of the Leech, *Hirudo Medicinalis*. *J Neurophysiol* **1964**, 27, 229-89.
101. Nicholls, J. G.; Kuffler, S. W., Extracellular Space as a Pathway for Exchange between Blood and Neurons in the Central Nervous System of the Leech: Ionic Composition of Glial Cells and Neurons. *J Neurophysiol* **1964**, 27, 645-71.
102. Kuffler, S. W.; Nicholls, J. G., How do materials exchange between blood and nerve cells in the brain? *Perspect Biol Med* **1965**, 9, (1), 69-76.
103. Lefebvre, C.; Salzet, M., Annelid neuroimmune system. *Curr Pharm Des* **2003**, 9, (2), 149-58.
104. Salzet, M.; Day, R., [Endocrine markers of cellular immunity: defining the endocrine phenotype]. *J Soc Biol* **2003**, 197, (2), 97-101.
105. Brogden, K. A.; Guthmiller, J. M.; Salzet, M.; Zasloff, M., The nervous system and innate immunity: the neuropeptide connection. *Nat Immunol* **2005**, 6, (6), 558-64.
106. Salzet, M., Neuropeptide-derived antimicrobial peptides from invertebrates for biomedical applications. *Curr Med Chem* **2005**, 12, (26), 3055-61.
107. Salzet, M.; Tasiemski, A.; Cooper, E., Innate immunity in lophotrochozoans: the annelids. *Curr Pharm Des* **2006**, 12, (24), 3043-50.
108. Blackshaw, S. E.; Henderson, L. P.; Malek, J.; Porter, D. M.; Gross, R. H.; Angstadt, J. D.; Levasseur, S. M.; Maue, R. A., Single-cell analysis reveals cell-specific patterns of expression of a family of putative voltage-gated sodium channel genes in the leech. *J Neurobiol* **2003**, 55, (3), 355-71.

109. Emes, R. D.; Wang, W. Z.; Lanary, K.; Blackshaw, S. E., HmCRIP, a cysteine-rich intestinal protein, is expressed by an identified regenerating nerve cell. *FEBS Lett* **2003**, 533, (1-3), 124-8.
110. Sawyer, R. T., Leech biology and behaviour. *Clarendon Press, Oxford, England*. **1986**, I, II, III.
111. Macagno, E. R., Number and distribution of neurons in leech segmental ganglia. *J Comp Neurol* **1980**, 190, (2), 283-302.
112. Baylor, D. A.; Nicholls, J. G., Patterns of regeneration between individual nerve cells in the central nervous system of the leech. *Nature* **1971**, 232, (5308), 268-70.
113. Nicholls, J. G.; Van Essen, D., The nervous system of the leech. *Sci Am* **1974**, 230, (1), 38-48.
114. Blackshaw, S. E.; Nicholls, J. G., Neurobiology and development of the leech. *J Neurobiol* **1995**, 27, (3), 267-76.
115. Miyazaki, S.; Nicholls, J. G., The properties and connections of nerve cells in leech ganglia maintained in culture. *Proc R Soc Lond B Biol Sci* **1976**, 194, (1116), 295-311.
116. Wallace, B. G.; Adal, M. N.; Nicholls, J. G., Regeneration of synaptic connections by sensory neurons in leech ganglia maintained in culture. *Proc R Soc Lond B Biol Sci* **1977**, 199, (1137), 567-85.
117. Bowling, D.; Nicholls, J.; Parnas, I., Destruction of a single cell in the central nervous system of the leech as a means of analysing its connexions and functional role. *J Physiol* **1978**, 282, 169-80.
118. Ready, D. F.; Nicholls, J., Identified neurones isolated from leech CNS make selective connections in culture. *Nature* **1979**, 281, (5726), 67-9.
119. Kristan, W. B., Jr.; Eisenhart, F. J.; Johnson, L. A.; French, K. A., Development of neuronal circuits and behaviors in the medicinal leech. *Brain Res Bull* **2000**, 53, (5), 561-70.
120. Baker, M. W.; Macagno, E. R., The role of a LAR-like receptor tyrosine phosphatase in growth cone collapse and mutual-avoidance by sibling processes. *J Neurobiol* **2000**, 44, (2), 194-203.
121. Wang, H.; Macagno, E. R., The establishment of peripheral sensory arbors in the leech: in vivo time-lapse studies reveal a highly dynamic process. *J Neurosci* **1997**, 17, (7), 2408-19.
122. Baptista, C. A.; Gershon, T. R.; Macagno, E. R., Peripheral organs control central neurogenesis in the leech. *Nature* **1990**, 346, (6287), 855-8.
123. Blair, S. S.; Martindale, M. Q.; Shankland, M., Interactions between adjacent ganglia bring about the bilaterally alternating differentiation of RAS and CAS neurons in the leech nerve cord. *J Neurosci* **1990**, 10, (10), 3183-93.

124. Gan, W. B.; Macagno, E. R., Developing neurons use a putative pioneer's peripheral arbor to establish their terminal fields. *J Neurosci* **1995**, 15, (5 Pt 1), 3254-62.
125. Becker, T. S.; Bothe, G.; Berliner, A. J.; Macagno, E. R., Identified central neurons convey a mitogenic signal from a peripheral target to the CNS. *Development* **1996**, 122, (8), 2331-7.
126. Johnson, L. A.; Kristan, W. B.; Jellies, J.; French, K. A., Disruption of peripheral target contact influences the development of identified central dendritic branches in a leech motor neuron in vivo. *J Neurobiol* **2000**, 43, (4), 365-78.
127. Jellies, J.; Johansen, K. M.; Johansen, J., Ectopic CNS projections guide peripheral neuron axons along novel pathways in leech embryos. *Dev Biol* **2000**, 218, (2), 137-45.
128. Duan, Y.; Panoff, J.; Burrell, B. D.; Sahley, C. L.; Muller, K. J., Repair and regeneration of functional synaptic connections: cellular and molecular interactions in the leech. *Cell Mol Neurobiol* **2005**, 25, (2), 441-50.
129. Schikorski, D., Cuvillier Hot, V., Leippe, M., Macagno, E., Salzet, M., Tasiemski, A., The medicinal leech as a model for studying the antimicrobial response of the central nervous system. *J. Immunol* **2008**.
130. Bretin, S.; Reibel, S.; Charrier, E.; Maus-Moatti, M.; Auvergnon, N.; Thevenoux, A.; Glowinski, J.; Rogemond, V.; Premont, J.; Honnorat, J.; Gauchy, C., Differential expression of CRMP1, CRMP2A, CRMP2B, and CRMP5 in axons or dendrites of distinct neurons in the mouse brain. *J Comp Neurol* **2005**, 486, (1), 1-17.
131. Zhang, J.; Goodlett, D. R., Proteomic approach to studying Parkinson's disease. *Mol Neurobiol* **2004**, 29, (3), 271-88.
132. Davidsson, P.; Sjogren, M., The use of proteomics in biomarker discovery in neurodegenerative diseases. *Dis Markers* **2005**, 21, (2), 81-92.
133. Zhang, J.; Sokal, I.; Peskind, E. R.; Quinn, J. F.; Jankovic, J.; Kenney, C.; Chung, K. A.; Millard, S. P.; Nutt, J. G.; Montine, T. J., CSF multianalyte profile distinguishes Alzheimer and Parkinson diseases. *Am J Clin Pathol* **2008**, 129, (4), 526-9.
134. Dickson, D. W.; Feany, M. B.; Yen, S. H.; Mattiace, L. A.; Davies, P., Cytoskeletal pathology in non-Alzheimer degenerative dementia: new lesions in diffuse Lewy body disease, Pick's disease, and corticobasal degeneration. *J Neural Transm Suppl* **1996**, 47, 31-46.
135. Uberbacher, E. C.; Xu, Y.; Shah, M. B.; Olman, V.; Parang, M.; Mural, R. J., An editing environment for DNA sequence analysis and annotation (extended abstract). *Pac Symp Biocomput* **1998**, 217-27.
136. Andersson, M.; Groseclose, M. R.; Deutch, A. Y.; Caprioli, R. M., Imaging mass spectrometry of proteins and peptides: 3D volume reconstruction. *Nat Methods* **2008**, 5, (1), 101-8.

137. Crecelius, A. C.; Cornett, D. S.; Caprioli, R. M.; Williams, B.; Dawant, B. M.; Bodenheimer, B., Three-dimensional visualization of protein expression in mouse brain structures using imaging mass spectrometry. *J Am Soc Mass Spectrom* **2005**, 16, (7), 1093-9.

### Annexes

#### 8.1 Publications

## 8.2 Code source

**MITICS (MALDI IMAGING TRUE-IMAGE COMPUTING SYSTEM):  
A NEW OPEN SOURCE MASS SPECTROMETRY IMAGING SOFTWARE**

**Olivia Jardin-Mathé<sup>1</sup>, David Bonnel<sup>1</sup>, Julien Franck<sup>1</sup>, Maxence Wisztorski<sup>1</sup>, Eduardo Macagno<sup>2</sup>, Isabelle Fournier<sup>1\*</sup>, Michel Salzet<sup>1\*</sup>**

<sup>1</sup>University of Lille1, CNRS FRE 2933, MALDI Imaging team, Laboratoire de Neuroimmunologie des Annélides, Villeneuve d'Ascq, France

<sup>2</sup>Section of Cell and Developmental Biology, Division of Biological Sciences, University of California San Diego, La Jolla, CA 92093, USA

**ABSTRACT**

MITICS (MALDI Imaging True-Imaging Computing System) is an open source, universal software package developed for the analysis of MS data from all MALDI-MSI instruments, unlike MMSIT and Fleximaging, which were developed specifically for Applied Biosystems and Bruker Daltonics instruments, respectively. Following automated spectrum acquisition with any MALDI instrument, MITICS converts all spectrum files to XML format prior to using an Interactive Data Language (IDL) environment to construct and display molecular images. For spectra obtained from either Parkinson's Disease model rat brain slices or whole mount dissected leech embryos, side-by-side comparison of molecular maps produced with the different software packages show a significantly better signal-to-noise ratio and sharpness for MITICS than, for non-normalized images obtained with company software package. Among the great advantages of MITICS is its capacity to generate sharp molecular maps without resorting to additional operations, such as smoothing or background removal, which are used (but not clearly described) by other software packages in order to render better images. Thus, MITICS offers an advantageous alternative to commercially available and instrument-specific software, such as *Create Target* and *Analyze This!*, which are solely devoted to Bruker instruments (Reflex and Ultraflex). MITICS is available free of charge at URL.

**Keywords:** MALDI Imaging, Mass Spectrometry Imaging Software, Parkinson's rat model, Leech embryo

\*Co-corresponding authors : [michel.salzet@univ-lille1.fr](mailto:michel.salzet@univ-lille1.fr); [Isabelle.fournier@univ-lille1.fr](mailto:Isabelle.fournier@univ-lille1.fr)

## Introduction

Mass spectrometry analyzes proteins based on the mass-to-charge ( $m/z$ ) ratio of ionized molecules. A typical mass spectrometer consists of an ionization source, a mass analyzer, and a detector for counting the number of analytes at each  $m/z$  ratio. In addition, mass spectrometers are commonly coupled with separation devices, such as liquid chromatography instrumentation. Aside from providing rapid data on various proteins present in biological samples, mass spectrometry also offers specific information about posttranslational modifications associated with, for example, a particular disease state. Recent instrument developments and electronic improvements now place mass spectrometry at the forefront of many biological applications. In particular, this technology has been crucial in efforts to identify the human proteome. In combination with classical biochemical methods, mass spectrometry has been largely responsible for revealing an expanded biological diversity at the protein level, due to alternative splicing and post-translational modifications that could not be predicted by genome sequence analysis alone.

Although mass spectrometry technologies are efficient in the identification of novel proteins, until recently they did not allow the determination of chemicals distribution or the regulation of peptide/proteins expressed in tissues or single cells. With new emerging mass spectrometry technologies, it has now been demonstrated that direct tissue analysis is feasible using matrix-assisted laser desorption/ionization (MALDI) sources. A major advantage of direct MALDI analysis is to avoid time-consuming extraction, purification or separation steps, which also have the potential for producing artifacts. Therefore, direct MALDI analysis in tissue sections enables the acquisition of cellular expression profiles while maintaining cellular and molecular integrity<sup>1-7</sup>. With automation and the ability to reconstruct complex spectral data with imaging software, it is now possible to produce multiplex imaging maps of selected bio-molecules within tissue sections<sup>8,9</sup>. Thus, direct MALDI analysis obtained from tissue sections is convertible into imaging maps, a method now known as MALDI-imaging. MALDI-imaging combines the power of mass spectrometry, namely exquisite sensitivity, with unequivocal positional information within a normal morphological context<sup>8,9</sup>. Every MALDI imaging experiment is comprised of two steps: (a) sample preparation and treatment, and (b) data acquisition and processing. Given the nature and magnitude of the physical measurements, computationally powerful and transparent data processing and displaying tools are a critical requirement for the successful analysis of the large number of spectra obtained in any MALDI imaging experiment.

Currently, four principal MALDI imaging software packages are available, some commercially (Table 1). The original one, MMSIT, was written by M. Stoeckli for use on Applied Biosystems MALDI-TOF and MALDI TOF/TOF instruments<sup>8</sup>. This software is available at <http://www.maldi-msi.org>. For Applied Biosystems, BioMap and more recently Tissue view 1.0 have been developed for 4700 and 4800 MALDI TOF/TOF™ Analyzers, QStar Pulsar System, QSTAR® XL Hybrid LC/MS/MS System and the QSTAR. Bruker and Shimadzu have developed their own softwares, e.g. FlexImaging and Axima2 Analyze. The latest ones, Create Target and

Analyze this! are two new software assisting MALDI-IMS on Bruker Reflex IV and Ultraflex II instruments<sup>10</sup>.

We have developed MITICS as a universal MALDI imaging data processing and display software independent of the type of instrument used for data recording. It converts all reading files, such as .dat files in Data Explorer for Applied Biosystems, to .xml files, the format used by FlexAnalysis for Bruker Daltonics. MITICS uses two different interfaces, one to load all the spectra from each run into a single database that is readily available for repeated analysis, and the second to reconstruct and display interactively the molecular images of interest. MITICS offers two alternative computational tools to create molecular maps, using the maximum value of a peak to show where the sample is present, or use the integrated area of the peak above a set threshold. Two applications of MITICS, to biomarker discovery in brain tissues of rats treated with 6-OHDA, fixed in formalin and embedded for the past 12 years in paraffin, and to peptide mapping in dissected leech embryo whole mounts, are presented and compared to results obtained using FlexImaging software.

## **EXPERIMENTAL**

### **Materials**

$\alpha$ -cyano-4-hydroxycinnamic acid (HCCA), ammonium bicarbonate, trisma base, toluene, ethanol (HPLC grade), acetonitrile (HPLC grade), methanol (HPLC grade), Trifluoroacetic acid, Angiotensin II, Des-Arg-Bradykinin, Substance P, ACTH 18-39, ACTH 7-38 and bovine Insulin were obtained from Sigma-Aldrich and used as described. Trypsin was obtained from Promega.

### **Animal Surgery and Treatment.**

Wistar male rats (180–200 g) were anesthetized with pentobarbital (50 mg/kg, i.p.) and infused over 8 min with 6-OHDA (8 mg in 4 ml of 0.05% ascorbic acid in saline) at coordinates A 5 23.8 mm, L 5 1.5 mm, H 5 28.5 mm<sup>11</sup>.

Leech embryos were obtained from a laboratory colony. Prior to dissection, they were anesthetized in 8% Ethanol Ringer's, then opened dorsally and pinned flat in a Sylgard-covered dish. After removal of the yolk, the embryos were briefly submerged in Methanol, transferred to metal-coated (ITO) slides and gently dried on a hot plate at ~45°C.

### **Tissue fixation**

Rats were sacrificed 4h after the last drug injection. Their brains were removed and fixed for 40 min at 48°C in 4% paraformaldehyde. The accuracy of the 6-OHDA lesion was checked by immunocytochemistry with an anti-tyrosine hydroxylase antibody according to a previously published procedure<sup>11</sup>. These brain tissue blocks were then stored in a box at room temperature 12 years before being used in this experimental plan procedure.

## **Tissue dewaxing**

Tissue sections 10 µm thick were obtained using a microtome (E. Leitz, Westlarch, Germany) and mounted on ITO (Indium Teen Oxided) one side coated conductive glass slides. Paraffin was removed by 2 baths of 5 minutes in toluene and lightly rehydrated with graded ethanol (100°, 96°, 70° and 30°) baths before drying at room temperature.

## ***In situ* trypsin digestion**

**For MALDI direct analysis**, 2µL of enzyme (trypsin, 0.033µg/µL in water) was added at different spots on the tissue to obtain representative protein/peptide profiles. The digest was performed at room temperature, with the tissue covered with a cap to decrease evaporation. Every 10 minutes, enzyme solution was added on the same spots. Digestion was stopped by rinsing the tissue with 80% cold ethanol (-20°C) to remove salts and the sections were allowed to dry at room temperature. 30µL of matrix solution (HCCA 20 mg in ACN:H<sub>2</sub>O TFA 0.1% (7:3, v/v) was then applied evenly over the whole tissue sections using a micropipette and allowed to dry at room temperature.

**For MALDI Imaging**, trypsin deposition (20 µg/mL in water) was performed using an automatic vibrational vaporization system (ImagePrep, Bruker Daltonics, Bremen, Germany). The quantity of trypsin was set, after optimization, to 1 mL over 30 cycles in order to cover the whole surface of the tissue. The system settings were: Spray power 30%, Spray modulation 40%, time spray on 2s, incubation time: 30s and dry time: 30s. The matrix preparation was described previously (ref). Briefly, just prior to use, 4.8 uL of Aniline (1 eq.) was added to the matrix solution (HCCA 10mg/mL in ACN/TFA 0.1%, 6:4, v/v) and sonicated for 2 minutes.

## **Mass Spectrometry**

**MALDI MS direct analysis of tissue sections after *in situ* automatic trypsin digestion** Trypsin fingerprint experiments on FFPE tissue sections after *in situ* digestion of the whole tissue section were performed on an Ultraflex II TOF-TOF instrument (Bruker Daltonics, Bremen, Germany) equipped with LIFT III cell and a Smartbeam laser with a repetition rate up to 200 Hz. For MS/MS experiments, parameters were set as follow: laser repetition rate was 100 Hz with 33% attenuation, ion source voltages were respectively 8 kV and 7.3 kV on a MALDI sample plate and 1<sup>st</sup> electrode; LIFT cell was pulse from ground for electrode 1 and 2 to 19 kV and in the last step electrode 3 was decrease to 3.2 kV; reflector end voltage was set to 29.5 kV and mid-grid to 13.85 kV. Trypsin fingerprint protein identification in databanks were performed using the Biotools 3.0 software (Bruker Daltonics, Bremmen, Germany) connected to Mascot search engine and interrogating the NCBI, Swissprot and ESTs databanks with Oxydation as Variable modification, Monoisotopic as

mass values, a peptide mass tolerance of  $\pm 0.1$  Da with a peptide charge state of 1, a number of maximum missed cleavages of 2 and number of queries of 300.

### **MALDI MSI**

MALDI-MSI of rat brain tissue sections was performed on an Ultraflex II TOF-TOF (Bruker Daltonics, Bremen, DE). After dewaxing, images were obtained in positive reflector mode after trypsin digestion and matrix deposition 1 mL of the mix was applied onto the tissue using the ImagePrep sprayer as explained above and dried at room temperature. Acquisition was performed using a Smartbeam laser, with a repetition rate of 100 Hz and a spatial resolution of 100x100  $\mu\text{m}$ . Images were reconstructed using either MITICS or FlexImaging v2.0 software (Bruker Daltonics, Bremen, Germany). Average mass spectra were recorded in positive mode for each of 12,919 spots, covering the whole tissue section with an average of 500 laser shots per position.

MALDI-MSI was also performed on several dissected leech embryos (see above) at the E12 stage of development, mounted either with the inside of the embryo or the epidermis facing the laser beam. In one case the endodermal layer was removed during dissection. For reference, histological images were recorded prior to covering the tissues with a layer of ionic matrix.

## **RESULTS AND DISCUSSION**

### **Software development**

The new MITICS MS imaging software is divided in two parts, namely “MITICS Control” and “MITICS Imaging” (Figure 1). MITICS Control is dedicated to data acquisition on the instrument and thus requires that the instruments acquisition software be run in parallel, while MITICS Imaging is dedicated to post-acquisition data processing and image reconstruction. MITICS Control has been programmed in C++ for faster execution, whereas MITICS Imaging has been coded with IDL (Interactive Data Language) v 6.0 (ITT VIS, Paris, France), since IDL offers a powerful computing environment for data analysis and visualization. MITICS needs the IDL Virtual machine and the IDL Dataminer module to run. A temporary licence to test it can be obtained by emailing RSI support.

### **Data Acquisition using MITICS: MITICS Control**

Almost all MALDI instruments now provide an automatic acquisition mode for high throughput analysis of samples in standard proteomics workflows. In the automatic acquisition mode, the software uses a table that defines the positions of the spots to be analyzed. MITICS provides a simple and easy to use interface with the automatic acquisition software of the instrument for defining the image raster and the acquisition parameters. For MALDI-TOF Voyager series Applied Biosystems instruments, MITICS provides to the acquisition software (Voyager Instrument Control Panel) all parameters that are required in the MALDI-IMS sequence. For this purpose, the interface creates three files: .bic, .plt and .seq that are given to the instrument for data

acquisition. The .bic file of MITICS Control substitutes itself for the acquisition parameters normally administered by the Voyager Instrument acquisition software. The analysis parameters *i.e.* laser intensity, number of shots to be acquired per position, acceleration voltage on target and extraction grids, guide wire voltage, extraction delay time, mass scale, and low mass threshold are define and registered in the .bic file at the tab “BIC file” of the graphic interface (GUI). By contrast, the .plt and .sql files replace the instrument software for adjustment automatic mode values, *i.e.* the definition of the spots that will be sampled by the laser. Based on user-provided information, the interface calculates a table that is stored in the .plt files and used by the instrument during the automatic acquisition. For raster creation, the user defines Start and End coordinates and either a step size between spots or the number of spots per line or per column. The software then calculates automatically either the number of points or the step size, respectively. The X and Y dimensions have been decoupled, so that interval size can be set to a different value for each dimension. The third file (.sql) contains all the file information used by MITICS Control for data acquisition, namely the name and the location of the .bic and .plt files, and the recorded data files. MITICS Control also allows the application of post-acquisition macros to the recorded data. For example, the user can define an automatic baseline correction to be performed on each spectrum, or apply a smoothing or noise removal function to the data. It is also possible to enter the name of the suitable function in the text area provided for defining such manipulations prior to acquiring the spectra. A similar interface is currently under development for Bruker Daltonics MALDI instruments.

### **Data Processing and Image reconstruction: MITICS Imaging**

Both Biomap and FlexImaging softwares provide sets of analytical tools for image reconstruction and data or image processing, including display an image of an ion of interest, review of spectra behind each pixel of an image, definition of regions of interest (ROI) for comparison and statistical analysis, or co-registration of images with tissue section pictures. However, the first step for an external reconstruction tool is to import and convert the data collected during the IMS experiments. Due to the differences between the formats of the recorded data depending on the instrument, different GUIs are required. For MITICS, in order to speed up the calculation and optimize the browsing of the spectra, we have chosen to import the data files from the instruments, convert them (if necessary) to XML file format, use XML parsing to collect data in the files and store them in a databank that is written in IDL. Spectra in XML format files are easily obtained thanks to existing command lines that run in FlexAnalysis (SaxBasic coding) for Bruker Daltonics and in DataExplorer (Visual Basic) for Applied Biosystems devices. MySQL has been chosen for the database mainly because of its accuracy but also for its speed and ease of use, and because it is linked to IDL through IDL Data Miner Open Database Connectivity interface (ODBC).

The initial loading of IMS data requires XML files to be written and data to be transferred to the database and stored. At the second loading of the data, the data already stored in the

database are extracted, which increase the speed of processing. This allows the user to store data in order to rebuild pictures later facilitates search and interrogation of the spectra and enables the delocalization of the data bank to another computer on the network. Actually, it is not necessary that the database reside on the computer where the software is running. Moreover, since it is possible to link several databases to MITICS, as long as the ODBC sources are correctly configured, the data can be recorded and stored on several computers. This is a great advantage, knowing a picture composed of about 7,000 points with a mass range from 600 to 24000 Da (a whole acquired data folder of 4,33 Go) requires a database size of ~1.2GB. Conversely, if MITICS is running on several computers on a network, several pictures can be reconstructed at the same time from a single data set of the database, allowing for high throughput data processing and image reconstruction. Because MALDI IMS data contains up to hundreds of different peaks, *i.e.*, potential images, this can be a great advantage.

#### *Xml files*

MITICS is designed to operate on .xml files, but in some instruments the XML files may not be automatically created during spectra acquisition. To solve this potential difficulty, a macro can be added to the pre-acquisition part of MITICS.. This function will be executed for every spectrum during the data acquisition, so as to get all the files available in XML as soon as the acquisition is completed.

If a picture has to be created from spectra previously acquired, XML files can still be produced by running a batch process in *FlexAnalysis* for an analysis realised on a Bruker Daltonics device, or by executing a visual basic macro in *DataExplorer* for an analysis performed on an Applied Biosystems instrument. In MITICS macro is available that allows the selection of an appropriate macro and its application to all the files of a folder in order to transform .dat file to .xml files.

#### *Processing time*

Tests were performed on a computer running Windows XP with an Intel Pentium 4 (3.2GHz) processor and 1.00Go of RAM. In these conditions, it took 45 min to first load and store in the database a 16000 point IMS data set. Then, reloading of the data directly from the databank took less than 1 min. Image reconstruction from firstly opened or reloaded data took also less than 1 min.

#### *Software Handling*

At the opening of the window, the user has three choices (Figure 1):

- To load data in the databank from an analysis folder either from an Applied Biosystems, or a Bruker Daltonics MS system running a complete analysis from collect of the spectra files in the folder where they have been recorded during acquisition, to the image reconstruction

- To use data already loaded in the databank and build the picture faster
- To change the parameters of the databank.

#### *Data loading and databank storage*

When data are first loaded on MITICS, the user has as first choice to define the MS system used for acquisition, e.g., an Applied Voyager DE-STR MALDI TOF button or a Bruker UltraFlex MALDI TOF-TOF button. For data originating from MITICS control, the .plt file (containing the coordinates of the picture's pixels) has to be provided. For all data, the location of the IMS data must be given. The user can browse the computer disks in order to choose the appropriate file. Collect of the data is run by clicking on the Accept button. If the user forgets to provide one of text areas, an error message appears, asking to enter a name for this feature. While processing the data, a continuously updated message appears at the bottom of the window to inform the user about the progress in the analysis (Figure 2). At the end of processing, a new window appears asking the user to define the parameters for image reconstruction.

If the user chooses to create a picture from data already loaded in the databank, he can click on the "See already loaded data" button. Then a new window appears showing the list of all the available projects in a list box. After he has chosen one of the names displayed in the list, he clicks on the "OPEN" button to run data extraction. Similarly as before, at the end of the processing a new window allowing the image reconstruction parameters to be defined appears.

The last part of the software consists of the configuration of the databank used to load the data. By clicking on the "Change BD parameters" button on the first window, the user will come to a new window where he has to choose between the different options shown. The software checks automatically on the computer where the available ODBC sources are located in the Administration Tools of the Control Panel. After entering the data source name, the user name and the password required, the user clicks on the "Use these parameters" button to apply this configuration to the databank. In this manner, when a new project is created, collected spectral data will be loaded into the right database.

#### *Image Reconstruction*

Figure 3 presents the user interface of MITICS for image reconstruction. Several parameters can be changed to define the image reconstruction criteria. First, the user can either create an image by measuring the peak maximum intensity. Because on MALDI instruments and especially for TOF analyzers the resolution is not constant over the whole mass range (in TOF peaks become broader with increasing  $m/z$ ), the maximum peak intensity does not accurately describe the data. For this reason, measurement of the peak area instead of the maximum intensity can be displayed in MITICS Imaging.

MITICS can also display different overviews of the data. In general, IMS reconstruction software packages offer one overview of the data, such as representative spectra for selected

individual spots or the average spectrum calculated from all recorded mass spectra, both of which have shortcomings. For example, it is difficult if not impossible to examine thousands of single spectra serially and get a general sense of the information content of the data set, and the average spectrum tends to highlight abundant species while rendering those more rare relatively undetectable. Therefore, we have implemented in MITICS several overviews of the collected data in MITICS that include (see Figure 3) the average spectrum (bottom representation), a spectrum corresponding to the sum of all spectra (middle) and a spectrum displaying the maximum value found in the data for defined  $m/z$  (top). The user can use each of this representation to define  $m/z$  values of interest. A “click” in the graph allows zooming on the spectra by opening a small window which offers many tools to get a more precise visualization. For image reconstruction, the user must provide an  $m/z$  value within a margin error. Threshold values (minimum and maximum) can also be defined for taking into account signal variability (according to the position on the tissue). Minimum intensity threshold allows for defining the minimum value for intensity or area to consider that the signal is a peak. If the value is under the threshold then the value is set to 0 (no signal). The maximum value algorithm considers signal variability from one spot to another and decides that above a certain threshold values will be set to 100% intensity. After parameter adjustments, clicking the “add” button includes the selected ion in a list of ions that have been analyzed. Regarding the error margin, it can be defined either in percent of the total mass range or with an absolute value to add and subtract from the  $m/z$  value to obtain the boundaries of the accepted interval. Finally, the pictures will be created by clicking on the “Create image” button. Once the text fields have been filled, an information message appears on the bottom of the window displaying the processing progress. In the case of a null query without results, an error message informs the user that he has to try some other values.

Initially, the developed pictures are shown in black and white. To visualize it in a colour-coded intensity mode, the user has just to click on the “Edit palette” button and to choose which features are wanted. Some colour ranges are predefined and available by clicking on the “Load predefined” button. On the picture displayed on the Figure 4 the “BLUE/WHITE” palette has been chosen. After image reconstruction, the software allows defining regions of interest, moving or turning the picture, resizing it, or adding some geometrical shapes on it. It is also possible to save the image or print it, or to open a saved file or a new empty one. Some copy and/or paste or zoom functions are available as well. Another picture can be opened in the same window, making comparisons easier to do and some operations are available to work on the image, like filters for example, which permits getting a smoother picture compared to the initial simple hard pixel display. It is possible, moreover, to mark the results with a wide range of colours and gradations of basic tints or pre-specified multichromatic sets which are adaptable according to the wanted intensities. Lastly, the picture can be reprocessed after it has been displayed without the need to initiate another computation from the beginning - for example, maximum and minimum thresholds can be

changed easily thanks to the Window Level Control in the Image Panel on the right of the main picture window.

## **Application to Parkinson's model animals**

### *Biomarkers of Parkinson's model animals*

As previously demonstrated<sup>12, 13</sup>, brain FFPE tissue blocks of ten 6-OHDA treated *versus* ten untreated rats were systematically studied and compared for biomarker hunting using trypsin digestion followed by nanoLC-ion-trap identification. Several biomarkers have been identified, including Ubiquitin, alpha enolase, Pep 19, CRMP2, peroxidoxin 2, profiling, ferritin heavy chain, HSP27, apolipoprotein E, and vimentin. These molecules have been previously identified as up-regulated proteins using classical proteomics studies<sup>14</sup> or in direct tissue analyses<sup>15, 16</sup>. Based on such results, we decided to perform *in silico* digestion of these biomarkers and try to find peptide mass fingerprint (PMF) on trypsin digested MALDI spectra (Table 2) Based on these specific biomarker PMFs, we tried to assess the efficiency of MITICS software for creating specific molecular images. Molecular images based on ubiquitin or apolipoprotein E were reconstructed using MITICS and are presented in Figures 5 and 6.

### *Reconstruction of images*

Visual comparison between MITICS and FlexImaging was performed on images apolipoprotein digestion fragments (Figure 6). All images obtained by FlexImaging without normalization are less defined, less contrasted, and less informative compared to those obtained with MITICS (Figure 6). It is worth noting that when spectral normalization is applied in Fleximaging, the images are much better in term of brightness, contrast and background suppression, but they are also more out of focus. A problem with commercial and proprietary softwares like FlexImaging is the fact that the specifics of the statistical and computational operations performed on the images are not made available to the user, hence it is difficult to determine how successful it is in predicting the localization of the biomolecules of interest. This is one major critical point of MALDI MSI, the delocalization of molecules of interest after adding the matrix. This is why several new methods and instruments were developed, including the automatic sprayer<sup>17</sup>, the spray droplet method<sup>18</sup>, and the oscillating capillary nebulizer<sup>19</sup> for spraying the matrix, as well as a robot<sup>12</sup> or a printer<sup>20</sup> for printing the matrix.

If instead of using black/white images shading pixels from the most predominant peptide to the least gives a vagueless image. Comparison of the zoom of specific region between MITICS and Fleximaging confirms such conclusion (Figure 6). Thus, in this context, immunocytochemistry experiments are needed to confirm the exact localization. The good point with MITICS, because no treatments are applied to the images, the image corresponds to the exact position of the biomolecule. It is also of interest to point out that in these series of images reconstructed with MITICS and FlexImaging, ubiquitin and apolipoprotein E are localized in the same brain area which is in line with previous results<sup>14-16</sup>. However, this localization is significantly different when

compared to the data presented in control animals (data from Allen Brain Atlas <http://www.brain-map.org>).

Another important point for image reconstruction is which approach gives better results, using the peak intensity or the peak area. As can be seen in Figure 8, the better resolution is obtained when peak areas are used to create molecular images. We came to a similar conclusion when the experiments were conducted on leech embryo tissues. Molecular images of different  $m/z$  values were reconstructed after data acquisition for each embryo. MSI images, superposed on the histological images, are shown for ions at  $m/z$  3507 and  $m/z$  6430 (Figure 8). For example, for the two embryos imaged from the internal surface, the ion  $m/z$  3507 was exclusively localized in the nervous system, in ganglia, interganglionic connective nerves and lateral roots, proximal to the ganglia. By contrast, the ion  $m/z$  6430 was not detected in the nerve chord but in nephridial structures, probably around the nephridiopores. The composite image illustrates very well this difference of localization between the two molecules. In comparison, for the embryo positioned with the epidermis towards the laser beam (right side in Figure 8), no significant signal was observed for the  $m/z$  3507 ion, confirming its localization to the internal CNS, while a reduced but detectable signal for the  $m/z$  6430 ion was observed at the nephridiopores, which travel through the body wall and open to the outside in the epidermis. These very preliminary results demonstrate the possibility of using MSI to obtain whole embryo molecular maps, with reasonable resolution; though a careful exploration of methods for obtaining depth information will be necessary. An interesting point is that the 6430 Da peptide corresponds to an antimicrobial peptide we recently purified and localized in brain and in the body <sup>21</sup>. Thus these molecular MALDI images are completely in line with the immunocytochemical and biochemical data.

## CONCLUSION

MITICS is novel integrated software for IMS that is designed for compatibility with all MALDI instruments. MITICS provide for the moment completely developed solutions for processing data from Applied Biosystems and Bruker Daltonics MALDI-TOF instruments, but it is aimed to evolve to be compatible with SCIEX or Shimadzu or even other instruments. MITICS includes two parts, namely MITICS Control for data acquisition on the instrument and MITICS Imaging for data processing and image reconstruction. Data are processed following XML conversion and data extraction using an XML parser before storage in a databank. Addition of databanks to the software allows for simultaneous reconstruction of numerous images in the goal of high throughput imaging to be developed. MITICS provides different views of data than other available software, namely different spectral representations for ions of interest, easier selection from the spectra, and offers the flexibility to use peak area instead of maximum intensity as the measurement basis for image reconstruction, which has proved to be better in describing mass spectra by comparison with other IMS software packages. Finally, MITICS is universal software, free of charge, easily

modified for integration of new tools and is available on request on our web site (<http://www.maldi-imaging.com>).

## ACKNOWLEDGEMENTS

Supported by grants from Centre National de la Recherche Scientifique (CNRS), Ministère de L'Education Nationale, de L'Enseignement Supérieur et de la Recherche (ACI Jeunes Chercheurs ACI JC4074 to I. Fournier), ANR PCV (to I. Fournier). Also supported by collaboration Bruker Daltonics and the laboratory for the MALDI Imaging technology development. The authors would also like to thank A. Page and C. Rolando of the Proteomic Platform of USTL.

## REFERENCES

1. Fournier, I.; Day, R.; Salzet, M., Direct analysis of neuropeptides by in situ MALDI-TOF mass spectrometry in the rat brain. *Neuro Endocrinol Lett* **2003**, 24, (1-2), 9-14.
2. Li, L.; Garden, R. W.; Romanova, E. V.; Sweedler, J. V., In situ sequencing of peptides from biological tissues and single cells using MALDI-PSD/CID analysis. *Anal Chem* **1999**, 71, (24), 5451-8.
3. Chaurand, P.; Stoeckli, M.; Caprioli, R. M., Direct profiling of proteins in biological tissue sections by MALDI mass spectrometry. *Anal Chem* **1999**, 71, (23), 5263-70.
4. Redeker, V.; Toullec, J. Y.; Vinh, J.; Rossier, J.; Soyeux, D., Combination of peptide profiling by matrix-assisted laser desorption/ionization time-of-flight mass spectrometry and immunodetection on single glands or cells. *Anal Chem* **1998**, 70, (9), 1805-11.
5. Jimenez, C. R.; Li, K. W.; Dreisewerd, K.; Spijker, S.; Kingston, R.; Bateman, R. H.; Burlingame, A. L.; Smit, A. B.; van Minnen, J.; Geraerts, W. P., Direct mass spectrometric peptide profiling and sequencing of single neurons reveals differential peptide patterns in a small neuronal network. *Biochemistry* **1998**, 37, (7), 2070-6.
6. Dreisewerd, K. K., R.; Geraerts, W. P. M.; Li, K. W., Direct mass spectrometric peptide profiling and sequencing of nervous tissues to identify peptides involved in male copulatory behavior in *Lymnaea stagnalis*. *International Journal of Mass Spectrometry and Ion Processes* **1997**, 169/170, 291-299.
7. Li, K. W.; Hoek, R. M.; Smith, F.; Jimenez, C. R.; van der Schors, R. C.; van Veelen, P. A.; Chen, S.; van der Greef, J.; Parish, D. C.; Benjamin, P. R.; et al., Direct peptide profiling by mass spectrometry of single identified neurons reveals complex neuropeptide-processing pattern. *J Biol Chem* **1994**, 269, (48), 30288-92.
8. Stoeckli, M.; Farmer, T. B.; Caprioli, R. M., Automated mass spectrometry imaging with a matrix-assisted laser desorption ionization time-of-flight instrument. *J Am Soc Mass Spectrom* **1999**, 10, (1), 67-71.
9. Caprioli, R. M.; Farmer, T. B.; Gile, J., Molecular imaging of biological samples: localization of peptides and proteins using MALDI-TOF MS. *Anal Chem* **1997**, 69, (23), 4751-60.

10. Clerens, S.; Ceuppens, R.; Arckens, L., CreateTarget and Analyze This!: new software assisting imaging mass spectrometry on Bruker Reflex IV and Ultraflex II instruments. *Rapid Commun Mass Spectrom* **2006**, 20, (20), 3061-6.
11. Tonnaer, J. A.; Lammers, A. J.; Wieringa, J. H.; Steinbusch, H. W., Immunohistochemical evidence for degeneration of cholinergic neurons in the forebrain of the rat following injection of AF64A-picrylsulfonate into the dorsal hippocampus. *Brain Res* **1986**, 370, (1), 200-3.
12. Stauber, J.; Lemaire, R.; Franck, J.; Bonnel, D.; Croix, D.; Day, R.; Wisztorski, M.; Fournier, I.; Salzet, M., MALDI imaging of formalin-fixed paraffin-embedded tissues: application to model animals of Parkinson disease for biomarker hunting. *J Proteome Res* **2008**, 7, (3), 969-78.
13. Lemaire, R.; Desmons, A.; Tabet, J. C.; Day, R.; Salzet, M.; Fournier, I., Direct analysis and MALDI imaging of formalin-fixed, paraffin-embedded tissue sections. *J Proteome Res* **2007**, 6, (4), 1295-305.
14. Basso, M.; Giraudo, S.; Corpillo, D.; Bergamasco, B.; Lopiano, L.; Fasano, M., Proteome analysis of human substantia nigra in Parkinson's disease. *Proteomics* **2004**, 4, (12), 3943-52.
15. Pierson, J.; Norris, J. L.; Aerni, H. R.; Svenningsson, P.; Caprioli, R. M.; Andren, P. E., Molecular profiling of experimental Parkinson's disease: direct analysis of peptides and proteins on brain tissue sections by MALDI mass spectrometry. *J Proteome Res* **2004**, 3, (2), 289-95.
16. Pierson, J.; Svenningsson, P.; Caprioli, R. M.; Andren, P. E., Increased levels of ubiquitin in the 6-OHDA-lesioned striatum of rats. *J Proteome Res* **2005**, 4, (2), 223-6.
17. Hardesty, W. M.; Caprioli, R. M., In situ molecular imaging of proteins in tissues using mass spectrometry. *Anal Bioanal Chem* **2008**.
18. Sugiura, Y.; Shimma, S.; Setou, M., Two-step matrix application technique to improve ionization efficiency for matrix-assisted laser desorption/ionization in imaging mass spectrometry. *Anal Chem* **2006**, 78, (24), 8227-35.
19. Chen, Y.; Allegood, J.; Liu, Y.; Wang, E.; Cachon-Gonzalez, B.; Cox, T. M.; Merrill, A. H., Jr.; Sullards, M. C., Imaging MALDI Mass Spectrometry Using an Oscillating Capillary Nebulizer Matrix Coating System and Its Application to Analysis of Lipids in Brain from a Mouse Model of Tay-Sachs/Sandhoff Disease. *Anal Chem* **2008**, 80, (8), 2780-2788.
20. Baluya, D. L.; Garrett, T. J.; Yost, R. A., Automated MALDI Matrix Deposition Method with Inkjet Printing for Imaging Mass Spectrometry. *Anal Chem* **2007**.
21. Schikorski, D., Cuvillier Hot, V., Leippe, M., Macagno, E., Salzet, M., Tasiemski, A., The medicinal leech as a model for studying the antimicrobial response of the central nervous system. *J. Immunol* **2008**.

## FIGURE LEGENDS

**Table 1:** Comparison between all imaging mass spectrometry software already known

**Table 2 :** Specific 6OHDA biomarkers found after trypsin digestion by comparing 6-OHDA FFPE tissues to control FFPE tissues of rat brains after nanoLC-nanoESI-IT MS followed by MALDI direct analyses of digested tissues. (in sequence box, matched peptides are indicated by a bar and are underlined in red)

**Figure 1:** Scheme of the MITICS software structure

**Figure 2:** Graphical User Interface of a new Applied Biosystems project creation, and examples of error messages and waiting messages

**Figure 3:** Graphical User Interface displaying the three reference spectra (Maxima values, sum of spectra and average of spectra) for the picture parameters definition: choice of the computing method (maximum intensity or area), mass-over-charge ratios, margin of error (absolute or in percent) and peak threshold (under which signal is considered as noise).

**Figure 4:** Example of reconstructed pictures and of different colour parameters available.

**Figure 5:** Molecular Image reconstructed with MITICS a on ubiquitin trypsin specific peptides (1063, 1527 and image based on the sum of all trypsin digested peptide). Insert pictures correspond to scan of the tissue slide before MALDI analyses, expression of ubiquitin mRNA in mouse brain (data from Allen Brain Atlas <http://www.brain-map.org>) and map of the corresponding tissue section (Bregma index).

**Figure 6:** Comparison between MITICS and FlexImaging (with or without spectra normalization) on Apolipoprotein E trypsin (ApoE) specific peptides (m/z 726). Insert pictures correspond to zoom of the same region. Intensity scale is presented in comparison as well as the scan of the brain slice before MALDI imaging.

**Figure 7 :** Comparison of the images realised with MITICS based on Apolipoprotein E trypsin (ApoE) specific peptides (m/z 726 and 1523) using either the peak area or the maximum peak intensity with a threshold of 20 and a margin of error of 2 Da. Chromatic scale is also presented.

**Figure 8 :** MALDI imaging of dissected leech embryos in whole mount.

Three stage E12 specimens were opened along the dorsal midline and the yolk removed, then pinned flat, exposed for ~2 min to methanol to harden the tissues, finally placed on metal-coated glass slides and immediately dried. MALDI images were reconstructed from the mass spectra

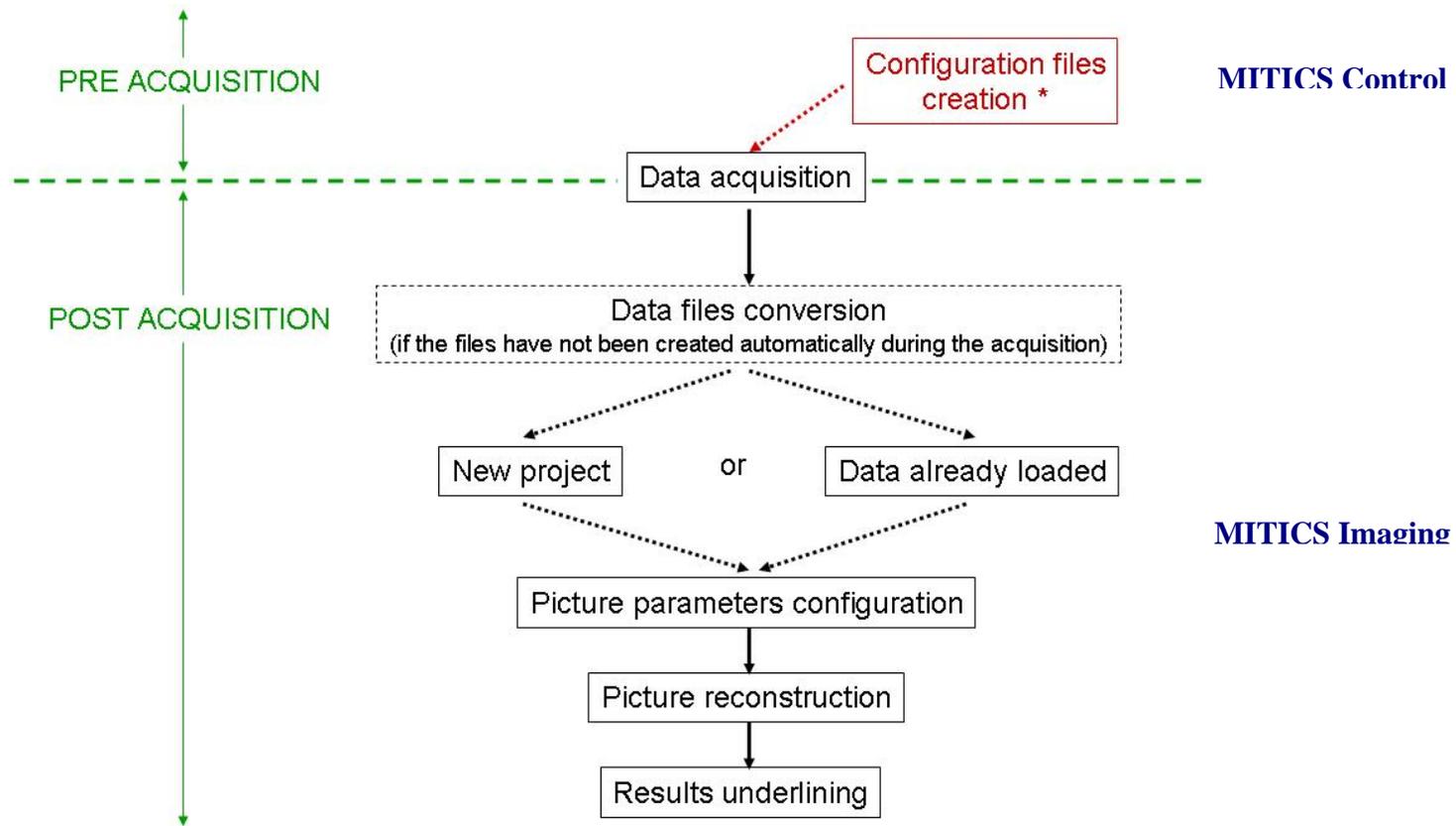
collected from 20,000 locations covering the embryos completely in a rectangular raster of points 60  $\mu\text{m}$  apart. **(a)** Transmitted light low-power micrographs of the unstained embryos (embryo dimensions  $\sim 2.5 \text{ mm} \times 10 \text{ mm}$ ) taken prior to the spraying of CHCA matrix on the embryos for MALDI analysis. The embryo at left was mounted so the internal surface (endoderm) faced the laser beam, while the embryo in the middle was similarly prepared but the endodermal layer ( $\sim 10\text{-}20\mu\text{m}$  thick) was removed. The embryo at right was mounted with the epidermis towards the laser source. Anterior is to the right in all cases. **(b)** Reconstructed MALDI images showing the distributions of ions with  $m/z$  values of 3507 (above) and 6430 (below), for the three embryos. The 3507 ions are maximally present in the segmental ganglia and interganglionic connective nerves, in the two animals analyzed with the inner surface towards the laser. It is essentially absent in the third animal, as might be expected since the CNS is close to the inner surface, indicating that the depth range into the tissue of the laser beam is less than  $\sim 100 \mu\text{m}$ . By contrast, the 6430 ion is more widely distributed but with maxima associated with the nephridiopores. Signal associated with the nephridiopores is visible in the specimen on the right column as well, presumably because these structures extend through the body wall and open to the outside. **(c)** Superposition of the images shown in (b), but here with abundance proportional to intensity in green ( $m/z = 3507$ ) and red ( $m/z = 6430$ ).

Software		Different choices for calculations	Input format	Output format	Spectrometers compatibility	Use of databank	Requirements	Maximum data points	Processing Time	Availability	F r e e	Authors	Code development
<b>MMSIT</b>	Acquisition	no	Instrument format .dat	Analyze 7.5	AB MALDI/TOF	yes	Aqiris DP211 digitizer board	unknown	Few seconds	<a href="http://www.maldi-msi.org">www.maldi-msi.org</a>	y e s	M. Stoeckli	VB6 and assembler
<b>BIOMAP</b>	Images reconstruction	no	Analyze 7.5, .img, .tiff, .dicom, .pnp	tiff, jpeg, Analyze 7.5	AB MALDI-TOF and 4700 and 4800 MALDI-TOF/TOF, MALDI Micro MX™(Waters)	yes	IDL RT 6.1 or higher	32 768 per spectrum	Few seconds	<a href="http://www.maldi-msi.org">www.maldi-msi.org</a>	y e s	M. Rausch with additions by M. Stoeckli	IDL
<b>oMALDI Server 4.0</b>	Acquisition/ Image reconstruction	no	unknown	unknown	AB/MDS SCIEX oMALDI QSTAR Hybrid LC/MS/MS	unknown	Instruments software's	unknown	unknown	AB	n o	AB/MDS SCIEX	C++
<b>4700/4800 Imaging</b>	Acquisition	no	Instrument format	Analyze 7.5	AB MALDI TOF/TOF	unknown	Instruments software's	32 768 per spectrum	-	<a href="http://www.maldi-msi.org">www.maldi-msi.org</a>	y e s	M. Stoeckli	C#
<b>Flex Imaging</b>	Acquisition/ Image reconstruction	no	Instrument format .img files	bmp, jpeg, jpg, tif, tiff	BD MALDI/TOF and MALDI-TOF/TOF	unknown	Instruments software's	unknown	Few seconds	BD	n o	BD	unknown
<b>Create Target AND Analyze This!</b>	Acquisition/ data conversion	no	binary	Analyze 7.5	BD MALDI/TOF and MALDI-TOF/TOF	no	Instruments software's and Biomap	Theoretically 1 million per picture	25mn for .img conversion	<a href="http://www.maldi-msi.org">www.maldi-msi.org</a>	y e s	C. Ceuppens et al.	Visual Basic 6
<b>Tissue view 1.0</b>	Acquisition/ Image reconstruction	no	Instrument format	unknown	AB/MDS SCIEX MALDI TOF/TOF and QSTAR	unknown	Instruments software's	unknown	unknown	AB	n o	AB/MDS SCIEX	IDL
<b>Axima2 Analyze</b>	Acquisition/ data conversion	no	Instrument format	Analyze 7.5	SZU	unknown	Windows	unknown	-	<a href="http://www.maldi-msi.org">www.maldi-msi.org</a>	y e s	M. Setou et al	unknown
<b>MITICS</b>	<b>Acquisition/ Image reconstruction</b>	<b>Maximum intensity or Area</b>	<b>.xml</b>	<b>bmp, jpg, jp2, gif, pict, png, tif</b>	<b>AB MALDI/TOF BD MALDI/TOF TOF/TOF</b>	<b>Yes, MySQL</b>	<b>Instruments software's, IDL VN and Data miner</b>	Theoretically 1 million per picture	<b>45 min/ 16 000 points</b>	<b>Upon email</b>	<b>y e s</b>	<b>O. Jardin-Mathé et al.</b>	<b>C++, IDL, Visual Basic 6</b>

Table 1

Name Mass (Da)	Sequence Coverage	Peptides Match	Digestion Peptide	
			[M+H] <sup>+</sup> m/z	Sequences
Ubiquitin M=8 559.62 Da	43.4%	<p>10 20 30 40 50 60 70 80</p> <p>MQIFVKLTG KTITLEVEPS DTIENVKAKI <u>QDKEGIPPDQ</u> QRLIFAGKQL EDGR<u>TLSDYN</u> IQKESTLHLV LRLRGG</p>	1067.62 1081.56 1336.67 1523.84	ESTLHLVLR TLSDYNIQK ESTLHLVLR IQDKEGIPPDQQR
α-Enolase M=47 098.24 Da	16.8%	<p>10 20 30 40 50 60 70 80 90 100 110</p> <p>MSILKIHARE <u>IFDSR</u>NETV EVDLYTAGL FRAAVPSGAS TGIYEALRLR DNDKTFMGK GVSKAVERHN <u>KTIAPALVSK</u> KLNVEQEKI <u>DQLMIEMDGT</u> ENHSKFGANA</p> <p>120 130 140 150 160 170 180 190 200 210 220</p> <p>ILGVSLAVCK AGAVERGVPL YRHIADLAGN PEVILPVPAF NVINGGSHAG NKLAMQEFMI LPVGASSFRE <u>AHRIGAEVYH</u> NLHNVIKKEY GKDATNVGDE GGFAPNILEN</p> <p>230 240 250 260 270 280 290 300 310 320 330</p> <p>KEALELLKSA IAKAGYTDQV VIGMDVAASE FYRAGKYDLD FKSPDDASRY ITPDQLADLY KSFIKDYPVV SIEDPFDQDD WDAWQKFTAT AGIQVDDDL TVTNPKR<u>IAK</u></p> <p>340 350 360 370 380 390 400 410 420 430 440</p> <p><u>AAGEKSNCL</u> LLKVNQIGSV TESLQACKLA QSNQGVWVMS HRSGETEDTF IADLVVGLCT GQIK<u>TGAPCR</u> <u>SERLAKYNIQI</u> LRIFEELGSK AKFAGRS<u>FRN</u> <u>PLAK</u></p>	591.08 766.46 786.47 932.53 976.49 1143.64 1636.85 2153.16	MSILK EIFDSR IAKAAGEK SFRNPLAK TGAPCRSER IGAEVYHNLK IDQLMIEMDGTENK EIFDSRGNPTVEVDLYTAK
PEP 19 M=6 803.29 Da	37.1%	<p>10 20 30 40 50 60 70</p> <p><u>MSERQSAGAT</u> NGKDKTSGDN <u>DGQKKVQEEF</u> DIDMDAPETE RAAVAIQSQF RKFQKKKAGS QS</p>	1049.53 1336.67	TSGDNDGQKK MSERQSAGATNGK
CRMP2 M=62 238.59 Da	15.0%	<p>10 20 30 40 50 60 70 80 90 100 110</p> <p>MSYQKKNIP <u>RITSDR</u>LLIK GGHVNDQDS FYADIYMEDG LIKIQIENLI VPGGVKTEA HSRNVIPGGI DVHTRFQMPD QGMSADFF QCTKAALAGG TTHIDHVPV</p> <p>120 130 140 150 160 170 180 190 200 210 220</p> <p>EPGTSLLAAF DQWREWADSK SCCDYSLVHD ITEWHKIQE EMEALVKDHG VNSFLVYMAF KDRFLQTSQ IYEVLSVIRD IGATAQVHAE NGDIIAEFQQ RILDLGITGP</p> <p>230 240 250 260 270 280 290 300 310 320 330</p> <p>EGHVLSPREE VEAEAV<u>NRSI</u> TIANQVNCPL YVTK<u>VMSKSA</u> <u>AEVIAQARKK</u> GTVVYGEPII ASLGTDSGHY WSKNNAKAAA FVTSPPLSPD PTPDFLNSL LSCGDLQVTC</p> <p>340 350 360 370 380 390 400 410 420 430 440</p> <p>SAHCTFNQA KAVGK<u>NPFL</u> IPEGT<u>NGTEE</u> RMSVIUDKAV VTGKMDENQF VAVTSTNAAK VFNLVPRKR ISVGSADLV IWDPSVKTI SAKTHMSALE YNIFEGMECR</p> <p>450 460 470 480 490 500 510 520 530 540 550</p> <p>GSPLVVISGG KIVLEDGTLH VTEGSGRYIP RKPPDFVYK RIKARS<u>PLAE</u> <u>LRGVPRLYD</u> <u>GPVCEVSVTP</u> <u>KTVTPASSAK</u> TSPAQQAPP VRNLHQS<u>GF</u> <u>LSGAQIDDI</u></p> <p>560 570 580</p> <p>PRRTTORIVA <u>PPGORANITS</u> LG</p>	591.08 766.46 841.52 844.48 1143.64 1460.79 2169.17 2406.33	ITSDR IVAPPGGR MSYQKKK SRLAELR SAAEVIAQARK VMSKSAAEVIAQAR NLHQSGFSLSGAQIDDNIPR GLYDGPVCEVSVTPKTVTPASSAK
Peroxidoxin M=21 770.06 Da	11.1%	<p>10 20 30 40 50 60 70 80 90 100 110</p> <p>MASGNAHIGK PAFDPTGTAV VDGAFKEIKL SDYRGYVVL FFYPLDFTTV CPTETIAFSD HAEDFRKLG EVLGVSVDQ FTHLAWINTP RKEGGLGPLN IPLLADVTKS</p> <p>120 130 140 150 160 170 180 190 200</p> <p>LSQNYGVLEN DEGIAYRGLF IIDAK<u>GVLRQ</u> <u>ITVNDLPVGR</u> <u>SVDEALR</u>LVQ AFQYTDHGE VCPAGWPGS DTIKPNVDS KEYFSKHN</p>	788.47 1636.85	SVDEALR GVLRLQITVNDLPVGR
Profilin M=14 977.52 Da	40.7%	<p>10 20 30 40 50 60 70 80 90 100 110</p> <p>MAGWNAIDS <u>LMADGTCQDA</u> <u>AIVGYRDSFS</u> <u>VMAAVPGKTF</u> VSITPAEVGV LVGRDRSFF VNGLTGGQK CSVIRDSLLQ DGEFTMDLRT KSTGGAPTFN <u>VTVTMTARTL</u></p> <p>120 130 140 150</p> <p>VLLMGKESVH <u>GGLINKKYE</u> <u>MASHLPSQY</u></p>	1052.62 1109.70 3959.72	ESVHGGLINK CYEMASHLR MAGWNAIDSLMADGTCQDA AIVGYKDSFS VMAAVPGK
Ferritin Heavy Chain M=20 982.27 Da	26.0%	<p>10 20 30 40 50 60 70 80 90 100 110</p> <p>TTASPSQVRQ <u>NYHQDSEAAI</u> NRQINLELYA SYVYLSMSCY <u>FDRDDVALKN</u> FAKYFLGSH EEREHAEMKL KLQNGRGGRI FLQDIKKPR DDWESGLNAM RCALHLEKSV</p> <p>120 130 140 150 160 170 180 190</p> <p><u>NQSLLEHL</u> ATDKNDPHLC DPIETHYLINE QVKS<u>IKELGD</u> <u>HVTNLR</u>KMGA PESGAEYLF DKHTLGHQDE S</p>	1481.8 4105.41	SIKELGDHVTNLR QNYHQDSEAAINRQINLELYASYVYLSMSCY FDR
Apolipoprotein E M=35 731.33 Da	25.3%	<p>10 20 30 40 50 60 70 80 90 100 110</p> <p>MKALWALLV PLLTGCLAEQ ELEVTDLPC QSDQWEQAL NRFWDYLRW QFLSDQVQEE LQSSQVTQEL TVLMEDTME VKAYKKELEE QLGPVAEETR <u>ARLAKEVQAA</u></p> <p>120 130 140 150 160 170 180 190 200 210 220</p> <p><u>QARLGADHED</u> LRNLGQYRN EVNTLQGSRT EELRS<u>LSTH</u> <u>LRKMRRLNR</u> DADDLQRLA <u>VYRAGAGEGA</u> <u>ERGVSAIRER</u> <u>LGPLVEGGRQ</u> <u>RTANLGAAGA</u> <u>QPLRDRQAAL</u></p> <p>230 240 250 260 270 280 290 300 310 320</p> <p>SDRIRGLEE VGNQARDLE EVRQMEVVR SK<u>HEEQTOQI</u> <u>RLQAEIFQAR</u> <u>IKGWFEPLVE</u> <u>DMQRQWANLM</u> EKQASVATH SIASTVPLE NQ</p>	726.23 888.44 968.53 1162.14 1183.80 1238.70 1505.86 1523.84	LSTHLR GVSAIRER LGPLVEQGR MEEQTOQIR LAKEVQAAQAR TANLGAQAQLRGWFEPLVEDMQR GWFEPLVEDMQR QRTANLGAQAQLR
HSP27 M=22 807.57 Da	20.5%	<p>10 20 30 40 50 60 70 80 90 100 110</p> <p>NTERR<u>VPPSL</u> <u>LESFSPWEFPR</u> DWYPAHSRLF DQAFGVPRFP DEWSQWFSSA GWPQVVRPLP AATAEGPAV TLARPPSRAL NRQLSSGVSE <u>IRQTADRWRV</u> SLDVNHFAPF</p> <p>120 130 140 150 160 170 180 190 200 210</p> <p>ELTVKTR<u>EGV</u> <u>VEITGKHEER</u> QDEHGYSRC FTRKYLPPG VDPYLVSSSL SPEGTLTVEA PLPKAVTQSA EITIPVTFEA <u>RAQIGGPESE</u> <u>QSGAK</u></p>	591.080 932.532 988.549 1358.716 1481.856	QTADR QTADRWR RVPPSLLR AQIGGPESESGAK EGVVEITGKHEER

Table 2



\* Only for Applied Biosystems spectrometers

Figure 1



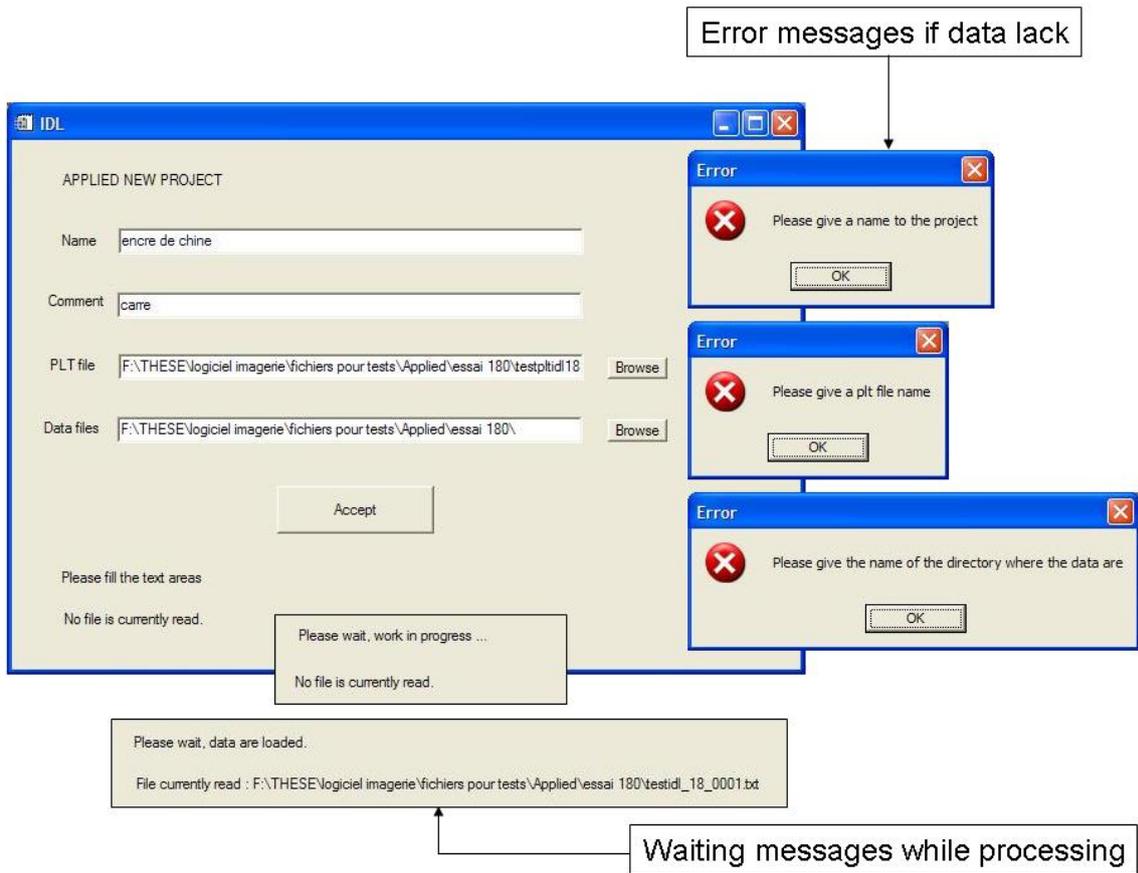


Figure 2

Figure 3

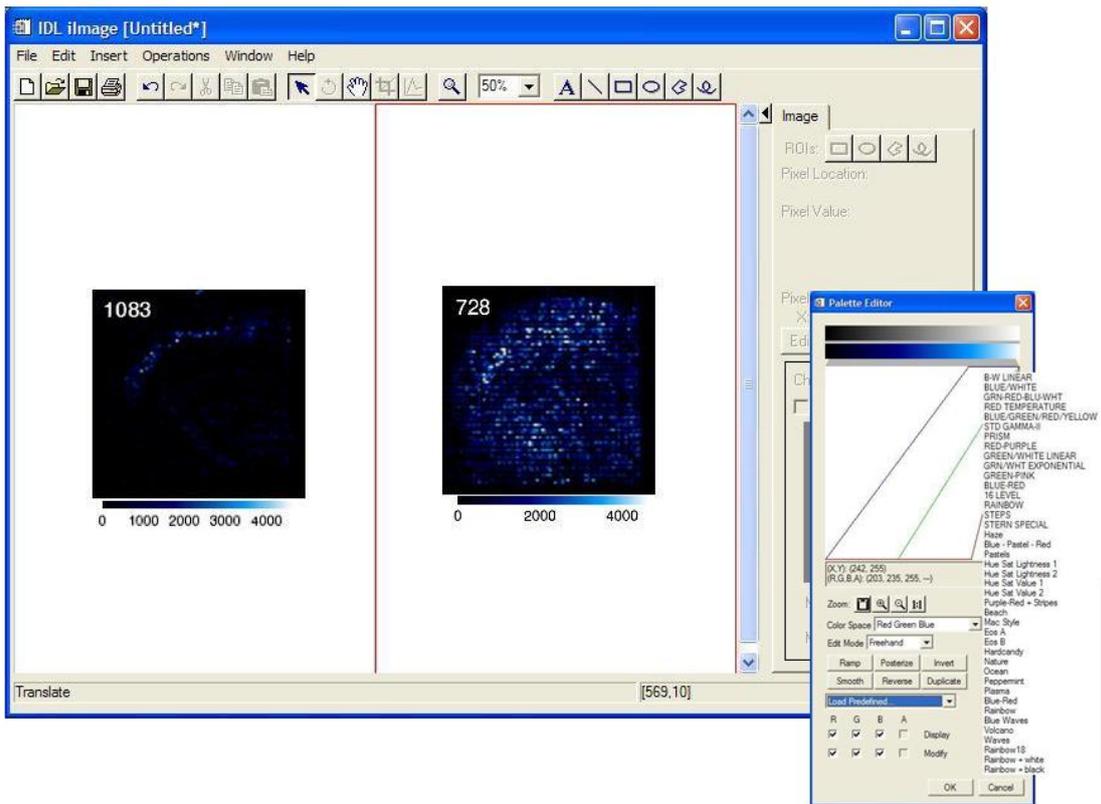
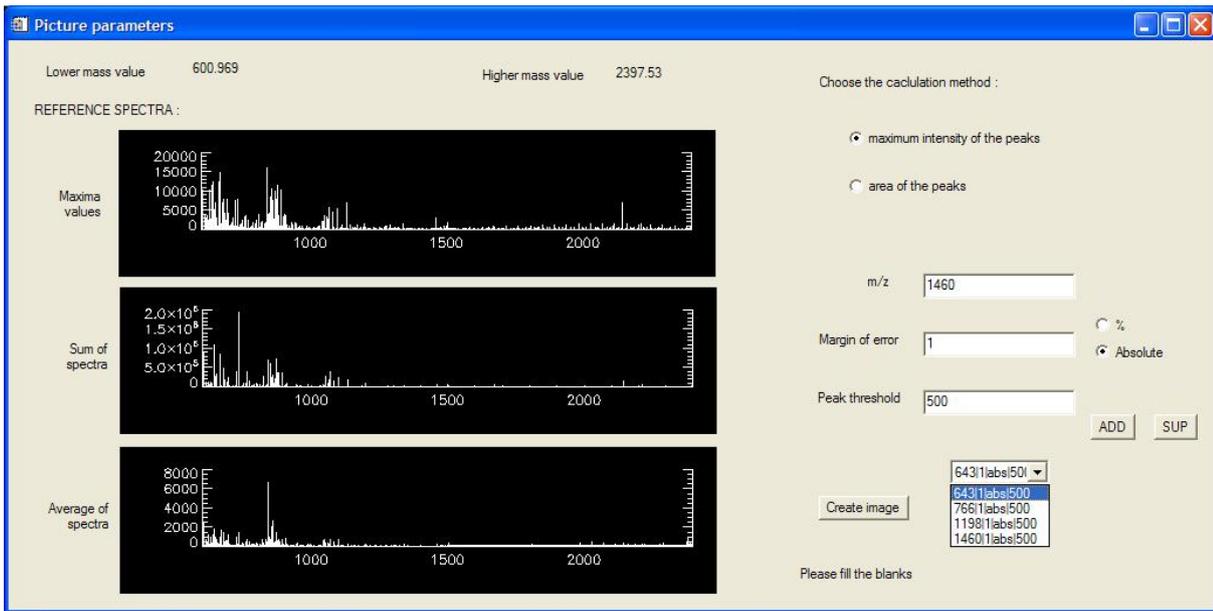


Figure 4

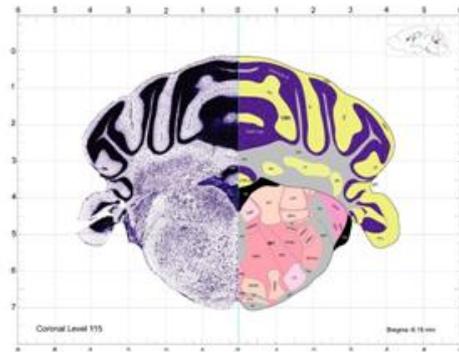
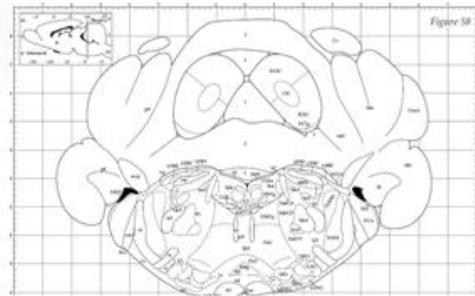
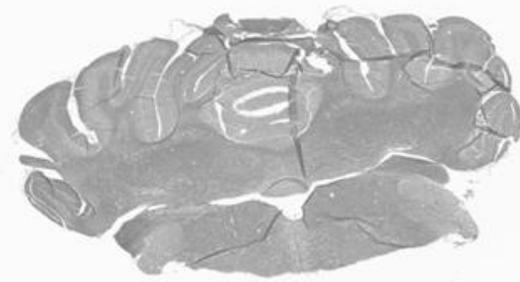
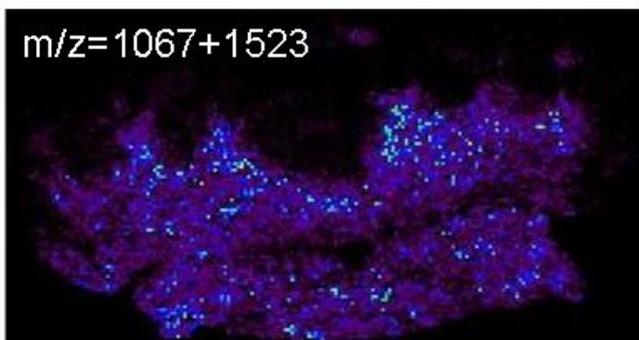
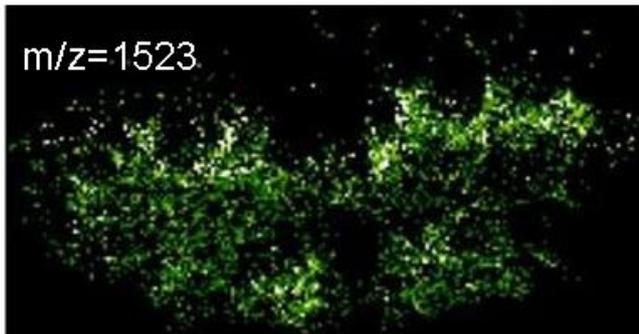
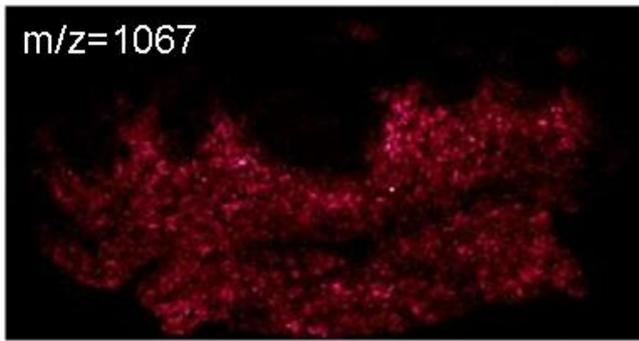
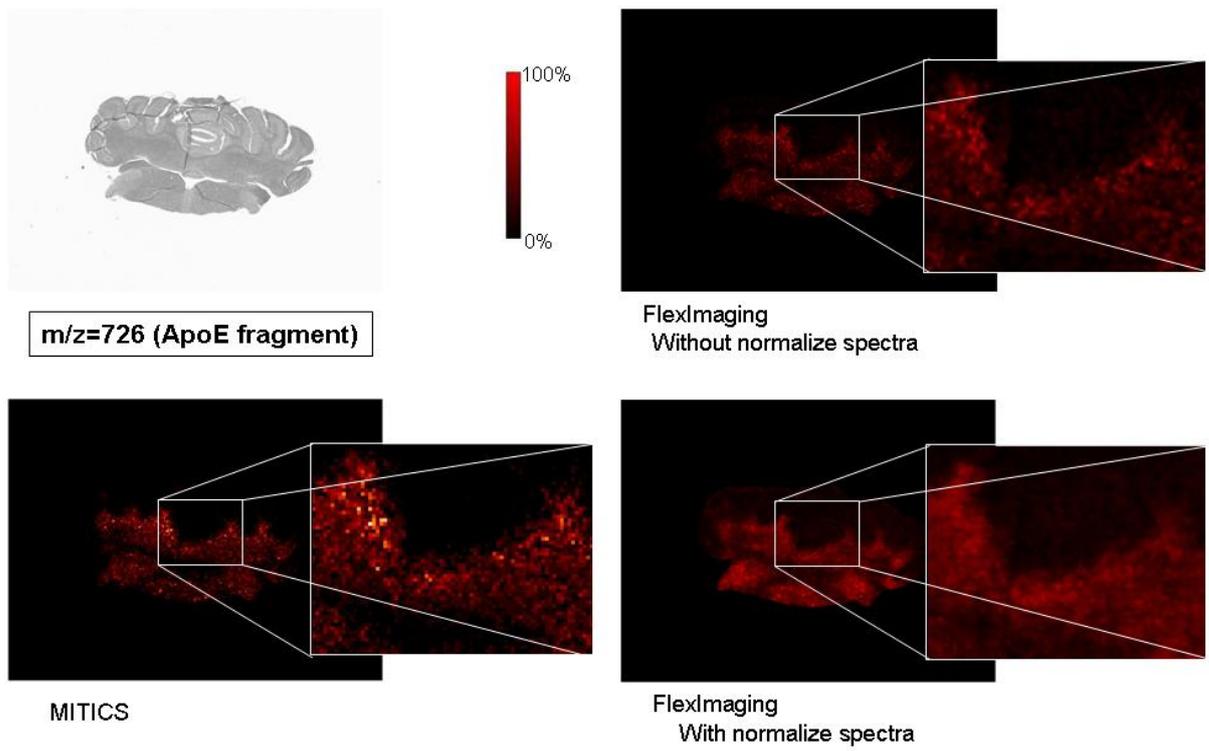
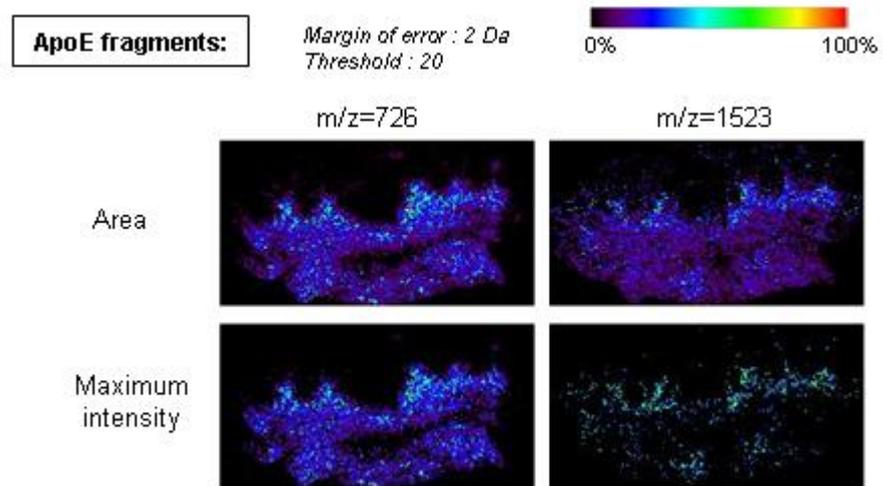


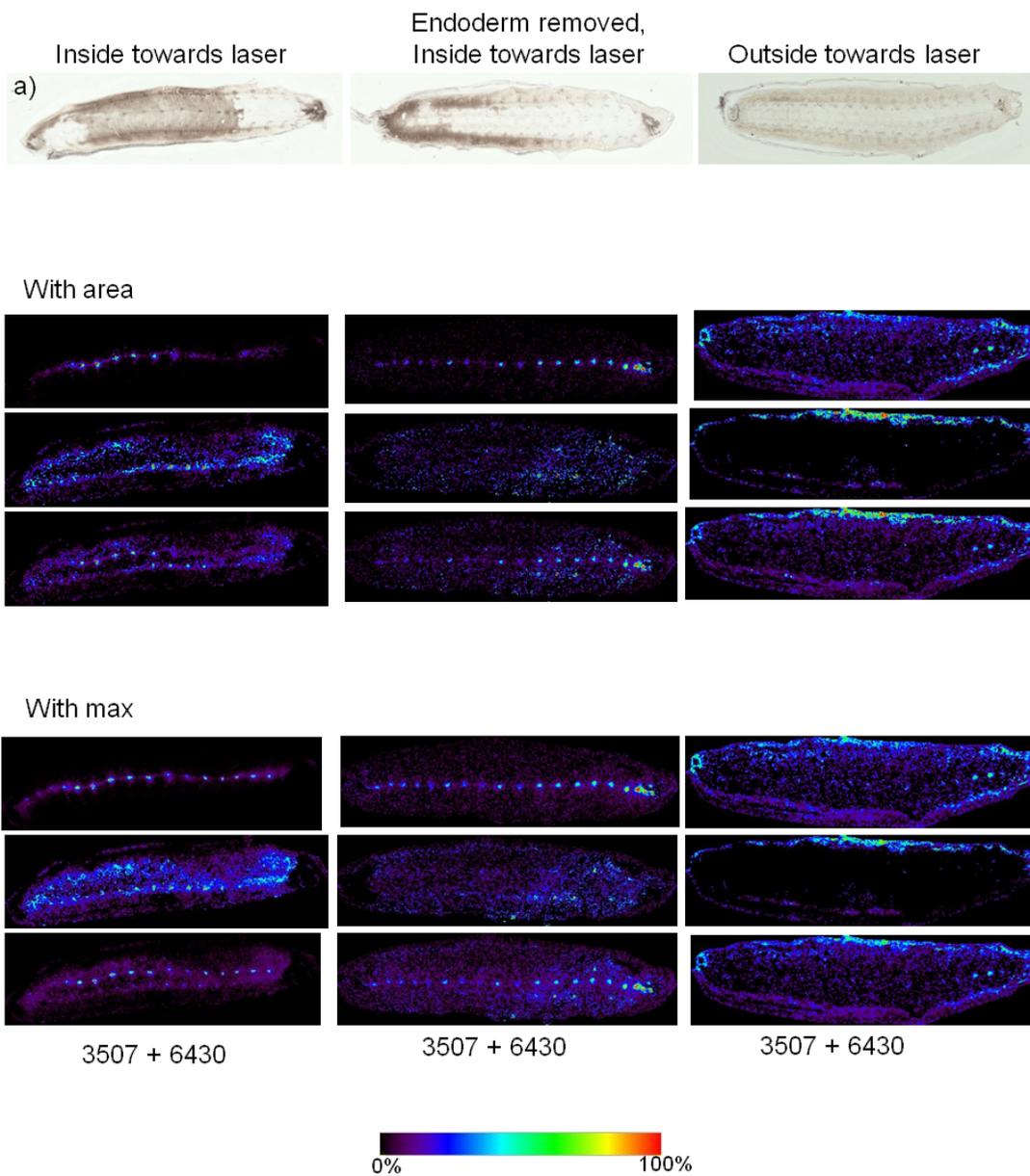
Figure 5



**Figure 6**



**Figure 7**



**Figure 8**

## New Developments in MALDI Imaging for Pathology Proteomic Studies

Maxence Wisztorski<sup>1</sup>, Remi Lemaire<sup>1</sup>, Jonathan Stauber<sup>1</sup>, Sonia Ait Menguelet<sup>1</sup>, Dominique Croix<sup>1</sup>, Olivia Jardin Mathé<sup>1</sup>, Robert Day<sup>2</sup>, Michel Salzet<sup>1</sup>, Isabelle Fournier<sup>1,\*</sup>

<sup>1</sup>Laboratoire de Neuroimmunologie des Annélides, FRE CNRS 2933, équipe Imagerie MALDI, Cité Scientifique, Université des Sciences et Technologies de Lille, 59650 Villeneuve d'Ascq and <sup>2</sup>Département de Pharmacologie, Faculté de médecine, Université de Sherbrooke, Sherbrooke, Québec, J1H 5N4, Canada

**Abstract:** With new emerging mass spectrometry technologies, it can now be demonstrated that *direct tissue analysis* is feasible using matrix-assisted laser desorption/ionization (MALDI) sources. A major advantage of direct MALDI analysis is to avoid time-consuming extraction, purification or separation steps, which have the potential for producing artifacts. Direct MALDI analysis of tissue sections enables the acquisition of cellular expression profiles while maintaining the *cellular and molecular integrity*. With automation and the ability to reconstruct complex spectral data using imaging software, it is now possible to produce *multiplex* imaging maps of selected biomolecules within tissue sections. Thus, direct MALDI spectral data obtained from tissue sections can be converted into imaging maps, a method now known as *MALDI-imaging*. MALDI-imaging combines the power of mass spectrometry, namely exquisite sensitivity and unequivocal structural information, within an intact and unaltered morphological context. Critical improvements to increase image resolution are presented in this manuscript e.g., solvent treatment, new solid ionic matrices, gold sputtering, nickel support or laser focalization. One of the most important developments is the ability to carry out either direct MALDI analysis or MALDI imaging on paraffin tissue sections, thus opening the path to an archival "gold-mine" of existing pathology samples to proteomic analysis. These developments provide new avenues for biomarker hunting and diagnostic follow-up in the clinical setting. Further developments in MALDI-imaging of specific targets provide an added dimension, as validated disease-marker-gene RNA transcripts can be analyzed along with their translation by targeting their specific protein products or metabolites. Disease/health states will thus be closely molecularly monitored at protein and nucleic acids levels, with a single technique. Taken together, MALDI imaging will become a key tool for pathology proteomic studies.

### INTRODUCTION

In the era of proteomics and high throughput studies, need for new technologies presenting higher sensitivity, versatility and specificity are sought. Classic proteomics is now a widely used strategy for obtaining wide ranging protein information from tissues as demonstrated by growing literature in that field. In general, proteomic studies require homogenization of a sample as well as long, tedious and consuming extraction, purification and separation steps before characterization, resulting in the loss of the anatomical data. Over these past ten years, improvements of mass spectrometry instruments together with growing importance of this method for compound identification in the field of biology had led to the development of direct analysis of tissue samples by the means of mass spectrometry. Mass spectrometry has become an analysis tool allowing identification of compounds directly from tissues without any extraction or separation, but adds one dimension to the analysis with some localization on tissue samples. Thus, in a single experiment, molecular information on hundreds of proteins can be retrieved. By incorporating automation of this method and data processing, molecular images can be obtained from tissue sections. A major advantage of this method, rely on the sensitivity of mass spectrometry instruments revealing hundred compound molecular images after one data set acquisition. MALDI ion sources are well suited for this application since it gives access to very different biomolecules families ranging from peptides, proteins to oligonucleotides, sugars or lipids with a spatial resolution at the cellular level. MALDI imaging was first introduced by Caprioli and collaborators [1-3] (Fig. 1). Few studies have involved the still recent developments in MALDI imaging. Of note are the studies of Pr B. Spengler (Giessen, Germany) with the development of a new microprobe MALDI mass spectrometer combining the Laser Microprobe Mass Spectrometer concept with a more refocused laser beam to increase images spatial resolution (SMALDI or Scanning micro

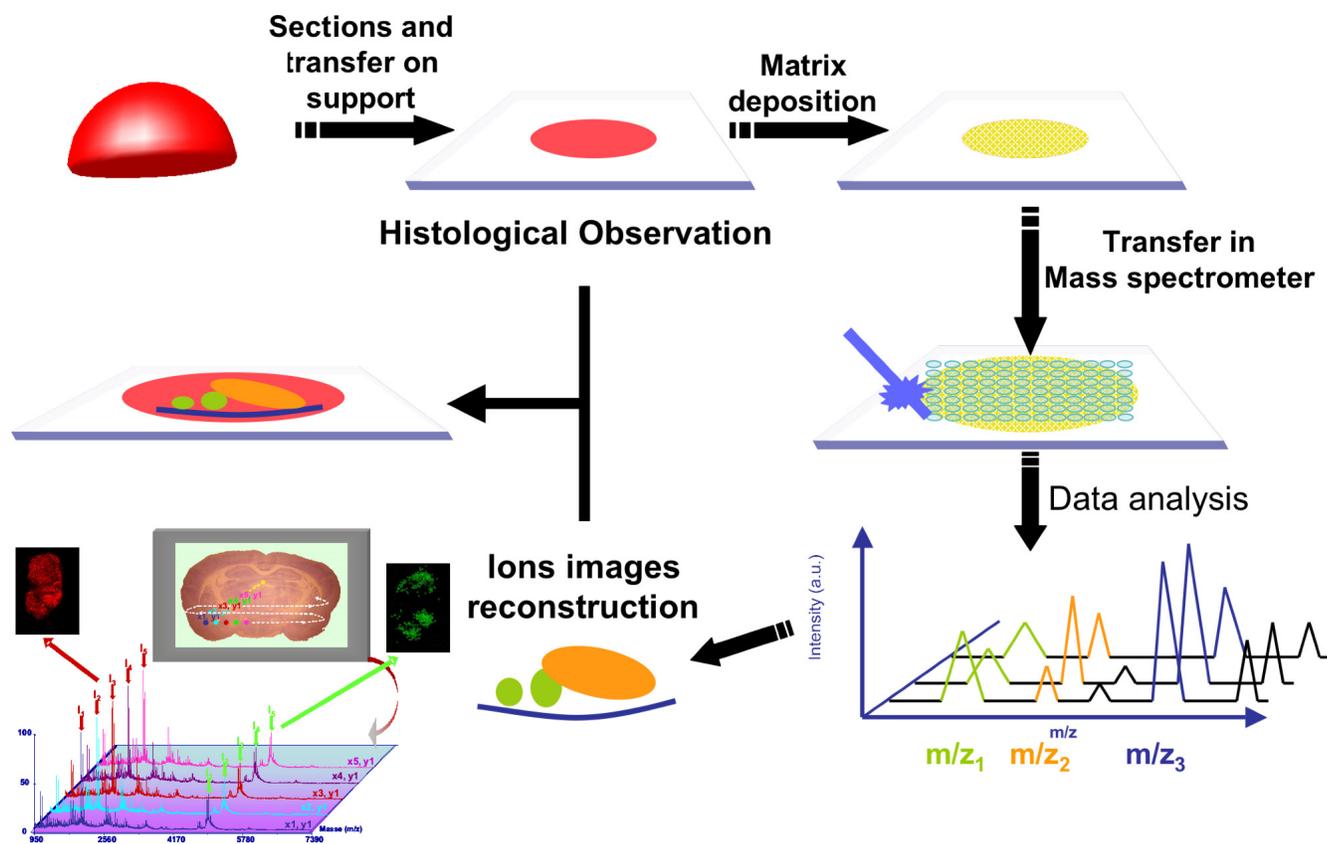
probe matrix-assisted laser desorption ionization) [4]. This group has also developed MALDI imaging instruments with atmospheric pressure MALDI (AP-MALDI). Other important developments have been established by Pr. R.M.A. Heeren (FOHM institute, Netherlands) introducing the concept of imaging using an instrument in the microscope mode with a position detector allowing image reconstruction on the base of ion arrival position [5, 6]. Interesting applications on aging diseases have also been reported by M. Stoeckli and coll. (Novartis Pharma, Switzerland) [7]. MALDI imaging of lipids [8-11] is also under current development in the group of Pr. A. Woods. Neuroproteomics is well developed by the group of J. Sweedler [12-15] and our group has focused on developing tissue preparation methods as well as probe specific MALDI-imaging [16-18].

MALDI-imaging combines the power of mass spectrometry, namely exquisite sensitivity and unequivocal structural information, within an intact and unaltered morphological context. Furthermore, our unprecedented ability to carry out this powerful technique on paraffin tissue sections opens up an archival "gold-mine" of existing pathology samples to proteomic analysis.

### NEW DEVELOPMENTS FOR ROUTINE HISTO-PROTEOMICS IN PATHOLOGY

Histo-proteomics MALDI technology can be developed for drug biodistribution studies, biomarker discovery or for molecular mechanism discovery. At the present time, MALDI imaging can provide new avenues for clinical proteomic studies with the goal of characterizing cellular circuitry and to understand the impact of disease and therapy on cellular networks though analysis of tissue pathology samples. In fact more than just new clinical proteomic tool, MALDI-imaging has the potential to become a major method in the way tissue samples are investigated in pathology. MALDI-imaging histo-proteomic is the application of proteomic technologies and bioinformatic tools to clinical material though tumor analysis. The translational nature of this technology provides unique challenges and yet unimagined opportunities that promise to transform the way disease is detected, treated, and managed. Rather

\*Address correspondence to this author at the Laboratoire de Neuroimmunologie des Annélides, FRE CNRS 2933, équipe Imagerie MALDI, Cité Scientifique, Université des Sciences et Technologies de Lille, 59650 Villeneuve d'Ascq; E-mail: isabelle.fournier@univ-lille1.fr



**Fig. (1).** Representative scheme of MALDI imaging technology.

Tissue sections from fresh organ or biopsy are laid out on the MALDI target sections first covered with a specific matrix depend to the nature of the bio-molecule under study. For peptides/proteins, very intense signals are obtained with  $\alpha$ -cyano-4-hydroxycinnamic acid ( $\alpha$ -CHCA) as a matrix. Sections are covered with  $\alpha$ -CHCA (or another matrix) and are then introduced in the MALDI-TOF for analysis. Next, MALDI laser is used to scan each point of the surface area and the mass spectra representative of the peptides/proteins or lipids present in each point are analyzed. Automated analysis of the complete tissue is obtained by performing mass spectra every 10-50  $\mu\text{m}$ , providing representative information of selected ions (each ion is a specific bio-molecule). Analysis is obtained within 2-6 hours and images are reconstructed using Flex-imaging software.

than focusing on genetic alterations that may lead to a particular disease, it is emerging that changes in multiple protein expression patterns are the most accurate way to identify diseases in their early stages and to determine the most effective course of treatment. Indeed, genome sequences do not provide information of post-translational modification events such as glycosylation, phosphorylation, acylation or partial proteolysis. One of the most common objectives in proteomics is the study of protein expression patterns (e.g., protein profiling) associated with diseases. In contrast to existing diagnostic assays, which examine protein biomarkers one at a time, MALDI-based histo-proteomics provides powerful simultaneously detection of hundreds or even thousands of proteins in a single assay directly from the tissue. The effectiveness of such a proteomics approach in pathology hinges on two technological components: rapid, multiplex protein detection assays and data analysis systems to assimilate vast amounts of protein expression data from healthy and diseased individuals into clinically relevant data sets. In the present review, we envisage the application of state-of-the-art peptide/protein profiling directly on tissues to study of multi-factorial diseases and to develop new methods based on nanotechnologies for high-throughput proteome characterization. Based on accumulated knowledge, we conclude that whatever the disease's molecular origin (e.g., inherited/acquired abnormalities of intracellular control or of bi-directional communication networks exerting homeostatic control), the corresponding affected tissues are characterized by abnormal patterns of protein/peptide expression and/or secretion.

## TISSUE PREPARATION

Our recent technological advancements in MALDI-imaging allow the identification of novel markers and *in situ* characterization from fresh sections/biopsy embedded in paraffin, including archived material.

As shown in the (Fig. 1), tissue sections from fresh organ or biopsy are laid out on the MALDI target. Sections are first covered with a specific matrix depending on the nature of the bio-molecule under study [19]. For peptides, very intense signals are obtained with ionic matrices [20]. Sections are laid out on conductive glass (i.e., Nickel or ITO, (Fig. 2a), treated with organic solvent to remove lipids [21] (Fig. 2b), covered with a solid ionic matrix [20], overlaid with gold in order to remove charges effects [22] and are then introduced in the MALDI-TOF for analysis (Fig. 2c). Next, the MALDI laser is used to scan each point of the designated surface area and the mass spectra representative of the biomolecules present in each point are analyzed. Automated analysis of the complete tissue is obtained by performing mass spectra every 50-200  $\mu\text{m}$ , providing representative information of selected ions (each ion is a specific bio-molecule) [20, 23, 24]. Analysis is obtained within 2-6 hours and images are reconstructed using imaging software (Fig. 1). We recently made a important advances with the development of specific new matrices, called ionic matrices, for high repetition laser (200 Hz). These ionic matrices advantageously prevent peptide delocalization after deposition, permit better coverage as well as low level ablation and the possibility to work in positive or nega-

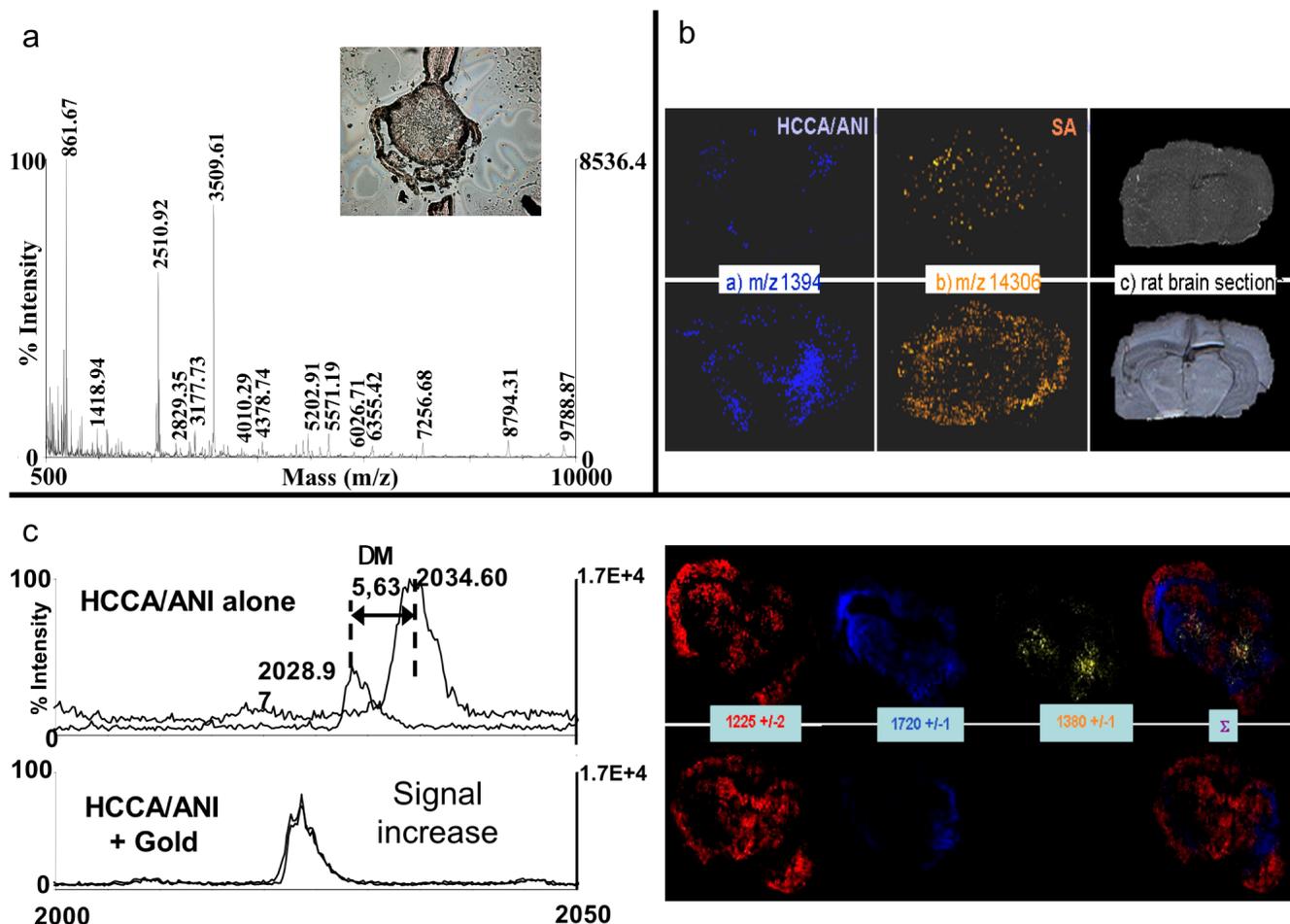


Fig. (2). Scheme of the MALDI imaging new developments.

a- Mass spectra of leech ganglia on Nickel Glasses for tissue application on conductive and transparent supports (inset represent a leech ganglia section on nickel Glass).

b- Solvents washing for removing lipids according to [21].

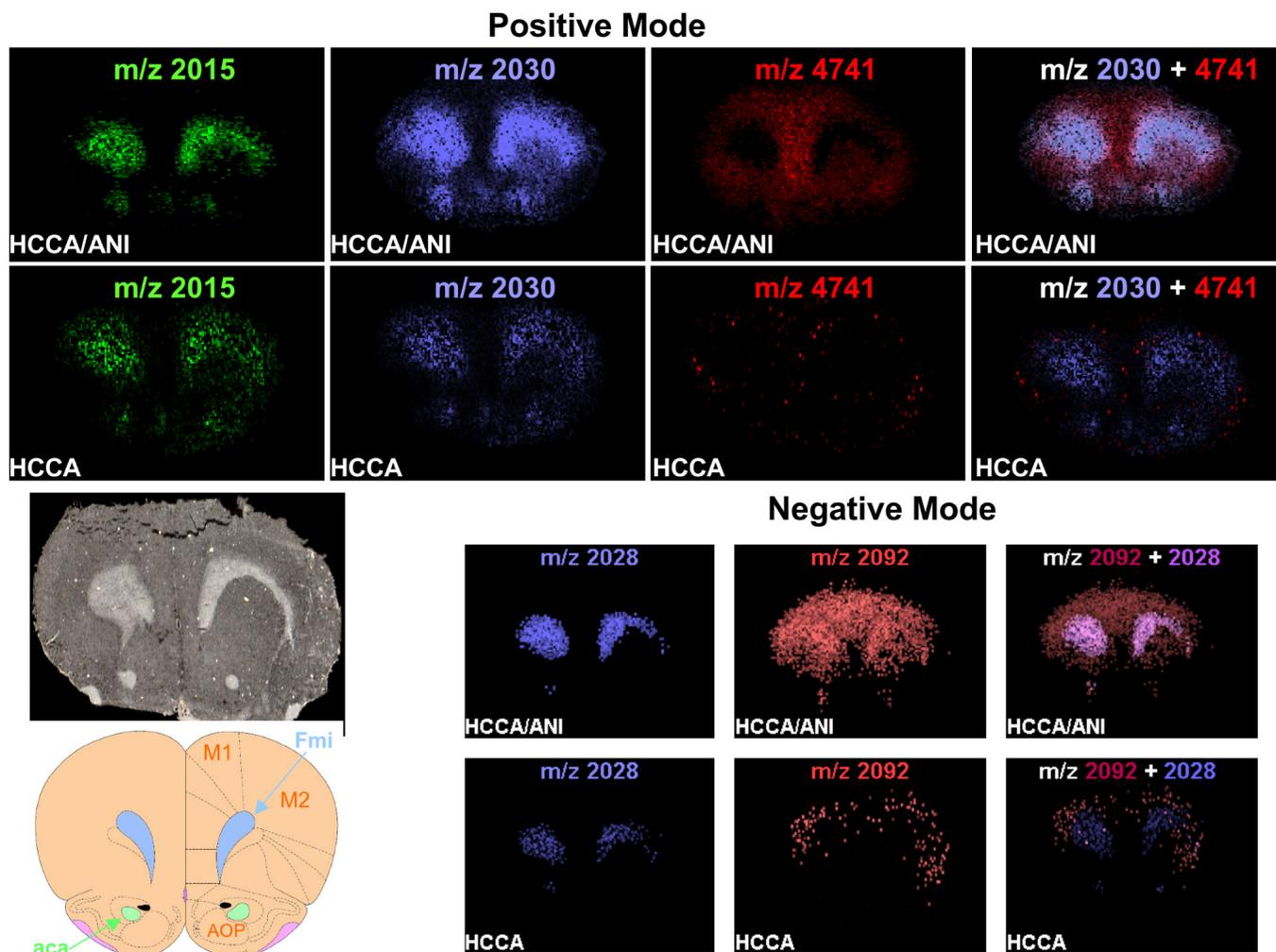
c- Gold sputtering of the slices for removing charge effects. Comparison was performed with HCCA/ANI matrix in present or absence of gold (Left panel). All spectra are obtained in a laser energy close to the minimal value for ions production and are obtained in a laser energy close to the minimal value for ions production. 15 shots are realized with an increasing of 15% of the laser fluence (no acquisition) and immediately after a spectrum is recorded in the same conditions of laser energy. Right panel represents MALDI-IMS using MALDI TOF/TOF in reflector mode at 50 Hz repetition rate with ionic matrix HCCA/ANI. The image have been reconstructed with FlexImaging software and represent the superposition of the repartition of  $m/z = 1225$  (blue),  $m/z = 1720$  (red) and  $m/z = 1380$  (yellow).

tive modes [20]. One such matrix, namely ANI (aniline/HCCA) is highly stable under vacuum and has high resistance to repetitive laser shots (Fig. 3).

The use of archived material in paraffin blocks from hospital pathology departments thus represents a major source of catalogued existing information. However, the major hurdle to analyse such samples is the cross-linking due to formalin fixation and embedded in paraffin (FFPE tissue). Two approaches have been established depending on the age of the tissue blocks. In the case of tissues stored less than 6 months, an active matrix, namely 2,4-dinitrophenylhydrazine (DNPH) is used. DNPH neutralizes formalin excess and allows analysis of embedded tissues such as those from paraffin sections (Fig. 4a) [25]. In the case of tissue blocks stored more than 6 months, micro-digestion with trypsin must be performed. On adjacent sections, one is used for protein characterization and the other for a raster image using a microspotter for spotting the trypsin (200  $\mu\text{m}$ ) then the matrices at the same level. The section used for the protein characterization is treated with trypsin after paraffin removal and rehydration. The digest is then injected

in nanoLC-nanoESI/IT MS trap for peptide characterization. The ions of the protein shown to change between normal or pathology are then examined on the adjacent slices in order to obtain an image of its localization. All ions deriving from the same protein after digestion can then be localized and the same cellular localization is obtained validating the methodology (Fig. 4b) [16, 25].

These developments in the histo-proteomics of MALDI-imaging for archived materials could lead to the creation of a national disease marker database, and allow the elaboration of early diagnostics for various pathologies as well as a follow up in disease progression. MALDI-imaging has the potential applications for drug development. In this context MALDI-imaging can be used to analyze drug metabolic pathways directly in tissues (e.g., through in situ multiplex analysis of metabolites), providing important information in order to understand secondary effects and unexpected feedback loops. In order to provide specific examples of the versatility and power of MALDI-imaging, we briefly present preliminary data on a well established animal model of Parkinson's disease, where mice are treated with 6 hydroxydopamine (6-OHDA). In



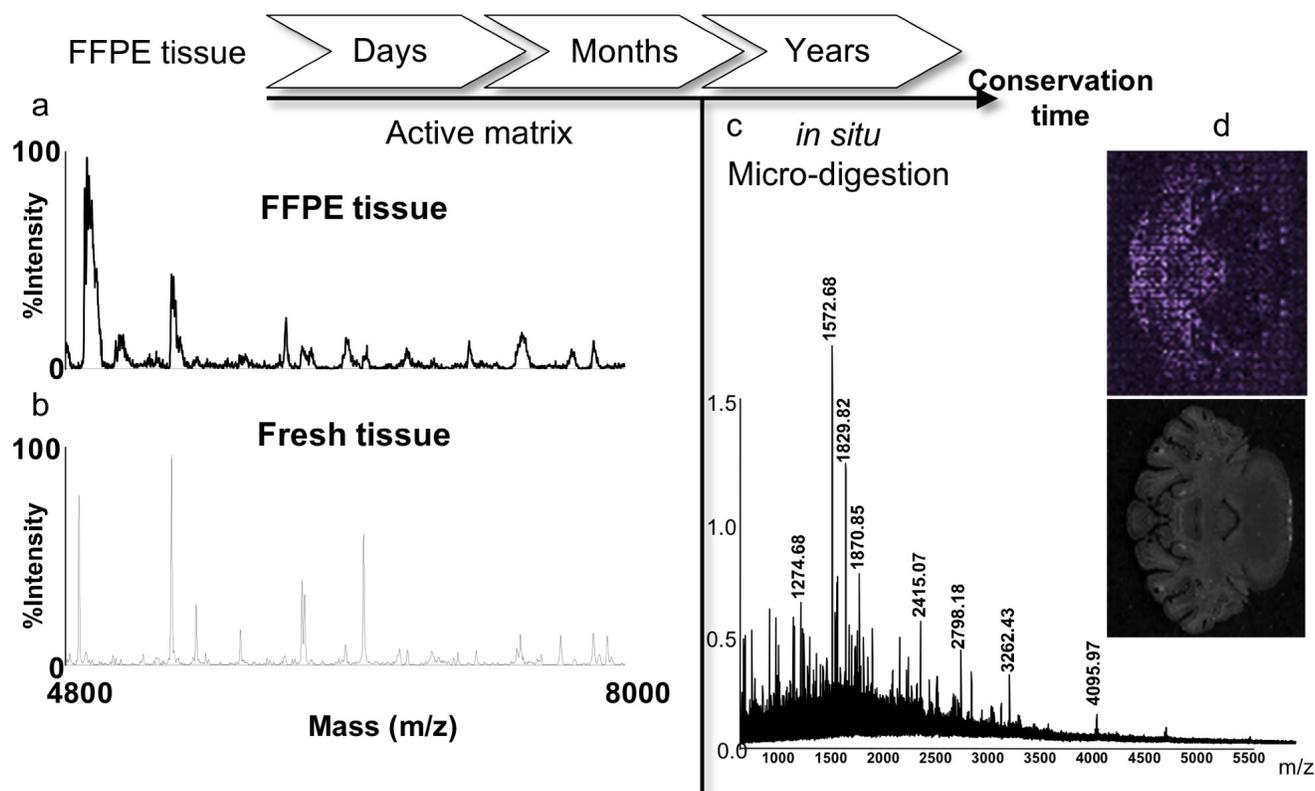
**Fig. (3).** MALDI-IMS using MALDI TOF-TOF in reflector mode at 50-Hz repetition rate with ionic matrices CHCA/ANI and CHCA in positive (a) and negative modes (b). MALDI imaging can be compared with rat brain anatomy (c). For CHCA/ANI and CHCA, acquisitions in both polarities were performed on the same rat brain cut. Images have been reconstructed with the same parameters for ionic matrix and CHCA using FlexImaging software and represent the repartition of a  $m/z$  in the tissue slice. Images with two colors correspond to the superposition of two  $m/z$  images (aforecortex minor of corpus callosum, banterior commissure, and c,dmortal cortex) With permission from analytical chemistry [20].

neurodegenerative diseases such as Parkinson's and Alzheimer's diseases, post-mortem tissues are used to identify disease markers. A well-studied animal model of Parkinson's disease is the rat model subjected to 6-OHDA treatment. Direct MALDI analysis on FFPE brain tissue sections (stored for 9 years) from treated and control rats were carried out with an enzymatic digestion followed by a tandem mass spectrometry (MS-MS) analyses and then subjecting the entire fragmentation spectra to the Mascot™ interface software (Matrix Science Ltd). A total of eight markers were considered noteworthy, either down- or up-regulated, five of which have previously been described in Parkinson's disease literature (see Table 1). This demonstrates the feasibility of this technique and its great potential in future histo-proteomics to discover new biomarkers from archived tissues from hospital library.

### SPECIFIC MALDI IMAGING

In order to add a dimension of specificity to MALDI-imaging, we have developed designed probes directed against specific targets. This strategy has been developed for various biomolecules, from mRNA and peptides/proteins [26]. In order to specifically amplify signals, we have developed the concept of "Tag-Mass". Tag-Mass is a novel method that uses labeled probes for specific

identification in MALDI-imaging. As shown in (Fig. 5a) the concept is schematized with an oligonucleotide sequence (or probe) that hybridizes mRNA in a tissue section as is done in standard *in situ* hybridization techniques. An oligonucleotide probe sequence is attached to a photo-cleavable group linked to a "Tag" marker which is an amino acid sequence of defined mass. This "Tag" marker can be modified to generate "Tags" of different known masses. Following hybridization of the Tag-Mass probe to the tissue section, MALDI analysis is performed as previously described, however, the pulse laser will cleave the photo-cleavable group to yield a Tag of a defined mass. The signals obtained for specific bio-molecules in Tag-Mass will be much higher and will yield unique signatures. Our results with a first generation of Tag-Mass probes have established a proof-of-principal, as we can detect hybridized mRNAs on tissue sections. We then developed a second generation Tag-Mass, with a modified uracil base that bears the photo-cleavable linker, allowing us to perform multiplex *in situ* hybridization using MALDI technology (Fig. 5b). Our results also establish the concept that a photo-cleavable linker can be used in other application with antibodies, lectin or aptamers for use in Tagged-specific MALDI-imaging. The development of "Universal Tags" for direct *in situ* tissue analysis by MALDI-TOF mass spectrometry is a significant



**Fig. (4).** Strategies used for FFPE tissue (a) stored less than 6 months, (b) stored more than 6 months.

**a:** Compared MALDI mass spectra in the linear positive mode of the direct analysis of a <1year old FFPE and fresh frozen rat brain tissues recorded in the same region With sinapinic acid as matrix.

**b:** MALDI mass spectrum in the linear positive mode of the direct analysis of a <1year old FFPE tissue using 2,4-DNPH as matrix. Zooming compared this spectrum to the one recorded in the same conditions and in the same region of the rat brain of a fresh frozen tissue.

**c:** MALDI mass spectrum in the linear positive mode of the direct analysis of a 2 years old FFPE tissue section after *in situ* trypsin digestion of the whole tissue section (15 min).

**d:** MALDI molecular images reconstructed from the data recorded on the 2 years old FFPE rat brain tissue section after micro-spotted *in situ* trypsin digestion followed by extraction and performed on MALDI-TOF-TOF/TOF using HCCA as matrix and compared to rat brain picture and morphology.

**Table 1.**

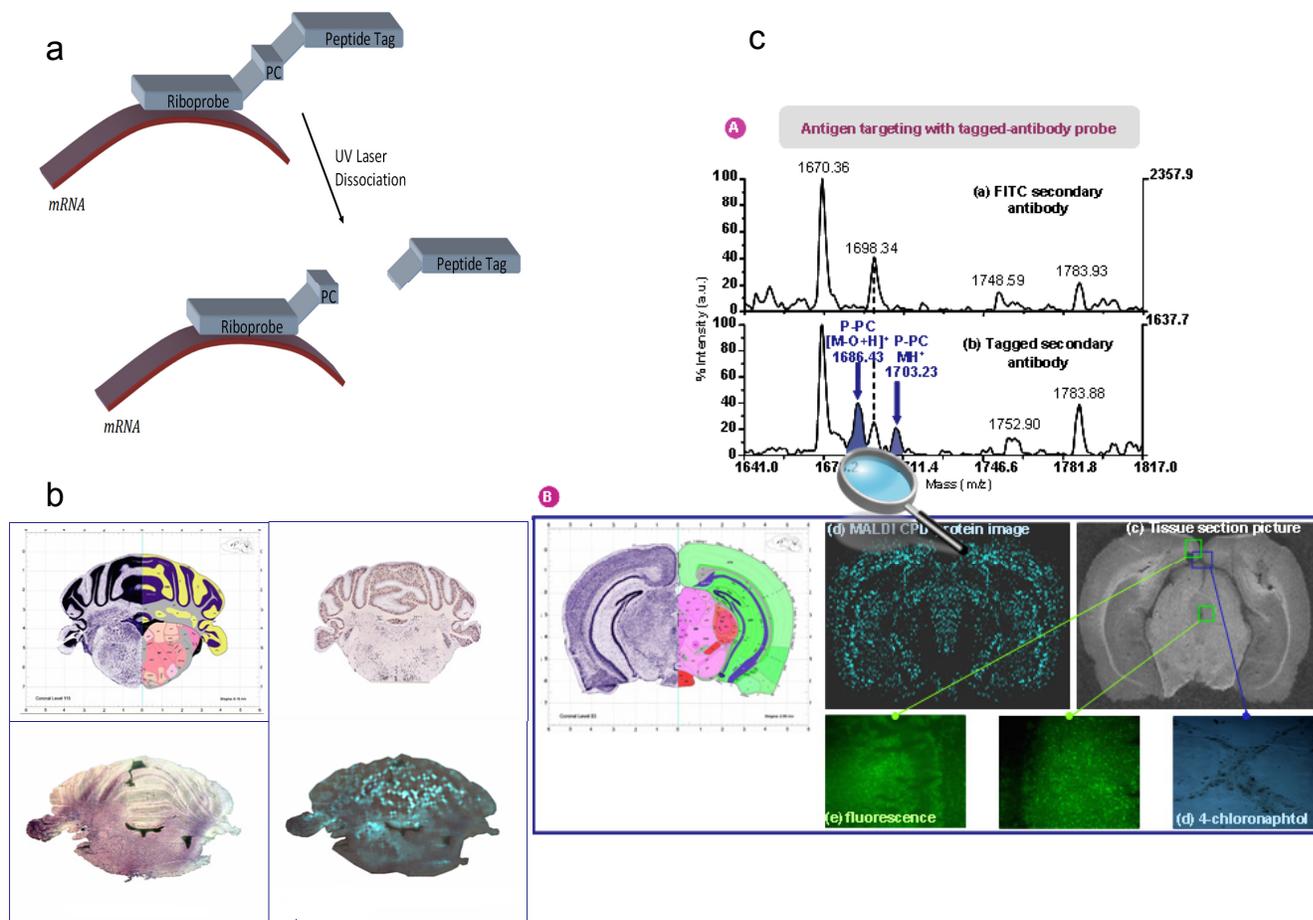
Markers	m/z	Mascot Score (%)	Known in Parkinson's Disease Literature
Downregulated	Ubc-Ubiquitin	78	?
	Trans elongation Factor 1 (eEF1)	66	?
	Hexokinase (2-7-1-1)	76	Pastosis <i>et al.</i> Pharm. Res. 1995 31, 361-369
Upregulated	Neurofilament M protein	57	Basso <i>et al.</i> Proteomics 2004 12, 3943-3952
	Peroxidoxin 6	65	Basso <i>et al.</i> Proteomics 2004 12, 3943-3952
	Collapsin response Mediateur Protein	70	?
	F1 ATPase	81	Seo <i>et al.</i> Human Gen Ther. 2004 15, 887-895
	Enolase alpha	62	Poon <i>et al.</i> Neurobiol Des. 2005 18, 492-498

achievement toward specific molecular diagnosis. Thus we anticipate targeting specific disease-marker-gene RNA transcripts, following their expression within tissues and then confirming their translation by targeting their specific protein products or metabolites. Disease/health states will thus be closely molecularly monitored.

#### IMPROVING IMAGE QUALITY

The desorption/ionization process is mediated in MALDI analysis by the irradiation of the sample by the laser beam, for a

fixed sample, the analyzed region is limited to the area irradiated by the laser beam. It is then possible to perform analyses on various positions of the sample and to obtain from each position a spectrum representing ionic species present in that position. Thus, by shifting the laser beam of a regular pitch defined by the user, the whole sample may be scanned, and a database comprising all spectra and their coordinates may be generated, which allows us to construct an expression map of any compound of known m/z ratio in the analyzed sample. UV lasers used in MALDI imaging, and especially commercialized N<sub>2</sub> lasers emitting at 337 nm, have a laser beam



**Fig. (5).** Scheme of the tag-mass probe concept (a). MALDI molecular image reconstructed from data recorded with a proenkephalin tagged probe compared to mouse brain morphology and digoxigenin in situ hybridization data of mouse proenkephalin gene repartition (b) Molecular image of the carboxypeptidase D, a membrane protein of 180 kDa detected with a tagged antibody and compared to the same experiments with fluorescein and chloronaphthal revelation (c) [26].

section area generally ranging between  $75 \times 75 \mu\text{m}^2$  and  $200 \times 200 \mu\text{m}^2$  with a classical focalization system. For tissue imaging, the minimum distance between two points will have to be superior to the laser beam diameter, resulting in an image definition of at most the laser beam diameter (thus at best  $75 \times 75 \mu\text{m}$ ), which corresponds to the irradiation of several cells in the tissue sample. Ideally, the image definition of a tissue sample should be or the order of a cell diameter ( $10\text{--}20 \mu\text{m}$  for small cells). In fact, the decrease by a factor 10 ( $10 \mu\text{m}$  steps) will increase imaging resolution.

Two types of solutions have been proposed to increase image resolution. The first, done by Caprioli team, is based on the different methods for matrix deposition to obtain more homogeneous microcrystalline layers and reduce peptide dispersion in the tissues (peptide dilution and delocalisation). An acoustic reagent multispotter has been developed to provide improved reproducibility for depositing matrix onto a sample surface [27]. For tissue sections, matrix spots of  $180\text{--}200 \mu\text{m}$  in diameter were obtained and a procedure is described for generating coordinate files readable by a mass spectrometer to permit automated profile acquisition. Mass spectral quality and reproducibility was found to be better than that obtained with manual pipet spotting [27].

The second solution, developed by our group, implies a smaller diameter of the laser beam in order to get a diameter of the beam close to single cell dimensions. For this position, conductive masks have been developed to decrease the area irradiated by the laser beam for MALDI analysis in order to get a more precise localiza-

tion of the biomarkers in the tissue. Moreover, improving the beam focusing will get a homogeneous [28] energy profile laser beam (flat top laser) (Fig. 6).

## CONCLUSION

The future impact of MALDI-imaging can be linked to the formidable advancements made several years ago in the field of nuclear magnetic resonance (NMR) or in the development of the polymerase chain reaction (PCR). NMR has led to the development of magnetic resonance imaging (MRI) in the clinical setting for the diagnosis of various pathologies, while PCR revolutionized molecular biology and clinical diagnosis. Likewise, we foresee that MALDI-imaging will have an enormous impact in the clinical setting for the diagnosis and research leading to treatments in pathology proteomics.

## ACKNOWLEDGEMENTS

Supported by grants from Centre National de la Recherche Scientifique (CNRS), the Canadian Institutes of Health Research (CIHR) to R. Day, Ministère de la Recherche et des Technologies (MRT, ACI, INCA, ANR PCV to I. Fournier), Région Nord-Pas de Calais (to M. Wisztorski), by the Société Française de Spectrométrie de Masse (to R. Lemaire and M. Wisztorski) and by the Société Française d'Electrophorèse et d'Analyse Protéomique (to J. Stauber and M. Wisztorski). Also, supported by a collaboration agreement between Bruker Daltonics and the Laboratoire de Neuroimmunologie des Annélides.

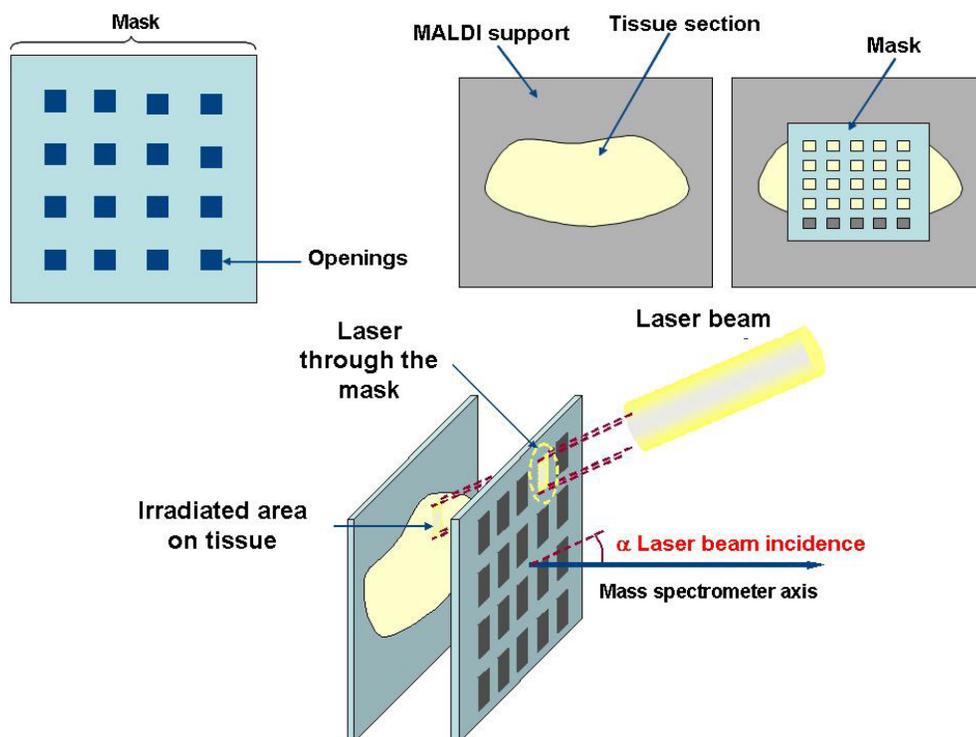


Fig. (6). Scheme of laser beam reduction through masks. Reduction is now 30 x 30  $\mu\text{m}$  [28]

## REFERENCES

- [1] Chaurand P, Cornett DS, Caprioli RM. Molecular imaging of thin mammalian tissue sections by mass spectrometry. *Curr Opin Biotechnol* 2006; 17(4): 431-6.
- [2] Chaurand P, Schwartz SA, Reyzer ML, Caprioli RM. Imaging mass spectrometry: principles and potentials. *Toxicol Pathol* 2005; 33(1): 92-101.
- [3] Chaurand P, Schwartz SA, Caprioli RM. Imaging mass spectrometry: a new tool to investigate the spatial organization of peptides and proteins in mammalian tissue sections. *Curr Opin Chem Biol* 2002; 6(5): 676-81.
- [4] Spengler B, Hubert M. Scanning microprobe matrix-assisted laser desorption ionization (SMALDI) mass spectrometry: instrumentation for sub-micrometer resolved LDI and MALDI surface analysis. *J Am Soc Mass Spectrom* 2002; 13(6): 735-48.
- [5] Heeren RM. Proteome imaging: a closer look at life's organization. *Proteomics* 2005; 5(17): 4316-26.
- [6] Luxembourg SL, Mize TH, McDonnell LA, Heeren RM. High-spatial resolution mass spectrometric imaging of peptide and protein distributions on a surface. *Anal Chem* 2004; 76(18): 5339-44.
- [7] Stoeckli M, Farmer TB, Caprioli RM. Automated mass spectrometry imaging with a matrix-assisted laser desorption ionization time-of-flight instrument. *J Am Soc Mass Spectrom* 1999; 10(1): 67-71.
- [8] Jackson SN, Wang HY, Woods AS. In situ structural characterization of phosphatidylcholines in brain tissue using MALDI-MS/MS. *J Am Soc Mass Spectrom* 2005; 16(12): 2052-6.
- [9] Jackson SN, Wang HY, Woods AS. Direct profiling of lipid distribution in brain tissue using MALDI-TOFMS. *Anal Chem* 2005; 77(14): 4523-7.
- [10] Woods AS, Jackson SN. Brain tissue lipidomics: direct probing using matrix-assisted laser desorption/ionization mass spectrometry. *AAPS J* 2006; 8(2): E391-5.
- [11] Woods AS, Ugarov M, Jackson SN, Egan T, Wang HY, Murray KK, *et al.* IR-MALDI-LDI combined with ion mobility orthogonal time-of-flight mass spectrometry. *J Proteome Res* 2006; 5(6): 1484-7.
- [12] Amare A, Hummon AB, Southey BR, Zimmerman TA, Rodriguez-Zas SL, Sweedler JV. Bridging neuropeptidomics and genomics with bioinformatics: Prediction of mammalian neuropeptide prohormone processing. *J Proteome Res* 2006; 5(5): 1162-7.
- [13] Hummon AB, Amare A, Sweedler JV. Discovering new invertebrate neuropeptides using mass spectrometry. *Mass Spectrom Rev* 2006; 25(1): 77-98.
- [14] Jurchen JC, Rubakhin SS, Sweedler JV. MALDI-MS imaging of features smaller than the size of the laser beam. *J Am Soc Mass Spectrom* 2005; 16(10): 1654-9.
- [15] Rubakhin SS, Jurchen JC, Monroe EB, Sweedler JV. Imaging mass spectrometry: fundamentals and applications to drug discovery. *Drug Discov Today* 2005; 10(12): 823-37.
- [16] Lemaire R, Desmons A, Ducroy P, Tabet J.C, Salzet M, Fournier I. Direct analysis and MALDI imaging on formalin fixed paraffin embedded tissue (FFPE): application to Parkinson disease. Proceedings of 54th ASMS conference on Mass Spectrometry, Seattle, Washington, May 30-June 2, 2006.
- [17] Fournier I, Lemaire R, Wisztorski M, Stauber J, Jardin-Mathé O, Van Camp C, *et al.* New insight in sample preparation for MALDI imaging and new developments to approach specific MALDI imaging of the transcriptome. Proceedings of 54th ASMS conference on Mass Spectrometry, Seattle, Washington, May 30-June 2, 2006.
- [18] Lemaire R, Ducroy P, Wisztorski M, Hendra JB, Tabet JC, Salzet M, *et al.* Exploring direct analysis using ionic matrices. proceedings of 53rd ASMS conference on mass spectrometry, San Antonio, Texas, June 5-9, 2005.
- [19] Fournier I, Day R, Salzet M. Direct analysis of neuropeptides by in situ MALDI-TOF mass spectrometry in the rat brain. *Neuro Endocrinol Lett.* 2003; 24(1-2): 9-14.
- [20] Lemaire R, Tabet JC, Ducroy P, Hendra JB, Salzet M, Fournier I. Solid ionic matrices for direct tissue analysis and MALDI imaging. *Anal Chem* 2006; 78(3): 809-19.
- [21] Lemaire R, Wisztorski M, Desmons A, Tabet JC, Day R, Salzet M, *et al.* MALDI-MS direct tissue analysis of proteins: Improving signal sensitivity using organic treatments. *Anal Chem* 2006; 78(20): 7145-53.
- [22] Wisztorski M, Brunet L, Dreiserwer K, Hillenkamp F, Berkenkamp S, Salzet M, *et al.* Effect of metals coating for YV MALDI-a-TOF mass spectrometry imaging (MALDI MSI) and direct tissue analysis in UV/IR MALDI-o-TOF mass spectrometry. Proceedings of

- 54th ASMS conference on Mass Spectrometry, Seattle, Washington, May 30-June 2, 2006.
- [23] Caprioli RM, Farmer TB, Gile J. Molecular imaging of biological samples: localization of peptides and proteins using MALDI-TOF MS. *Anal Chem* 1997; 69(23): 4751-60.
- [24] Stoeckli M, Chaurand P, Hallahan DE, Caprioli RM. Imaging mass spectrometry: a new technology for the analysis of protein expression in mammalian tissues. *Nat Med* 2001; 7(4): 493-6.
- [25] Lemaire R, Desmons A, Day R, Fournier I, Salzet M. Analysis of peptides and proteins in formalin fixed and paraffin embedded rat brain tissue sections by MALDI mass spectrometry. *J Proteome Res*, in press.
- [26] Lemaire R, Stauber J, Wisztorski M, Van Camp C, Desmons A, Deschamps M, *et al.* Tag-mass: Specific molecular imaging of transcriptome and proteome by mass spectrometry based on photocleavable tag. *J Proteome Res* 2007; 6(6): 2057-67.
- [27] Aerni HR, Cornett DS, Caprioli RM. Automated acoustic matrix deposition for MALDI sample preparation. *Anal Chem* 2006; 78(3): 827-34.
- [28] Wisztorski M, Verplanck N, Thomy V, Stauber J, Camart JC, Salzet M, *et al.* Use of masks in MALDI MS: an easy tool for increasing spatial resolution of images by decreasing irradiated area. Proceedings of the 55th ASMS conference on Mass Spectrometry, Indianapolis, Indiana, June 3-7, 2007.

## **Imagerie MALDI : Une nouvelle technologie pour découvrir et valider de nouveaux biomarqueurs**

Maxence Wisztorski, Rémi Lemaire , Jonathan Stauber, , Sonia Ait Menguelet, Olivia Jardin-Mathé, Michel Salzet et Isabelle Fournier\*

Laboratoire de Neuroimmunologie des Annélides, FRE CNRS 2933, équipe Imagerie MALDI, Bâtiment SN3, Cité Scientifique, Université des Sciences et Technologies de Lille, 59650 Villeneuve d'Ascq

### **Résumé**

A l'heure de la protéomique, la spectrométrie de masse s'est révélée un outil puissant pour la recherche et l'identification des protéines et des peptides à partir d'échantillons purifiés. Une nouvelle ère s'ouvre, avec l'imagerie MALDI, permettant à la fois la recherche, l'identification et maintenant la localisation de biomolécules telles que les peptides, les protéines ou les lipides au sein des tissus. Des développements cruciaux au niveau de la préparation des échantillons permettent l'analyse de tissus ou biopsies conservées congelées (-80°C) depuis plus de 6 mois ou conservés après fixation au paraformaldéhyde et inclus en paraffine. Le verrou technologique permettant l'accès aux tumorothèques est enfin circonscrit et la chasse aux biomarqueurs au sein de ces banques, objectif déterminant, est devenue possible. Le rapprochement des techniques d'imagerie MALDI de celles utilisées en routine par les anatomopathologistes (colorations ou utilisation d'anticorps) est également un objectif important. Le développement de l'imagerie spécifique grâce à l'utilisation de sondes appelées « Tag-Mass » ouvre des perspectives potentielles pour cette technologie. En effet, il est dorénavant possible de localiser une protéine et son ARNm de façon spécifique ainsi que les voies de signalisation sur une même coupe ou l'expression du phénotype pathologique à partir d'une biopsie. Le développement de ce type de technologie rappelle les travaux similaires effectués il y a plusieurs années en Résonance Magnétique Nucléaire et qui ont conduit au développement des techniques d'imagerie dites IRM dont l'implantation en milieu médical est considérable pour le diagnostic de certaines pathologies.

---

\* pour la correspondance

## **Abstract**

At the proteomics' time, the mass spectrometry appears to be a powerful tool to find and identify peptides and proteins from purified sample. A new era is now starting, with the MALDI imaging, allowing both research, identification while conserving the localization of biomolecules like peptides, proteins, oligonucleotides and lipids in tissues. Crucial developments at the level of the sample preparations are in progress, in view of tissue analysis or frozen conserved biopsies (-80°C) since more than 6 months or conserved after paraformaldehyde fixation and paraffin embedded. The technical problem allowing the access to hospital libraries is now off and the biomarkers hunting from tissues derived from these libraries is now an objective key. The relationship between MALDI imaging and immunocytochemistry used by pathologist is important. The development of specific MALDI imaging using probes with a tag (peptide or organic) called "Tag-Mass" open the field of new perspectives. In fact, it is possible henceforth to localize a protein with its specific mRNA and particularly the signalisation pathway on the same slices or a pathology expression phenotype from a biopsy. Development of such a technology is similar to the one that occurred several years ago for Nuclear Magnetic Resonance that leads the development of imaging technologies called MRI in hospital which is intensively used for pathologies diagnostic.

## Introduction

Les récents développements réalisés dans le domaine de la spectrométrie de masse et particulièrement sur le plan instrumental font actuellement de cette technique un outil de choix pour l'application aux recherches biologiques [1]. Les nombreux efforts menés ces dernières années pour développer l'étude du protéome ont démontré le rôle incontournable de cette technologie [2]. La combinaison des méthodes de biochimie classique à la spectrométrie de masse a déjà permis l'identification de nombreuses protéines aux travers d'études protéomiques associées aux connaissances croissantes des génomes [3]. Néanmoins de telles méthodes conduisent principalement à l'identification de nouvelles protéines et ne permettent pas d'obtenir d'informations sur la localisation de l'expression ou encore de suivre l'évolution du profil d'expression de l'ensemble des peptides/protéines au sein de tissus ou de groupes cellulaires voire même de cellule unique.

Les nouvelles technologies émergentes, permettent à l'heure actuelle de développer des méthodes alternatives pouvant répondre à ces questions. A ce titre, le développement d'analyse directe sur tissus ou sur cellules par Matrix Assisted Laser Desorption/Ionization time of flight (MALDI-TOF) est extrêmement prometteur [4]. En effet, de telles analyses permettent, tout en s'affranchissant des lourdes étapes d'extraction, purifications et séparation, d'obtenir directement les profils d'expression cellulaires sans perdre la localisation des composés au sein du tissu. Une telle stratégie d'étude s'adapte parfaitement à la spectrométrie de masse au travers de l'utilisation de sources MALDI. Le MALDI utilise une méthode d'ionisation parfaitement adaptée à l'étude des composés biologiques puisqu'elle permet l'analyse de composés polaires et de hautes masses moléculaires sous leur forme intacte, malgré leur fragilité thermique. Son principe de fonctionnement nécessite la formation d'un réseau cristallin qui sera par la suite soumis à des tirs laser. Ainsi, cette méthode permet l'introduction de matériaux sous leur forme solide et ne nécessite pas de disposer de composés en solution pour obtenir des ions en phase gazeuse. Elle est donc adaptable à l'introduction de matériel biologique, comme par exemple des cellules isolées ou des coupes de tissu. Les images sont obtenues par ionisation de l'échantillon grâce à un faisceau laser qui parcourt toute la coupe (**Figure 1**). Après acquisition des données, un logiciel permet de sélectionner des gammes de masse correspondant à des molécules d'intérêt et de reconstruire des images de localisation. Chaque point correspondra à 1 pixel de l'image et la distance entre chaque point définira la résolution latérale de l'image.

Cette technique ne se limite pas seulement à la localisation des peptides, mais peut aussi s'appliquer aux lipides, protéines et permet à partir d'une seule coupe l'obtention des

informations de localisation de ces différents composés. Dans ces conditions, il est possible d'analyser sur un même tissu les différentes zones le constituant de manière hautement spécifique.

## **I. De l'analyse directe de tissu par spectrométrie de masse vers l'Imagerie MALDI**

Depuis longtemps évoquée, la faisabilité de l'analyse directe de tissus biologiques par spectrométrie de masse de type MALDI-TOF a été démontrée pour la première fois en 1993 par l'équipe du Pr. van der Greef grâce à des travaux sur des neurones de *Lymnaea stagnalis* [5]. Il a fallu attendre 1996 et la reprise de ces travaux par le groupe de Jonathan Sweedler [6] puis les développements de l'équipe de RM Caprioli [7] pour que le profiling de tissu par spectrométrie de masse voie le jour. Les études menées par ce dernier groupe ont permis de montrer la faisabilité de la méthode pour suivre les profils d'expression des peptides/protéines directement au sein de tissus sans traitement préalable [8]. Les résultats obtenus par d'autres équipes ont porté sur la cartographie, la distribution spatiale ou la modulation, suite à des traitements, de la libération de neuropeptides dans le système nerveux de différents modèles d'études [9-11]. Sur des modèles invertébrés (*Aplysia californica* et *Lymnaea stagnalis*), des analyses directes ont été réalisées sur les cellules exprimant les gènes de l'hormone de ponte de ces deux modèles [12, 13]. Une caractérisation complète des mécanismes de maturation de cette hormone au niveau d'un seul neurone a pu être réalisée. Du point de vue clinique, cette technique de « peptides profiling » est en plein essor [4, 14-17]

Le développement récent de logiciels informatiques spécifiques a montré la possibilité de suivre l'expression d'un peptide/protéine au sein du tissu et d'obtenir ainsi des cartographies de l'expression de la molécule. L'imagerie par spectrométrie de masse consiste en une automatisation de l'analyse directe [7, 18, 19]. Tout comme pour l'analyse directe, le faisceau laser va irradier l'échantillon en un endroit, ce qui va définir un « spot » et un spectre de masse sera enregistré. Le plateau du support de l'échantillon va ensuite se déplacer d'une distance définie par avance, puis un nouveau spectre sera enregistré à l'endroit du nouveau spot. Le support va ainsi se déplacer à intervalles réguliers jusqu'à « scanner » toute la zone d'analyse préalablement choisie. Une fois toute la zone analysée, un logiciel va reconstruire une image de la répartition d'un rapport masse sur charge choisi (**Figure 1**). Trois éléments sont décisifs dans cette technique : 1) la préparation de l'échantillon, 2) la matrice et 3) son mode de dépôt.

## Développement

L'utilisation de l'imagerie MALDI au niveau clinique oblige les groupes à développer de nouveaux protocoles, de nouveaux supports ainsi que de nouvelles matrices compatibles avec les colorations histologiques et les procédures mises en œuvre par les pathologistes.

### 1. Supports conducteurs transparents

La comparaison des données histologiques, obtenues par l'observation des coupes après coloration en microscopie optique, aux images moléculaires, obtenues par spectrométrie de masse, montrent une très bonne corrélation et une compatibilité presque totale [20], une fois résolus les problèmes liés aux supports d'analyse. En effet, il est difficile de localiser beaucoup d'éléments tissulaires par microscopie optique quand les coupes sont montées sur un support opaque. C'est pourquoi il est devenu impératif que les tissus soient montés sur des supports transparents et conducteurs tels que des lames ITO, Nickel ou des lames classiques recouvertes d'or [21](figure 2). Effectivement la propriété conductrice des lames est très importante pour l'utilisation d'analyseurs temps de vol, car des différences de tension sur la surface de l'échantillon peuvent induire des variations dans la mesure des temps de vol des ions présents dans le spectromètre de masse, empêchant par la suite d'avoir accès aux analyses structurales par MS/MS. Cette caractérisation structurale des peptides au sein de la coupe par des études de la fragmentation des peptides directement à partir des tissus est un verrou technologique qui pourra être levé suite à l'arrivée d'instruments de type MALDI-TOF/TOF.

### 2. Préparation de l'échantillon et digestion trypsique.

Un des points critiques de l'imagerie, comme toutes les techniques biochimiques, est la conservation de l'échantillon, *i.e.* son mode et son temps de conservation. En effet, les tissus peuvent être, dès leur prélèvement, congelés puis coupés et ensuite analysés. A l'heure actuelle, la plupart des travaux utilisent cette procédure [22].

Dans le cas de coupes congelées ayant été conservées plus de 6 mois, des traitements par des solvants organiques peuvent être réalisés pour éliminer la couche de lipides qui saturent la coupe [23] et ainsi obtenir un meilleur signal. De plus la possibilité de faire de la digestion trypsique *in situ* sur coupes fraîches [24] par le biais ou non d'un transfert sur membrane [25] devra permettre l'accès à l'information moléculaire. Cependant, la plupart des échantillons

conservés par les pathologistes au sein des hôpitaux sont des tissus fixés au formaldéhyde puis inclus dans la paraffine. Le formaldéhyde forme au cours du temps des ponts méthylènes. Pour résoudre ce problème, des stratégies alliant l'emploi de matrice active [26] et la microdigestion à la trypsine déposée par microspotteur, ont été développées [27] (**Figure 3**). Ces stratégies sont couplées à la caractérisation par ESI-Trap des protéines issues d'une coupe de tissu adjacente ayant subi un déparaffinage puis une digestion enzymatique [27]. Le développement de ce type d'études et l'amélioration de cette méthode permettront directement, à partir des tissus prélevés, de déterminer les peptides exprimés dans une zone bien définie du tissu puis de les identifier en obtenant leur séquence en acides aminés.

### 3. La matrice

Un autre développement consiste en la mise au point de nouvelles matrices mieux adaptées aux contraintes de l'imagerie [26]. Pour réaliser une image moléculaire, les matrices doivent entraîner une délocalisation réduite des composés, et donc une cristallisation rapide et homogène sur le tissu [26]. De plus, la quantité de matériel présent sur la coupe est limitée, donc les nouvelles matrices doivent permettre une augmentation de la sensibilité du signal et supporter des fréquences de tir laser élevées (200Hz), ainsi que le vide poussé de la source, durant le temps nécessaire à la réalisation d'une image. Les matrices ioniques répondent parfaitement à l'ensemble de ces critères [28](figure 4).

### 4. Le dépôt de matrice

Le dépôt de matrice est un point important pour l'Imagerie MALDI. En effet la matrice doit extraire les analytes du tissu de façon verticale et des cristaux doivent être formés pour éviter la délocalisation des molécules. Plusieurs approches ont été testées, comme recouvrir toute la surface de la coupe avec de la matrice par application à la micropipette, déposer la matrice par spray pneumatique [16] ou par electrospray [7]. Ces méthodes peuvent s'avérer difficiles à mettre en place et à optimiser. Actuellement différents groupes de recherche s'orientent vers un dépôt de matrice localisé. Des microgouttes de matrices vont être déposées et espacées d'un pas régulier. Ces microgouttes sont obtenues par dépôts manuels ou automatiquement par des systèmes robotiques de micro dépôts, dépôt piezo, voire dépôt par éjection acoustique de microgouttes [29]. Ces micros dépôts de matrices vont permettre à la fois de limiter la délocalisation d'analytes mais aussi de réduire artificiellement la taille de la zone d'éjection de matière due à l'irradiation par le laser.

Tous ces développements en imagerie MALDI permettent la reconstruction de cartes en 2D. Si on étudie toute une série de coupes d'un organe, grâce à un procédé informatique [30], il est alors possible de reconstruire non plus une image en 2D mais en 3D. En corrélant ces résultats aux images de ces coupes obtenues en histologie, il est désormais possible de voyager à l'intérieur de l'organe et de faire une localisation précise d'une molécule dans celui-ci.

### **Perspectives**

Les études réalisées jusqu'ici utilisent la technologie de MALDI-UV. Des travaux sont en cours pour développer l'imagerie MALDI infrarouge [21]. En effet, les potentialités de l'imagerie MALDI, combinées aux récents développements dans le domaine de l'analyse MALDI avec des lasers IR permettent d'envisager le transfert de technologie déjà acquise à l'imagerie par MALDI-IR[31, 32]. Les potentialités de ce mode sont importantes puisque les mécanismes mis en jeu dans les processus conduisant à l'obtention, à partir d'un dépôt solide, d'ions en phase gazeuse reposent non pas, comme pour les lasers UV, sur des transitions électroniques, mais sur de l'excitation vibrationnelle, et principalement sur les vibrations des liaisons O-H et N-H. Ainsi, de nombreux composés sont potentiellement de bonnes matrices, comme par exemple l'eau. Ce mode autorise donc l'utilisation de matrices liquides ou solides se rapprochant des conditions physiologiques. De plus, de part la nature des processus physicochimiques mis en jeu en IR, cette méthode a permis d'obtenir des résultats prometteurs dans le cadre de l'analyse d'oligonucléotides et d'ADN.

Pour palier à ce manque en MALDI-UV, l'un des développements les plus prometteurs est ce que l'on appelle l'imagerie spécifique. Sous cette appellation se cache l'utilisation de sondes (anticorps, ribosondes, aptamers, lectines) marqués à l'aide d'un lien photoclivable et d'un peptide rapporteur [33]. Ce lien photoclivable est clivé à la longueur d'onde du laser du MALDI. L'utilisation de ces sondes spécifiques permet de réaliser des images en multiplexes d'ARNm spécifiques, d'associer la répartition d'un ligand avec son récepteur, ou d'un précurseur avec ses enzymes de maturation. Le concept du « Tag-mass » permet également d'étudier, après avoir identifié des biomarqueurs, la répartition et l'expression de ceux-ci au sein d'un tissu. La corrélation de leur expression avec celle de facteurs de transcription conduit à la localisation au sein du tissu des cellules saines, des cellules exprimant le phénotype tumeur ou de celles en cours de changement de phénotype (**Figure 5**).

## Conclusion

La comparaison des résultats provenant de l'imagerie par spectrométrie de masse avec ceux obtenus par des méthodes biologiques classiques telles que l'immunocytochimie confirment la fiabilité de la méthode puisque les images d'expressions obtenues sont très semblables. Ainsi, l'imagerie par spectrométrie de masse pourra par sa haute spécificité, sa rapidité et sa sensibilité être un nouvel outil d'investigation pour l'analyse et l'étude des biomolécules.

Le couplage de l'analyse du transcriptome et du peptidome directement sur des coupes de tissu ou au niveau d'une cellule unique est le prochain développement de l'analyse MALDI.

## Remerciements

Les auteurs remercient pour leur soutien financier le CNRS, le Ministère de la Recherche (ACI, I. Fournier), la région Nord-Pas de Calais (M. Wisztorski). Ils remercient également Mr. D. Maréchal (Eurogentec, Belgique), Mr W. Amoyal (Société Disruptive Technology Inc, France) et Mr G. Rubel (Bruker Daltonic, France) pour leur soutien et leur coopération dans ce projet.

## Bibliographie

1. Karas, M., *Matrix-assisted laser desorption ionization MS: a progress report*. Biochem Soc Trans, 1996. **24**(3): p. 897-900.
2. Aebersold, R. and D.R. Goodlett, *Mass spectrometry in proteomics*. Chem Rev, 2001. **101**(2): p. 269-95.
3. Kuster, B., et al., *Mass spectrometry allows direct identification of proteins in large genomes*. Proteomics, 2001. **1**(5): p. 641-50.
4. Chaurand, P., M. Stoeckli, and R.M. Caprioli, *Direct profiling of proteins in biological tissue sections by MALDI mass spectrometry*. Anal Chem, 1999. **71**(23): p. 5263-70.
5. P. A. van Veelen, C.R.J., K. W. Li, W. C. Wildering, W. P. M. Geraerts, U. R. Tjaden, J. van der Greef, *Direct Peptide profiling of Single Neurons by Matrix-Assisted Laser Desorption Ionization Mass Spectrometry*. Organic Mass Spectrometry, 1993. **28**(12): p. 1542-1546.
6. Garden, R.W., et al., *Excess salt removal with matrix rinsing: direct peptide profiling of neurons from marine invertebrates using matrix-assisted laser desorption/ionization time-of-flight mass spectrometry*. J Mass Spectrom, 1996. **31**(10): p. 1126-30.
7. Caprioli, R.M., T.B. Farmer, and J. Gile, *Molecular imaging of biological samples: localization of peptides and proteins using MALDI-TOF MS*. Anal Chem, 1997. **69**(23): p. 4751-60.
8. Caldwell, R.L. and R.M. Caprioli, *Tissue profiling by mass spectrometry: a review of methodology and applications*. Mol Cell Proteomics, 2005. **4**(4): p. 394-401.

9. Fournier, I., R. Day, and M. Salzet, *Direct analysis of neuropeptides by in situ MALDI-TOF mass spectrometry in the rat brain*. Neuro Endocrinol Lett, 2003. **24**(1-2): p. 9-14.
10. Kruse, R. and J.V. Sweedler, *Spatial profiling invertebrate ganglia using MALDI MS*. J Am Soc Mass Spectrom, 2003. **14**(7): p. 752-9.
11. Sweedler, J.V., et al., *Mass spectrometric survey of peptides in cephalopods with an emphasis on the FMRFamide-related peptides*. J Exp Biol, 2000. **203**(Pt 23): p. 3565-73.
12. Garden, R.W., et al., *Proteolytic processing of the Aplysia egg-laying hormone prohormone*. Proc Natl Acad Sci U S A, 1998. **95**(7): p. 3972-7.
13. Li, L., et al., *Egg-laying hormone peptides in the aplysiidae family*. J Exp Biol, 1999. **202**(Pt 21): p. 2961-73.
14. Reyzer, M.L. and R.M. Caprioli, *MALDI mass spectrometry for direct tissue analysis: a new tool for biomarker discovery*. J Proteome Res, 2005. **4**(4): p. 1138-42.
15. Reyzer, M.L., et al., *Direct analysis of drug candidates in tissue by matrix-assisted laser desorption/ionization mass spectrometry*. J Mass Spectrom, 2003. **38**(10): p. 1081-92.
16. Schwartz, S.A., M.L. Reyzer, and R.M. Caprioli, *Direct tissue analysis using matrix-assisted laser desorption/ionization mass spectrometry: practical aspects of sample preparation*. J Mass Spectrom, 2003. **38**(7): p. 699-708.
17. Meistermann, H., et al., *Biomarker discovery by imaging mass spectrometry: Transthyretin is a biomarker for gentamicin-induced nephrotoxicity in rat*. Mol Cell Proteomics, 2006.
18. Stoeckli, M., et al., *Imaging mass spectrometry: a new technology for the analysis of protein expression in mammalian tissues*. Nat Med, 2001. **7**(4): p. 493-6.
19. Stoeckli, M., T.B. Farmer, and R.M. Caprioli, *Automated mass spectrometry imaging with a matrix-assisted laser desorption ionization time-of-flight instrument*. J Am Soc Mass Spectrom, 1999. **10**(1): p. 67-71.
20. Chaurand, P., et al., *Integrating histology and imaging mass spectrometry*. Anal Chem, 2004. **76**(4): p. 1145-55.
21. Wisztorski M, B.L., Dreiserwer K, Hillenkamp F, Berkenkamp S, Salzet M, Fournier I, *Effect of metals coating for YV MALDI-a-TOF mass spectrometry imaging (MALDI MSI) and direct tissue analysis in UV/IR MALDI-o-TOF mass spectrometry*. Proceedings of 54rd ASMS conference on Mass Spectrometry, Seattle, Washington, May 30-June 2, 2006.
22. Chaurand, P., D.S. Cornett, and R.M. Caprioli, *Molecular imaging of thin mammalian tissue sections by mass spectrometry*. Curr Opin Biotechnol, 2006.
23. Lemaire R, W.M., Tabet JC, Day R, Salzet M, Fournier I, *Organic treatments : a way for improving signal sensitivity for profiling on tissue* Anal Chem, 2006: p. in press.
24. Aerni HR, C.D., Caprioli RM. *In situ proteomics in artificially created tissue microwells*. in Proceedings of 53rd ASMS conference on Mass Spectrometry, San Antonio, Texas, June 5-9. 2005.
25. Rohner, T.C., D. Staab, and M. Stoeckli, *MALDI mass spectrometric imaging of biological tissue sections*. Mech Ageing Dev, 2005. **126**(1): p. 177-85.
26. Lemaire R, D.P., Wisztorski M, Hendra JB, Tabet JC, Salzet M, Fournier I. *Exploring Direct Analysis Using Ionic Matrices*. in Proceedings of 53rd ASMS conference on Mass Spectrometry, San Antonio, Texas, June 5-9. 2005.
27. Lemaire R, D.A., Ducroy P, Tabet JC, Salzet M, Fournier I, *Direct analysis and MALDI imaging on formalin fixed paraffin embedded tissue (FFPE): Application to*

- Parkinson disease*. Proceedings of 54rd ASMS conference on Mass Spectrometry, Seattle, Washington, May 30-June 2, 2006.
28. Lemaire, R., et al., *Solid ionic matrixes for direct tissue analysis and MALDI imaging*. Anal Chem, 2006. **78**(3): p. 809-19.
  29. Aerni, H.R., D.S. Cornett, and R.M. Caprioli, *Automated acoustic matrix deposition for MALDI sample preparation*. Anal Chem, 2006. **78**(3): p. 827-34.
  30. Crecelius, A.C., et al., *Three-dimensional visualization of protein expression in mouse brain structures using imaging mass spectrometry*. J Am Soc Mass Spectrom, 2005. **16**(7): p. 1093-9.
  31. Berkenkamp, S., M. Karas, and F. Hillenkamp, *Ice as a matrix for IR-matrix-assisted laser desorption/ionization: mass spectra from a protein single crystal*. Proc Natl Acad Sci U S A, 1996. **93**(14): p. 7003-7.
  32. Kraft, P., et al., *Infrared, surface-assisted laser desorption ionization mass spectrometry on frozen aqueous solutions of proteins and peptides using suspensions of organic solids*. J Am Soc Mass Spectrom, 1998. **9**(9): p. 912-24.
  33. Fournier I, L.R., Wisztorski M, Stauber J, Jardin-Mathè O, Van Camp C, Deschamps M, Proess G, Day R, Salzert M, *New insight in sample preparation for MALDI imaging and new developments to approach specific MALDI imaging of the transcriptome*. Proceedings of 54rd ASMS conference on Mass Spectrometry, Seattle, Washington, May 30-June 2, 2006.

## Légendes des figures

Figure 1 : Principe de l'imagerie MALDI-TOF.

- a) Une coupe congelée est déposée sur un support. Une analyse microscopique est effectuée. La coupe est recouverte de matrice avant d'être analysée en MALDI-TOF.
- b) Le laser va balayer la surface de la coupe et un logiciel informatique va permettre de reconstituer une image bidimensionnelle de la répartition des molécules. Une superposition entre la carte de répartition des molécules et l'observation microscopique peut être réalisée pour obtenir une localisation précise de la molécule dans le tissu.

Figure 2 : a) Image par microscopie optique d'une coupe de ganglion de sangsue déposée sur une lame conductrice recouverte de nickel. b) Analyse directe de la coupe en mode réflecteur. c) Mode PSD de l'ion  $m/z$  861.7

Figure 3 : Schémas des étapes permettant l'analyse par spectrométrie de masse de type MALDI-TOF de tissus fixés et paraffinés dépendantes du temps de stockage de l'échantillon

Figure 4 : Images moléculaires reconstruites de la répartition de différents ions au sein de la coupe de cerveau de rat obtenue par balayage de 10000 points à 50 Hz et 200 tirs par position pour la matrice ionique HCCA/ANI a) 1ère acquisition en mode positif et b) seconde acquisition en mode négatif (avec la permission de Analytical Chemistry)

Figure 5 : Principe de détection de plusieurs ARNm sur une coupe de tissu en utilisant des sondes marquées à l'aide d'UTP-lié par une liaison photoclivable par le laser UV et un rapporteur peptidique.

Figure 6 : Principe de l'imagerie spécifique en utilisant des anticorps taggés (lies à une liaison photoclivable par le laser du MALDI et comportant un peptide rapporteur). a) répartition moléculaire de la carboxypeptidase D (protéine membranaire de 160 kDa) détectée par le concept du Tag-mass, b) scanner de la coupe adjacente c) immunodétection avec un anticorps marqué à la peroxydase et immunorévéle par du chloronaphtol.

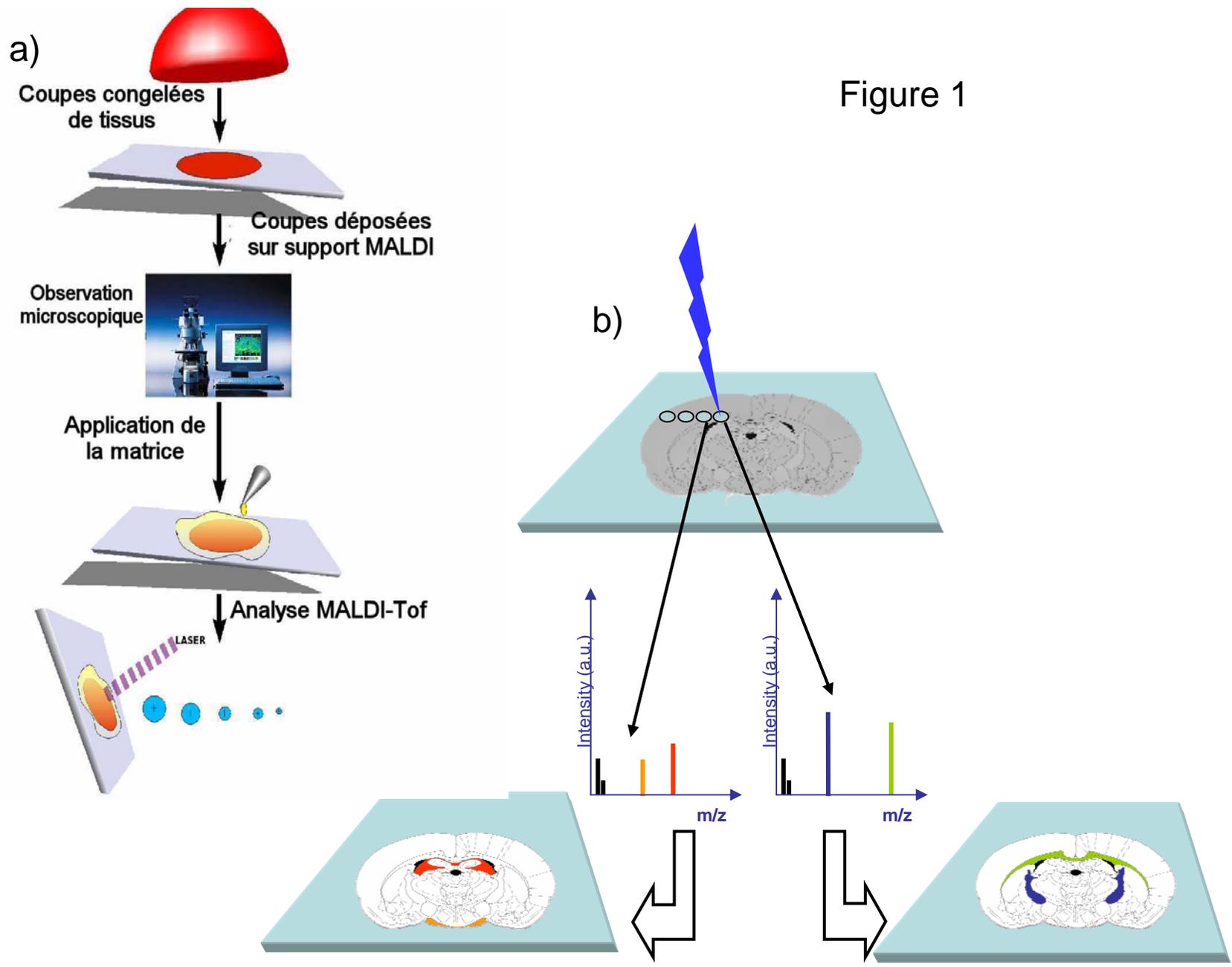


Figure 1

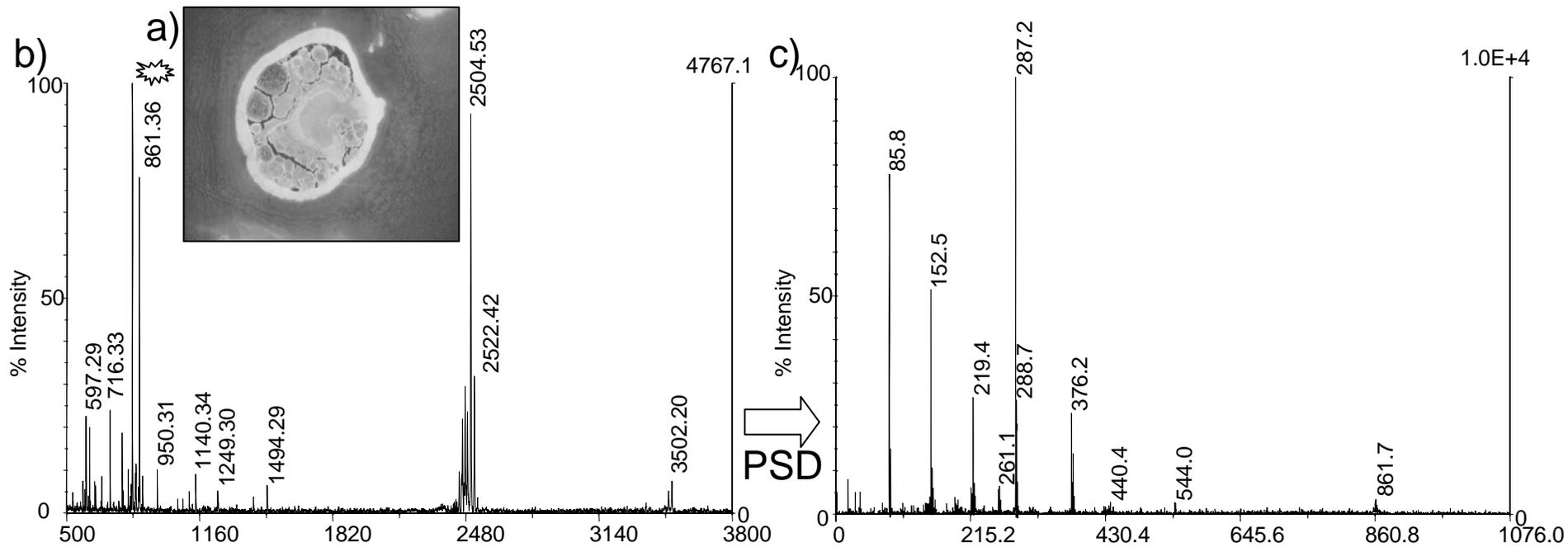


Figure 2:

Figure 3:

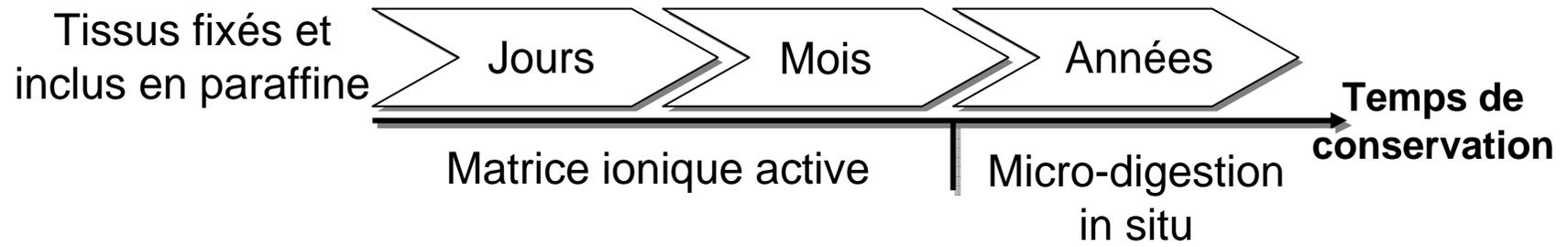
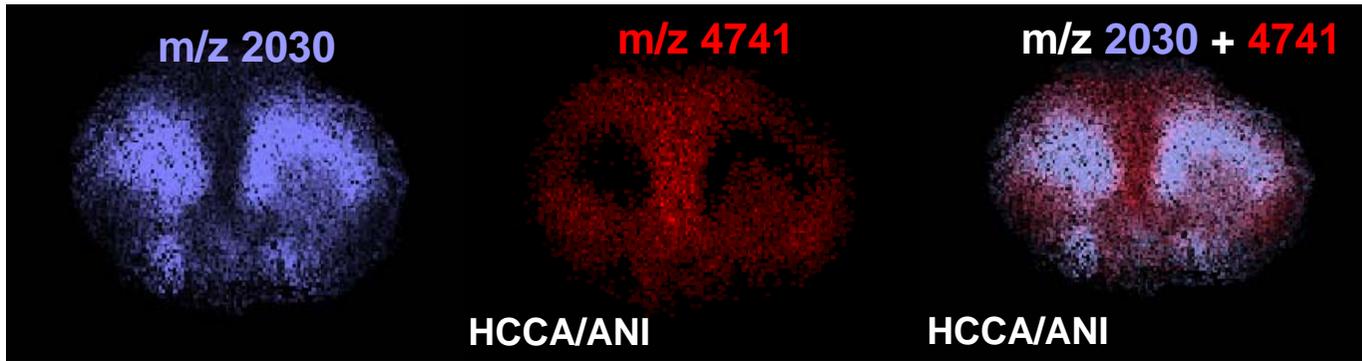


Figure 4

a) Mode positif



b) Mode négatif

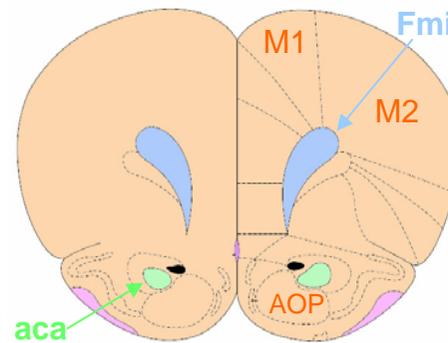
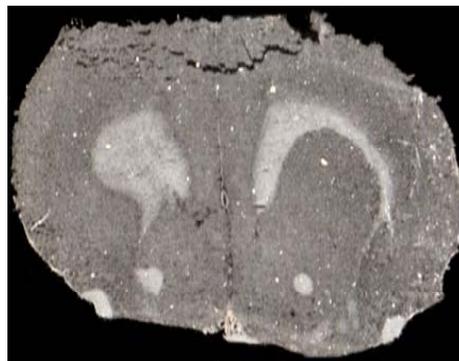
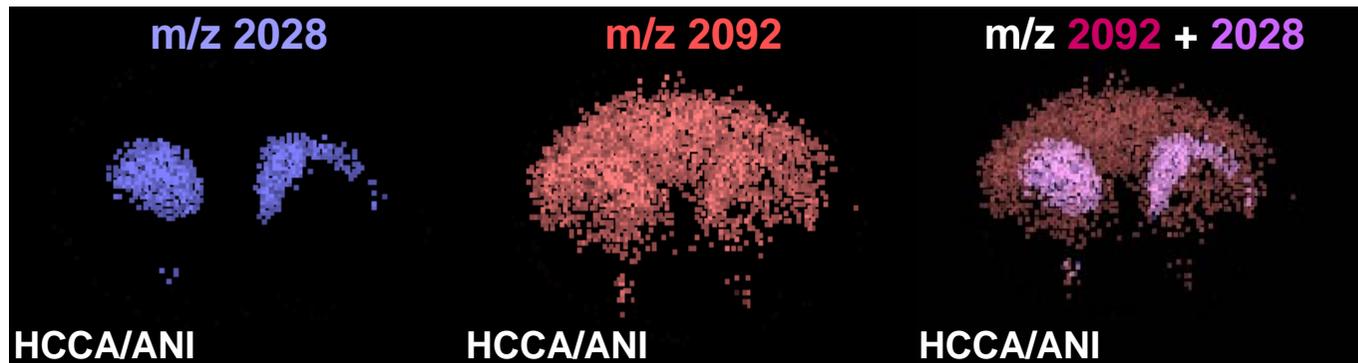


Figure 5:

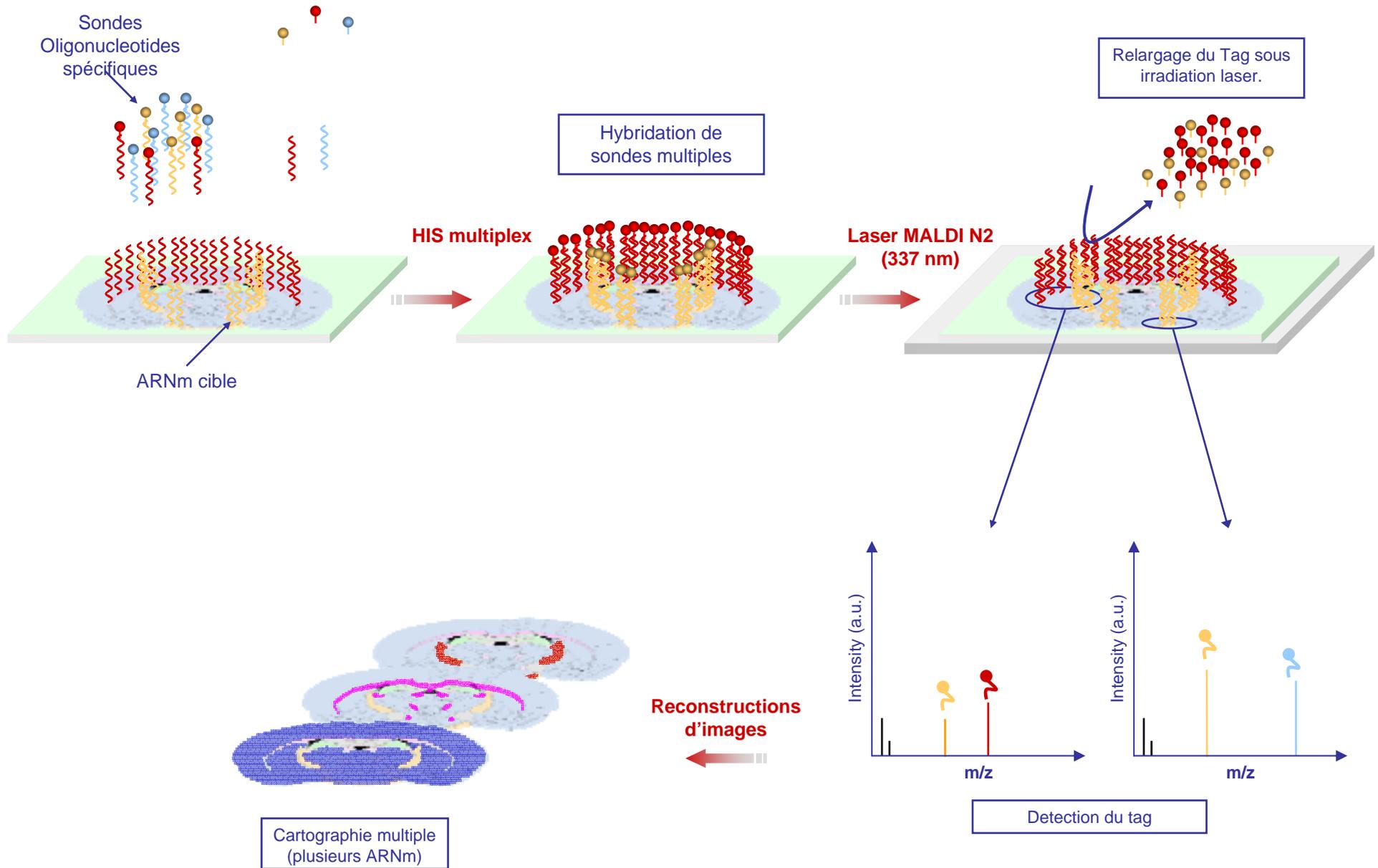
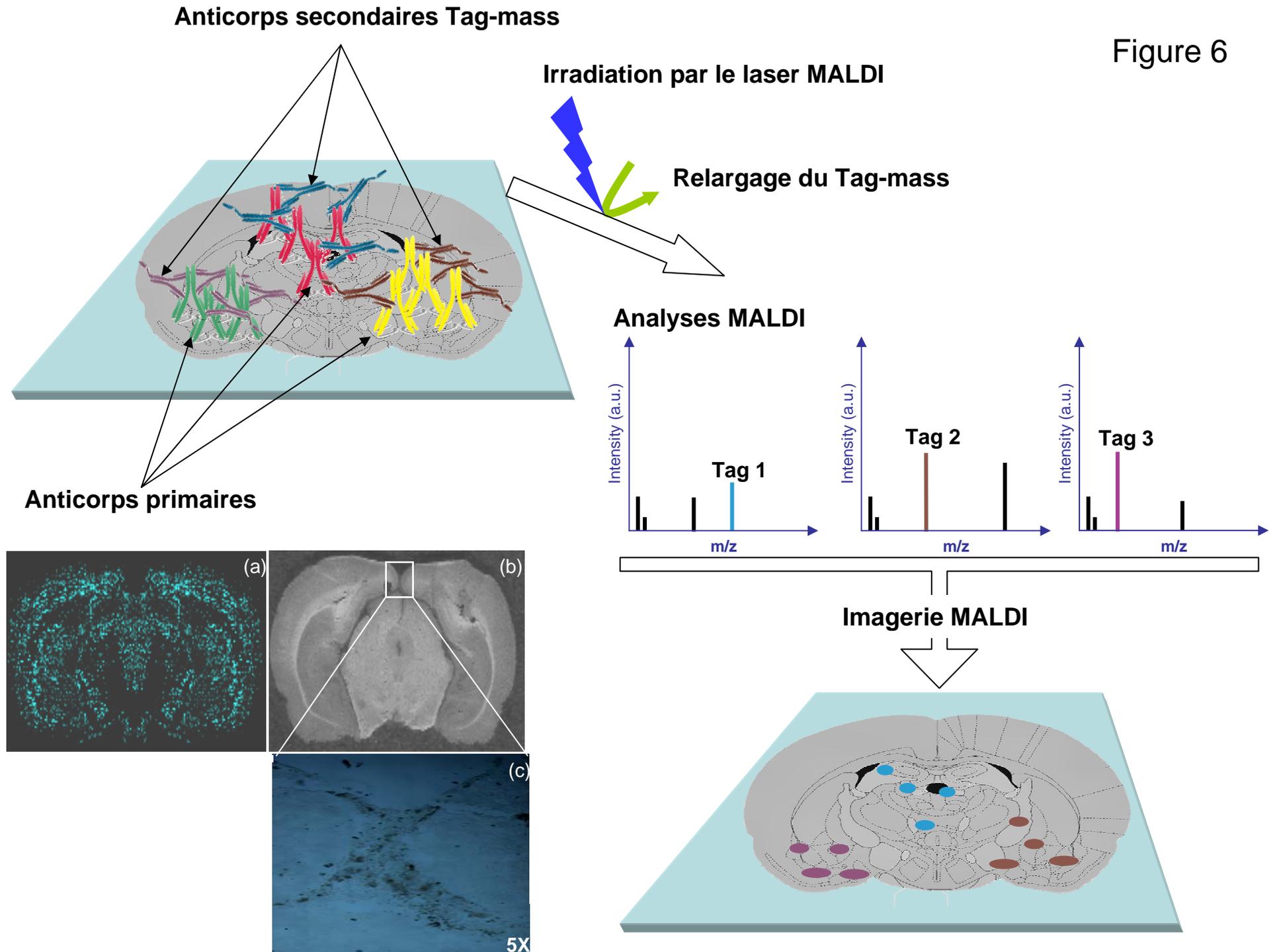


Figure 6



## AssemblyInfo.cpp

```
#include "stdafx.h"

using namespace System;
using namespace System::Reflection;
using namespace System::Runtime::CompilerServices;
using namespace System::Runtime::InteropServices;
using namespace System::Security::Permissions;

//
// General Information about an assembly is controlled through the following
// set of attributes. Change these attribute values to modify the information
// associated with an assembly.
//
[assembly: AssemblyTitle("MITICS")];
[assembly: AssemblyDescription("")] ;
[assembly: AssemblyConfigurationAttribute("")] ;
[assembly: AssemblyCompany("")] ;
[assembly: AssemblyProduct("MITICS")];
[assembly: AssemblyCopyright("Copyright (c) 2008")];
[assembly: AssemblyTrademarkAttribute("")] ;
[assembly: AssemblyCultureAttribute("")] ;

//
// Version information for an assembly consists of the following four values:
//
//      Major Version
//      Minor Version
//      Build Number
//      Revision
//
// You can specify all the value or you can default the Revision and Build
Numbers
// by using the '*' as shown below:

[assembly: AssemblyVersion("1.0.*")];

[assembly: ComVisible(false)];

[assembly: CLSCompliantAttribute(true)];

[assembly: SecurityPermission(SecurityAction::RequestMinimum, UnmanagedCode =
true)];
```

## Coordonnees.cpp

```
////////////////////////////////////
// Coordonnees.cpp
// Implementation of the Class Coordonnees
// Created on:      18-août-2004 13:40:46
////////////////////////////////////
#include "stdafx.h"
#include "Coordonnees.h"
```

```

Coordonnees::Coordonnees(){
}

Coordonnees::~~Coordonnees(){
}

void Coordonnees::SetX(double abscisse){
this->X = abscisse;
}

void Coordonnees::SetY(double ordonnee){
this->Y = ordonnee;
}

double Coordonnees::GetX(){
return this->X;
}

double Coordonnees::GetY(){
return this->Y;
}

```

## **EnsSpectre.cpp**

```

////////////////////////////////////
//  EnsSpectre.cpp
//  Implementation of the Class EnsSpectre
//  Created on:      18-août-2004 13:40:47
////////////////////////////////////
#include "stdafx.h"
#include "EnsSpectre.h"

EnsSpectre::EnsSpectre(){
}

EnsSpectre::~~EnsSpectre(){
}

void EnsSpectre::SetGraphe(Spectre graphe){

```

```

this->Graphe = graphe;
}

void EnsSpectre::SetIdPlaque(int identifiant){
this->IdPlaque = identifiant;
}

Spectre EnsSpectre::GetGraphe(){
return this->Graphe;
}

int EnsSpectre::GetIdPlaque(){
return this->IdPlaque;
}

//Masse EnsSpectre::RechercherAire(EnsSpectre ensemble_spectres, double
masse){

//}

//Masse EnsSpectre::RechercherIntensite(EnsSpectre ensemble_spectres, double
masse){

//}

```

## **FichierBIC.cpp**

```

////////////////////////////////////
// FichierBIC.cpp
// Implementation of the Class FichierBIC
// Created on: 19-juil.-2004 15:15:37
////////////////////////////////////

#include <iostream>
#include <fstream>
#include <string>
#include "stdafx.h"
#include "FichierBIC.h"

using namespace std;

FichierBIC::FichierBIC(){

this->LaserIntensity = 1450;
this->AcceleratingVoltage = 20000;
this->GridVoltage = 94;
this->GuideWireVoltage0 = 0.05;
this->DelayTime = 200;
this->TotalLaserShots = 50;

```

```
this->AcqMassMin = 500;
this->AcqMassMax = 6000;
this->LowMassGate = 1;
this->LowMassGateMass = 500;
}
```

```
FichierBIC::~FichierBIC(){
}
```

```
void FichierBIC::SetLaserIntensity(int intensite_laser){
this->LaserIntensity = intensite_laser;
}
```

```
void FichierBIC::SetAcceleratingVoltage(int acceleration){
this->AcceleratingVoltage = acceleration;
}
```

```
void FichierBIC::SetGridVoltage(double voltage){
this->GridVoltage = voltage;
}
```

```
void FichierBIC::SetGuideWireVoltage0(double voltage0){
this->GuideWireVoltage0 = voltage0;
}
```

```
void FichierBIC::SetDelayTime(int temps){
this->DelayTime = temps;
}
```

```
void FichierBIC::SetTotalLaserShots(int nb_tirs){
this->TotalLaserShots = nb_tirs;
}
```

```
void FichierBIC::SetAcqMassMin(int masse_min){
this->AcqMassMin = masse_min;
}
```

```
void FichierBIC::SetAcqMassMax(int masse_max){
this->AcqMassMax = masse_max;
}
```

```
void FichierBIC::SetLowMassGate(int coche){
this->LowMassGate = coche;
}
```

```
void FichierBIC::SetLowMassGateMass(double masse_mini){  
this->LowMassGateMass = masse_mini;  
}
```

```
void FichierBIC::SetLieuBIC(string enregistrement){  
this->LieuBIC = enregistrement;  
}
```

```
int FichierBIC::GetLaserIntensity(){  
return this->LaserIntensity;  
}
```

```
int FichierBIC::GetAcceleratingVoltage(){  
return this->AcceleratingVoltage;  
}
```

```
double FichierBIC::GetGridVoltage(){  
return this->GridVoltage;  
}
```

```
double FichierBIC::GetGuideWireVoltage0(){  
return this->GuideWireVoltage0;  
}
```

```
int FichierBIC::GetDelayTime(){  
return this->DelayTime;  
}
```

```
int FichierBIC::GetTotalLaserShots(){  
return this->TotalLaserShots;  
}
```

```
int FichierBIC::GetAcqMassMin(){  
return this->AcqMassMin;  
}
```

```
int FichierBIC::GetAcqMassMax(){  
return this->AcqMassMax;  
}
```

```
int FichierBIC::GetLowMassGate(){  
return this->LowMassGate;  
}
```

```
double FichierBIC::GetLowMassGateMass(){
```

```

return this->LowMassGateMass;
}

string FichierBIC::GetLieuBIC(){
return this->LieuBIC;
}

void FichierBIC::Sauvegarder(string NomFichier){
ofstream dest(NomFichier.c_str());

dest << "ControlPanel\\PSDDecrementRatio=0.75" << endl;
dest << "ControlPanel\\PSDTrackGuideWire=0" << endl;
dest << "ControlPanel\\OperationMode=0 " << endl;
dest << "ControlPanel\\ExtractionType=1 " << endl;
dest << "ControlPanel\\PolarityType=0" << endl;
dest << "ControlPanel\\LaserType=0 " << endl;
dest << "ControlPanel\\LaserRateType=0" << endl;
dest << "ControlPanel\\MirrorVoltageRatio=1.12" << endl;
dest << "ControlPanel\\AutomatedAcquisition=1" << endl;
dest << "ControlPanel\\DigitizerLinearBinSize=4." << endl;
dest << "ControlPanel\\DigitizerLinearVerticalScale=200." << endl;
dest << "ControlPanel\\DigitizerLinearVerticalOffset=0." << endl;
dest << "ControlPanel\\DigitizerLinearInputBandwidth=20. " << endl;
dest << "ControlPanel\\DigitizerReflectorBinSize=2." << endl;
dest << "ControlPanel\\DigitizerReflectorVerticalOffset=0." << endl;
dest << "ControlPanel\\PSDMirrorVoltageRatio0=1." << endl;
dest << "ControlPanel\\PSDMirrorVoltageRatioCount=1" << endl;
dest << "ControlPanel\\DigitizerReflectorVerticalScale0=200." << endl;
dest << "ControlPanel\\DigitizerReflectorVerticalScaleCount=1" << endl;
dest << "ControlPanel\\DigitizerReflectorInputBandwidth0=20." << endl;
dest << "ControlPanel\\DigitizerReflectorInputBandwidthCount=1" << endl;
dest << "ControlPanel\\AcceleratingVoltage=";

```

```
dest << this->AcceleratingVoltage << endl;

dest << "ControlPanel\\GridVoltage= ";
dest << this->GridVoltage << endl;

dest << "ControlPanel\\GuideWireVoltage0= " ;
dest << scientific << this->GuideWireVoltage0 << endl;

dest << "ControlPanel\\GuideWireVoltageCount=1" << endl;

dest << "ControlPanel\\DelayTime=";
dest << this->DelayTime << endl;

dest << "ControlPanel\\TimedIonSelector=0 " << endl;

dest << "ControlPanel\\TimedIonSelectorMass=1000." << endl;

dest << "ControlPanel\\MSMSEinzell1=0." << endl;
dest << "ControlPanel\\MSMSEinzell2=0." << endl;
dest << "ControlPanel\\MSMSInitCell=0." << endl;
dest << "ControlPanel\\MSMSFinalCell=0. " << endl;
dest << "ControlPanel\\MSMSFinalSource=0." << endl;
dest << "ControlPanel\\MSMSX1Deflector=0." << endl;
dest << "ControlPanel\\MSMSX2Deflector=0." << endl;
dest << "ControlPanel\\MSMSY1Deflector=0." << endl;
dest << "ControlPanel\\MSMSY2Deflector=0." << endl;
dest << "ControlPanel\\MSMSGridVoltage=0." << endl;
dest << "ControlPanel\\MSMSLinearBias=0." << endl;
dest << "ControlPanel\\MSMSReflectorBias=0." << endl;
dest << "ControlPanel\\MSMSLinearGain=0." << endl;
dest << "ControlPanel\\MSMSReflectorGain=0." << endl;
dest << "ControlPanel\\MSMSDelayTime=0." << endl;

dest << "ControlPanel\\LaserIntensity=" ;
dest << this->LaserIntensity << endl;

dest << "ControlPanel\\LaserAutomation=0" << endl;

dest << "ControlPanel\\LaserAutoMinIntensity=1000" << endl;
dest << "ControlPanel\\LaserAutoMaxIntensity=2000" << endl;
```

```
dest << "ControlPanel\\LaserAutoIntensityStepSize=10" << endl;
dest << "ControlPanel\\LaserAutoIntensityPrescan=0" << endl;
dest << "ControlPanel\\LaserAutoIntensityPrescanStart=0" << endl;
dest << "ControlPanel\\LaserAutoIntensityStopAfterFails=0 " << endl;
dest << "ControlPanel\\LaserAutoIntensityNumFails=5" << endl;
dest << "ControlPanel\\SamplePositioningAutomation=0" << endl;
dest << "ControlPanel\\AutomatedSamplePositioningType=0" << endl;
dest << "ControlPanel\\AutomatedSamplePositioningRandomType=0" << endl;
dest << "ControlPanel\\SearchPatternFile=C:\\VOYAGER\\DEFAULT.SP " << endl;
dest << "ControlPanel\\SearchPatternNumPos=0" << endl;
dest << "ControlPanel\\AutomatedStorageMode=0 " << endl;
dest << "ControlPanel\\NumSpectra=1" << endl;
dest << "ControlPanel\\StopAfterConsecutiveFails=0" << endl;
dest << "ControlPanel\\NumberFails=5" << endl;
dest << "ControlPanel\\SaveBestAfterFails=0" << endl;
dest << "ControlPanel\\TotalLaserShots=" ;
dest << this->TotalLaserShots << endl;
dest << "ControlPanel\\AcqMassMin=" ;
dest << this->AcqMassMin << endl;
dest << "ControlPanel\\AcqMassMax=" ;
dest << this->AcqMassMax << endl;
dest << "ControlPanel\\LowMassGate=" ;
dest << this->LowMassGate << endl;
dest << "ControlPanel\\LowMassGateMass=" ;
dest << this->LowMassGateMass << endl;
dest << "ControlPanel\\CalibrationType=0" << endl;
dest << "ControlPanel\\CalibrationExternalFile=" << endl;
dest << "ControlPanel\\CalibrationMatrix=a-Cyano-4-hydroxycinnamic acid" <<
endl;
dest << "ControlPanel\\CalibrationMatrixVelocity=300." << endl;
dest << "ControlPanel\\CalibrationType_PSD=0" << endl;
dest << "ControlPanel\\CalibrationExternalFile_PSD=" << endl;
```

```

dest << "ControlPanel\\AutoAcqMinIntensityEnabled=0" << endl;
dest << "ControlPanel\\AutoAcqMinIntensity=0." << endl;
dest << "ControlPanel\\AutoAcqMaxIntensityEnabled=0 " << endl;
dest << "ControlPanel\\AutoAcqMaxIntensity=0." << endl;
dest << "ControlPanel\\AutoAcqResolutionEnabled=0" << endl;
dest << "ControlPanel\\AutoAcqResolution=0." << endl;
dest << "ControlPanel\\AutoAcqSNREnabled=0" << endl;
dest << "ControlPanel\\AutoAcqSNR=0." << endl;
dest << "ControlPanel\\AutoAcqSNRBaselineMassMin=2000." << endl;
dest << "ControlPanel\\AutoAcqSNRBaselineMassMax=100000." << endl;
dest << "ControlPanel\\AutoAcqMassMin=2000." << endl;
dest << "ControlPanel\\AutoAcqMassMax=100000." << endl;
dest << "ControlPanel\\PSDLaserIncrement0=0." << endl;
dest << "ControlPanel\\PSDLaserIncrementCount=1" << endl;
}

```

## **FichierPLT.cpp**

```

////////////////////////////////////
// FichierPLT.cpp
// Implementation of the Class FichierPLT
// Created on:      18-août-2004 13:40:48
////////////////////////////////////
#include "stdafx.h"
#include <math.h>
#include "FichierPLT.h"

#include <iostream>
#include <fstream>
#include <string>

FichierPLT::FichierPLT(){

}

FichierPLT::~FichierPLT(){

```

```
}
```

```
int is_even (int nombre){  
return (nombre%2)==0;  
}
```

```
void FichierPLT::SetNumeros(PointImage *points){  
this->Numeros = points;  
}
```

```
void FichierPLT::SetDep(Coordonnees *point_depart){  
this->Dep = point_depart;  
}
```

```
void FichierPLT::SetAbsDep(double abscisse){  
this->AbsDep = abscisse;  
}
```

```
void FichierPLT::SetOrdDep(double ordonnee){  
this->OrdDep = ordonnee;  
}
```

```
void FichierPLT::SetArr(Coordonnees *point_arrivee){  
this->Arr = point_arrivee;  
}
```

```
void FichierPLT::SetAbsArr(double abscisse){  
this->AbsArr = abscisse;  
}
```

```
void FichierPLT::SetOrdArr(double ordonnee){  
this->OrdArr = ordonnee;  
}
```

```
void FichierPLT::SetPasH(double pas){  
this->PasH = pas;  
}
```

```
void FichierPLT::SetPasV(double pas){  
this->PasV = pas;  
}
```

```
void FichierPLT::SetNbPtsH(int nb_points_horiz){  
this->NbPtsH = nb_points_horiz;  
}
```

```
void FichierPLT::SetNbPtsV(int nb_points_vert){  
this->NbPtsV = nb_points_vert;  
}
```

```

}

void FichierPLT::SetTrajetSimple(int trajet_simple){
this->TrajetSimple = trajet_simple;
}

void FichierPLT::SetLieuPLT(string nom_fichier){
this->LieuPLT = nom_fichier;
}

//PointImage FichierPLT::GetNumeros(){
//return this->Numeros;
//}

//Coordonnees FichierPLT::GetDep(){
//return this->Dep;
//}

double FichierPLT::GetAbsDep(){
//return this->AbsDep;
return this->Dep->GetX();
}

double FichierPLT::GetOrdDep(){
//return this->OrdDep;
return this->Dep->GetY();
}

//Coordonnees FichierPLT::GetArr(){
//return this->Arr;
//}

double FichierPLT::GetAbsArr(){
//return this->AbsArr;
return this->Arr->GetX();
}

double FichierPLT::GetOrdArr(){
//return this->OrdArr;
return this->Arr->GetY();
}

double FichierPLT::GetPasH(){
return this->PasH;
}

double FichierPLT::GetPasV(){
return this->PasV;
}

int FichierPLT::GetNbPtsH(){

```

```
return this->NbPtsH;
}
```

```
int FichierPLT::GetNbPtsV(){
return this->NbPtsV;
}
```

```
string FichierPLT::GetLieuPLT(){
return this->LieuPLT;
}
```

```
double FichierPLT::CalculNbPtsLigne(Coordonnees *point_arrivee, Coordonnees
*point_depart, double pas_horiz){
double nombre ;// = 0;
nombre = ((point_arrivee->GetX()) - (point_depart->GetX()));
// nombre = int(nombre / pas_horiz);
nombre = nombre / (double(pas_horiz));
return ceil(nombre)+1;
// return nombre;
}
```

```
double FichierPLT::CalculNbPtsColonne(Coordonnees *point_arrivee, Coordonnees
*point_depart, double pas_vert){
double nombre ;// = 0;
nombre = ((point_depart->GetY()) - (point_arrivee->GetY()));
// nombre = int(nombre / pas_vert);
nombre = nombre / (double(pas_vert));
return ceil(nombre)+1;
// return nombre;
}
```

```
double FichierPLT::CalculPasLigne(Coordonnees *point_arrivee, Coordonnees
*point_depart, int nb_points_ligne){
double pas;
// pas = 0;
pas = (point_arrivee->GetX()) - (point_depart->GetX());
// pas = int(pas / nb_points_ligne);
pas = pas / (double(nb_points_ligne-1));
return pas;
}
```

```
double FichierPLT::CalculPasColonne(Coordonnees *point_arrivee, Coordonnees
*point_depart, int nb_points_colonne){
double pas;
// pas = 0;
pas = (point_depart->GetY()) - (point_arrivee->GetY());
// pas = int(pas / nb_points_colonne);
pas = pas / (double(nb_points_colonne-1));
return pas;
}
```

```

void FichierPLT::Sauvegarder(string nom_fichier)
{

ofstream dest (nom_fichier.c_str());
dest << "wellunits=2" << endl;
// define the units of the sample positions that are displayed in the Manual
Laser/Sample Positionning control page.
// Allowed values: 1 = Steps (default)
//                2 = Microns
dest << "wellshape=1" << endl;
// defines the shape of the sample positions that are displayed in the Manual
Laser/Sample Positionning control page.
// allowed values: 1 = Elliptical (default)
//                2 = Rectangular
dest << "wellwidth=150.00" << endl;
// defines the width of the sample positions that are displayed in the Manual
Laser/Sample Positionning control page.
// default values: default step value = 800
//                default micron value = 2,540
dest << "wellheight=150.00" << endl;
// defines the height of the sample positions that are displayed in the Manual
Laser/Sample Positionning control page.
// default values: default step value = 800
//                default micron value = 2,540
dest << "welldepth=0.00" << endl;
//depth (mm) of indentation of a non-flat plate from the surface of the plate
to the bottom of the indentation.

int position = 1;
double a = this->AbsDep;
double b = this->OrdDep;
int ligne = 1;
int i;
//dest << "AbsDep=" << this->AbsDep << " et NbPtsH=" << this->NbPtsH << " et
PasH=" << this->PasH << " donc this->AbsDep*this->NbPtsH*this->PasH=" << this-
>AbsDep*this->NbPtsH*this->PasH << " et AbsArr=" << this->AbsArr << endl;
//if((this->AbsDep*this->NbPtsH*this->PasH)<this->AbsArr )
//{
//    this->NbPtsH++;
//}
//dest << "OrdArr=" << b << " et NbPtsV=" << this->NbPtsV << " et PasV=" <<
this->PasV << " donc this->OrdArr*this->NbPtsV*this->PasV=" << this-
>OrdArr*this->NbPtsV*this->PasV << " et OrdDep=" << this->OrdDep << endl;
//if((this->OrdArr*this->NbPtsV*this->PasV)<this->OrdDep )
//{
//    this->NbPtsV++;
// }

for ( int j = this->NbPtsV ; j >= 1 ; j--)
{
    // réinitialisation de a selon le trajet
    if ( this->TrajetSimple )
    { // si on va toujours de gauche à droite a reprend la valeur de X de
départ
        a = this->AbsDep;
    }
}

```

```

        else
        { // si on va une fois de gauche a droite et une fois de droite a
gauche,
            if (is_even(ligne))
            {
                a = a-this->PasH ; // a garde sa dernière valeur pour
les lignes paires
            }
            else
            {
                a = this->AbsDep; // a reprend la valeur du x de
départ pour les lignes impaires
            }
        }

i=1;
do
{
    dest << a << ", " << b << ", " << position++ << endl;
    if (this->TrajetSimple)
    {
        a = a + this->PasH; // si c'est un trajet toujours de gauche
à droite, a augmente toujours
    }
    else
    {
        if (is_even(ligne))
        {
            a = a - this->PasH; // si c'est un trajet variable, a
baisse pour les lignes paires
        }
        else
        {
            a = a + this->PasH; // et augmente pour les lignes
impaires
        }
    }
    i++;
} // ferme la boucle do while
while(i<=this->NbPtsH );

b = b - this->PasV;
ligne++;

} // ferme la boucle for

} // ferme la fonction

```

## FichierSEQ.cpp

```
////////////////////////////////////
// FichierSEQ.cpp
// Implementation of the Class FichierSEQ
// Created on:      18-août-2004 13:40:48
////////////////////////////////////
#include <iostream>
#include <fstream>
#include <string>
#include "stdafx.h"
#include "FichierSEQ.h"

using namespace std;

FichierSEQ::FichierSEQ(){

}

FichierSEQ::~FichierSEQ(){

}

void FichierSEQ::SetSamplePosition(int num_position){
this->SamplePosition = num_position;
}

void FichierSEQ::SetDataFile(string nom_fichier){
this->DataFile = nom_fichier;
}

void FichierSEQ::SetInstrumentSettingFile(string nom_fichier){
this->InstrumentSettingFile = nom_fichier;
}

void FichierSEQ::SetPreMacro(string nom_macro){
this->PreMacro = nom_macro;
}

void FichierSEQ::SetPostMacro(string nom_macro){
this->PostMacro = nom_macro;
}

void FichierSEQ::SetSampleDescription(string description){
this->SampleDescription = description;
}
}
```

```
void FichierSEQ::SetComments(string commentaires){  
this->Comments = commentaires;  
}
```

```
void FichierSEQ::SetRun(int case_cochee){  
this->Run = case_cochee;  
}
```

```
int FichierSEQ::GetSamplePosition(){  
return this->SamplePosition;  
}
```

```
string FichierSEQ::GetDataFile(){  
return this->DataFile;  
}
```

```
string FichierSEQ::GetInstrumentSettingFile(){  
return this->InstrumentSettingFile;  
}
```

```
string FichierSEQ::GetPreMacro(){  
return this->PreMacro;  
}
```

```
string FichierSEQ::GetPostMacro(){  
return this->PostMacro;  
}
```

```
string FichierSEQ::GetSampleDescription(){  
return this->SampleDescription;  
}
```

```
string FichierSEQ::GetComments(){  
return this->Comments;  
}
```

```
int FichierSEQ::GetRun(){  
return this->Run;  
}
```

```

void FichierSEQ::SetLieuSEQ(string enregistrement){
this->LieuSEQ = enregistrement;
}

string FichierSEQ::GetLieuSEQ(){
return this->LieuSEQ;
}

void FichierSEQ::Sauvegarder(string nom_fichier){

int n = this->DataFile.rfind(".");
this->DataFilebis = this->DataFile;

ofstream dest (nom_fichier.c_str());
dest << "AutoSequencer\\Number of Rows=" << this->SamplePosition << endl;

dest << "AutoSequencer\\Author=" << endl;

dest << "AutoSequencer\\Comments=" << endl;

dest << "AutoSequencer\\Data File Root Directory=N/A" << endl;

dest << "AutoSequencer\\Run Log File=" << (this->DataFilebis.substr(0,n)) <<
endl;

int i = 1;

while (i <= this->SamplePosition) {

dest << "AutoSequencer\\Row #" << i << "\\Sample Position=" << i << endl;

dest << "AutoSequencer\\Row #" << i << "\\Data File=" << this-
>DataFile.c_str() << endl;

dest << "AutoSequencer\\Row #" << i << "\\Instrument Settings File=" << this-
>InstrumentSettingFile.c_str() << endl;

dest << "AutoSequencer\\Row #" << i << "\\Pre-Processing Macro=" << this-
>PreMacro.c_str() << endl;

dest << "AutoSequencer\\Row #" << i << "\\Unknown-Standard=" << endl;

dest << "AutoSequencer\\Row #" << i << "\\Data Explorer Set File=" << endl;

dest << "AutoSequencer\\Row #" << i << "\\Calibration File=" << endl;

dest << "AutoSequencer\\Row #" << i << "\\Post-Processing Macro=" << this-
>PostMacro.c_str() << endl;
}
}

```

```

dest << "AutoSequencer\\Row #" << i << "\\Sample Description=" << this-
>SampleDescription.c_str() << endl;

dest << "AutoSequencer\\Row #" << i << "\\Comments=" << this-
>Comments.c_str() << endl;

dest << "AutoSequencer\\Row #" << i << "\\Run State=" << this->Run << endl;

i++;

}

}

```

## Masse.cpp

```

////////////////////////////////////
// Masse.cpp
// Implementation of the Class Masse
// Created on:      18-août-2004 13:40:49
////////////////////////////////////
#include "stdafx.h"
#include "Masse.h"

Masse::Masse(){
}

Masse::~Masse(){
}

void Masse::SetValMasse(double valeur_masse_etudiee){
this->ValMasse = valeur_masse_etudiee;
}

void Masse::SetPics(PointImage courbe){
this->Pics = courbe;
}

void Masse::SetAire(int aire_ou_non){
this->Aire = aire_ou_non;
}

void Masse::SetIntensiteMax(int intensitemax_ou_non){
this->IntensiteMax = intensitemax_ou_non;
}

```

```
void Masse::SetMargefenetre(Coordonnees fenetre){
this->MargeFenetre = fenetre;
}

void Masse::SetMargePourcent(double pourcentage_erreur){
this->MargePourcent = pourcentage_erreur;
}

double Masse::GetValMasse(){
return this->ValMasse;
}

PointImage Masse::GetPics(){
return this->Pics;
}

int Masse::GetAire(){
return this->Aire;
}

int Masse::GetIntensiteMax(){
return this->IntensiteMax;
}

Coordonnees Masse::GetMargeFenetre(){
return this->MargeFenetre;
}

double Masse::GetMargePourcent(){
return this->MargePourcent;
}

void Masse::Dessiner(Masse image){
}

void Masse::ModifParam(Masse dessin){
}
```

## MITICS.cpp

```
// MITICS.cpp : main project file.

#include "stdafx.h"
#include "Form1.h"

using namespace MITICS;

[STAThreadAttribute]
int main(array<System::String ^> ^args)
{
    // Enabling Windows XP visual effects before any controls are created
    Application::EnableVisualStyles();
    Application::SetCompatibleTextRenderingDefault(false);

    // Create the main window and run it
    Application::Run(gcnew Form1());
    return 0;
}
```

## PointImage.cpp

```
////////////////////////////////////
// PointImage.cpp
// Implementation of the Class PointImage
// Created on:      18-août-2004 13:40:50
////////////////////////////////////
#include "stdafx.h"
#include "PointImage.h"

PointImage::PointImage(){
}

PointImage::~PointImage(){
}

void PointImage::SetPosition(Coordonnees position){
this->Position = position;
}

void PointImage::SetNumero(int numero){
this->Numero = numero;
}

void PointImage::SetValAire(double aire){
this->ValAire = aire;
}
```

```
}
```

```
void PointImage::SetValIntensite(double intensite){  
this->ValIntensite;  
}
```

```
void PointImage::SetCouleur(double couleur){  
this->Couleur = couleur;  
}
```

```
Coordonnees PointImage::GetPosition(){  
return this->Position;  
}
```

```
int PointImage::GetNumero(){  
return this->Numero;  
}
```

```
double PointImage::GetValAire(){  
return this->ValAire;  
}
```

```
double PointImage::GetValIntensite(){  
return this->ValIntensite;  
}
```

```
double PointImage::GetCouleur(){  
return this->Couleur;  
}
```

## **Spectre.cpp**

```
////////////////////////////////////  
// Spectre.cpp  
// Implementation of the Class Spectre  
// Created on: 18-août-2004 13:40:50  
////////////////////////////////////  
#include "stdafx.h"  
#include "Spectre.h"  
  
Spectre::Spectre(){  
  
}
```

```

Spectre::~Spectre(){
}

void Spectre::SetCoordPlaque(Coordonnees emplacement_plaque){
this->CoordPlaque = emplacement_plaque;
}

void Spectre::SetCoordSpectre(Coordonnees point){
this->CoordSpectre = point;
}

void Spectre::SetRef(int reference_ou_non){
this->Ref = reference_ou_non;
}

Coordonnees Spectre::GetCoordPlaque(){
return this->CoordPlaque;
}

Coordonnees Spectre::GetCoordSpectre(){
return this->CoordSpectre;
}

int Spectre::GetRef(){
return this->Ref;
}

void Spectre::TracerSpectre(Spectre graphe){
}

//Spectre Spectre::Reference(EnsSpectre spectres_plaque){
//}

```

## **Stdafx.cpp**

```

// stdafx.cpp : source file that includes just the standard includes
// MITICS.pch will be the pre-compiled header
// stdafx.obj will contain the pre-compiled type information

#include "stdafx.h"

```

## Coordonnees.h

```
class Coordonnees
{
public:
    Coordonnees();
    virtual ~Coordonnees();

public:
    void SetX(double abscisse);
    void SetY(double ordonnee);
    double GetX();
    double GetY();

private:
    double X;
    double Y;
};
```

## EnsSpectre.h

```
////////////////////////////////////
// EnsSpectre.h
// Implementation of the Class EnsSpectre
// Created on: 18-août-2004 13:40:47
////////////////////////////////////

#include "Masse.h"

class EnsSpectre
{
public:
    EnsSpectre();
    virtual ~EnsSpectre();

public:
    Spectre *m_Spectre;
    void SetGraphe(Spectre graphe);
    void SetIdPlaque(int identifiant);
    Spectre GetGraphe();
    int GetIdPlaque();
    Masse RechercherAire(EnsSpectre ensemble_spectres, double masse);
    Masse RechercherIntensite(EnsSpectre ensemble_spectres, double masse);

private:
    Spectre Graphe;
    int IdPlaque;
};
```

## FichierBIC.h

```
#include <iostream>
#include <fstream>
#include <string>
using namespace std ;

class FichierBIC
{
public:
    FichierBIC();
    virtual ~FichierBIC();

public:
    void SetLaserIntensity(int intensite_laser);
    void SetAcceleratingVoltage(int acceleration);
    void SetGridVoltage(double voltage);
    void SetGuideWireVoltage0(double voltage0);
    void SetDelayTime(int temps);
    void SetTotalLaserShots(int nb_tirs);
    void SetAcqMassMin(int masse_min);
    void SetAcqMassMax(int masse_max);
    void SetLowMassGate(int masse_la_plus_basse);
    void SetLowMassGateMass(double masse_mini);
    void SetLieuBIC(string enregistrement);
    int GetLaserIntensity();
    int GetAcceleratingVoltage();
    double GetGridVoltage();
    double GetGuideWireVoltage0();
    int GetDelayTime();
    int GetTotalLaserShots();
    int GetAcqMassMin();
    int GetAcqMassMax();
    int GetLowMassGate();
    double GetLowMassGateMass();
    string GetLieuBIC();
    void Sauvegarder(string NomFichier);

private:
    int LaserIntensity;
    int AcceleratingVoltage;
    double GridVoltage;
    double GuideWireVoltage0;
    int DelayTime;
    int TotalLaserShots;
    int AcqMassMin;
    int AcqMassMax;
    int LowMassGate;
    double LowMassGateMass;
    string LieuBIC;
};
```

## FichierPLT.h

```
////////////////////////////////////
// FichierPLT.h
// Implementation of the Class FichierPLT
// Created on: 18-août-2004 13:40:48
////////////////////////////////////

#include <iostream>
#include <fstream>
#include <string>
#include "EnsSpectre.h"

using namespace std ;

class FichierPLT
{
public:
    FichierPLT();
    virtual ~FichierPLT();

public:
    PointImage *m_PointImage;
    void SetNumeros(PointImage *points);
    void SetDep(Coordonnees *point_depart);
    void SetAbsDep(double abscisse);
    void SetOrdDep(double ordonnee);
    void SetArr(Coordonnees *point_arrivee);
    void SetAbsArr(double abscisse);
    void SetOrdArr(double ordonnee);
    void SetPasH(double pas);
    void SetPasV(double pas);
    void SetNbPtsH(int nb_points_horiz);
    void SetNbPtsV(int nb_points_vert);
    void SetTrajetSimple(int trajet_simple);
    void SetLieuPLT(string nom_fichier);
    PointImage GetNumeros();
    //Coordonnees GetDep();
    double GetAbsDep();
    double GetOrdDep();
    //Coordonnees GetArr();
    double GetAbsArr();
    double GetOrdArr();
    double GetPasH();
    double GetPasV();
    int GetNbPtsH();
    int GetNbPtsV();
    string GetLieuPLT();
    double CalculNbPtsLigne(Coordonnees *point_arrivee, Coordonnees
*point_depart, double pas_horiz);
    double CalculNbPtsColonne(Coordonnees *point_arrivee, Coordonnees
*point_depart, double pas_vert);
    double CalculPasLigne(Coordonnees *point_arrivee, Coordonnees
*point_depart, int nb_points_ligne);
};
```

```

    double CalculPasColonne(Coordonnees *point_arrivee, Coordonnees
*point_depart, int nb_points_colonne);
    void Sauvegarder(string nom_fichier);

private:
    PointImage *Numeros;
    Coordonnees *Dep;
    double AbsDep;
    double OrdDep;
    Coordonnees *Arr;
    double AbsArr;
    double OrdArr;
    double PasH;
    double PasV;
    int NbPts;
    int NbPtsH;
    int NbPtsV;
    int TrajetSimple;
    string LieuPLT;
};

```

## **FichierSEQ.h**

```

////////////////////////////////////
// FichierSEQ.h
// Implementation of the Class FichierSEQ
// Created on:      18-août-2004 13:40:48
////////////////////////////////////

#include <iostream>
#include <fstream>
#include <string>

using namespace std ;

class FichierSEQ
{
public:
    FichierSEQ();
    virtual ~FichierSEQ();

public:
    //Ligne *m_Ligne;
    void SetSamplePosition(int num_position);
    void SetDataFile(string nom_fichier);
    void SetInstrumentSettingFile(string nom_fichier);
    void SetPreMacro(string nom_macro);
    void SetPostMacro(string nom_macro);
    void SetSampleDescription(string description);
    void SetComments(string commentaires);
    void SetRun(int case_cochee);
    int GetSamplePosition();
    string GetDataFile();

```

```

    string GetInstrumentSettingFile();
    string GetPreMacro();
    string GetPostMacro();
    string GetSampleDescription();
    string GetComments();
    int GetRun();

    void SetLieuSEQ(string enregistrement);
    string GetLieuSEQ();
    void Sauvegarder(string nom_fichier);

private:
    //Ligne Parametres;
    int SamplePosition;
    string DataFile;
    string DataFilebis;
    string InstrumentSettingFile;
    string PreMacro;
    string PostMacro;
    string SampleDescription;
    string Comments;
    int Run;
    string LieuSEQ;
};

```

## Masse.h

```

////////////////////////////////////
// Masse.h
// Implementation of the Class Masse
// Created on:      18-août-2004 13:40:49
////////////////////////////////////

#include <string>
#include "PointImage.h"

class Masse
{
public:
    Masse();
    virtual ~Masse();

public:
    PointImage *m_PointImage;
    void SetValMasse(double valeur_masse_etudiee);
    void SetPics(PointImage courbe);
    void SetAire(int aire_ou_non);
    void SetIntensiteMax(int intensitemax_ou_non);
    void SetMargefenetre(Coordonnees fenetre);
    void SetMargePourcent(double pourcentage_erreur);
    double GetValMasse();
    PointImage GetPics();
    int GetAire();
};

```

```

    int GetIntensiteMax();
    Coordonnees GetMargeFenetre();
    double GetMargePourcent();
    void Dessiner(Masse image);
    void ModifParam(Masse dessin);

private:
    double ValMasse;
    PointImage Pics;
    int Aire;
    int IntensiteMax;
    Coordonnees MargeFenetre;
    double MargePourcent;
};

```

## **PointImage.h**

```

////////////////////////////////////
// PointImage.h
// Implementation of the Class PointImage
// Created on: 18-août-2004 13:40:49
////////////////////////////////////

#include <string>
#include "Spectre.h"

class PointImage
{

public:
    PointImage();
    virtual ~PointImage();

public:
    Coordonnees *m_Coordonnees;
    //FichierPLT *m_FichierPLT;
    void SetPosition(Coordonnees position);
    void SetNumero(int numero);
    void SetValAire(double aire);
    void SetValIntensite(double intensite);
    void SetCouleur(double couleur);
    Coordonnees GetPosition();
    int GetNumero();
    double GetValAire();
    double GetValIntensite();
    double GetCouleur();

private:
    Coordonnees Position;
    int Numero;
    double ValAire;
    double ValIntensite;
    double Couleur;
};

```

## Spectre.h

```
////////////////////////////////////
// Spectre.h
// Implementation of the Class Spectre
// Created on:      18-août-2004 13:40:50
////////////////////////////////////

#include <string>
#include "Coordonnees.h"

class Spectre
{
public:
    Spectre();
    virtual ~Spectre();

public:
    Coordonnees *_m_Coordonnees;
    void SetCoordPlaque(Coordonnees emplacement_plaque);
    void SetCoordSpectre(Coordonnees point);
    void SetRef(int reference_ou_non);
    Coordonnees GetCoordPlaque();
    Coordonnees GetCoordSpectre();
    int GetRef();
    void TracerSpectre(Spectre graphe);
    //Spectre Reference(EnsSpectre spectres_plaque);

private:
    Coordonnees CoordPlaque;
    Coordonnees CoordSpectre;
    int Ref;

};
```

## Stdafx.h

```
// stdafx.h : include file for standard system include files,
// or project specific include files that are used frequently, but
// are changed infrequently
#pragma once

// TODO: reference additional headers your program requires here
```

## Form1.h

```
#pragma once

using namespace System;
using namespace System::ComponentModel;
using namespace System::Collections;
using namespace System::Windows::Forms;
using namespace System::Data;
using namespace System::Drawing;
```

```

#include "FichierBIC.h"
#include "FichierPLT.h"
#include "FichierSEQ.h"

namespace MITICS {

    /// <summary>
    /// Description résumée de Form1
    ///
    /// AVERTISSEMENT : si vous modifiez le nom de cette classe, vous devrez modifier la
    /// propriété 'Nom du fichier de ressources' de l'outil de compilation de
ressource managée
    /// pour tous les fichiers .resx dont dépend cette classe. Dans le cas contraire,
    /// les concepteurs ne pourront pas interagir correctement avec les ressources
    /// localisées associées à ce formulaire.
    /// </summary>
    public ref class Form1 : public System::Windows::Forms::Form
    {
    public:
        Form1(void)
        {
            InitializeComponent();
            //
            //TODO : ajoutez ici le code du constructeur
            //
        }

    protected:
        /// <summary>
        /// Nettoyage des ressources utilisées.
        /// </summary>
        ~Form1()
        {
            if (components)
            {
                delete components;
            }
        }
    private: System::Windows::Forms::TabControl^ pcLogiciel;
    protected:

    private: System::Windows::Forms::TabPage^ tsWelcome;
    private: System::Windows::Forms::TabPage^ tsBIC;
    private: System::Windows::Forms::TabPage^ tsPLT;
    private: System::Windows::Forms::TabPage^ tsPLTbis;
    private: System::Windows::Forms::TabPage^ tsSEQ;
    private: System::Windows::Forms::TabPage^ tsConvert;
    private: System::Windows::Forms::TabPage^ tsImageReconstruction;
    protected:

    private: System::Windows::Forms::PictureBox^ pictureBox1;
    private: System::Windows::Forms::PictureBox^ pictureBox2;
    private: System::Windows::Forms::Label^ lFilesCreated;
    private: System::Windows::Forms::Label^ lRed;
    private: System::Windows::Forms::Label^ lGreen;

    private: System::Windows::Forms::GroupBox^ groupBox1;
    private: System::Windows::Forms::Label^ lISF;
    private: System::Windows::Forms::CheckBox^ cbFichierSEQ;
    private: System::Windows::Forms::CheckBox^ cbFichierPLT;
    private: System::Windows::Forms::CheckBox^ cbFichierBIC;

    private: System::Windows::Forms::Label^ lAF;

    private: System::Windows::Forms::Label^ lPCF;
    private: System::Windows::Forms::Button^ cClearAll;
    private: System::Windows::Forms::Button^ bClose;

```

```

private: System::Windows::Forms::PictureBox^ pictureBox4;
private: System::Windows::Forms::PictureBox^ pictureBox3;
private: System::Windows::Forms::ProgressBar^ pbLogiciel;

private: System::Windows::Forms::Label^ label7;
private: System::Windows::Forms::TrackBar^ tbLaserIntensity;

private: System::Windows::Forms::Label^ label1;
private: System::Windows::Forms::TextBox^ edLowMassGateMass;

private: System::Windows::Forms::TextBox^ edAcqMassMin;
private: System::Windows::Forms::TextBox^ edAcqMassMax;

private: System::Windows::Forms::TextBox^ edTotalLaserShots;

private: System::Windows::Forms::TextBox^ edDelayTime;

private: System::Windows::Forms::TextBox^ edGuideWireVoltage0;

private: System::Windows::Forms::TextBox^ edGridVoltage;

private: System::Windows::Forms::TextBox^ edAcceleratingVoltage;

private: System::Windows::Forms::TextBox^ edLaserIntensity;
private: System::Windows::Forms::CheckBox^ cbLowMassGate;

private: System::Windows::Forms::Label^ label9;
private: System::Windows::Forms::Label^ label8;
private: System::Windows::Forms::Label^ label6;
private: System::Windows::Forms::Label^ label5;
private: System::Windows::Forms::Label^ label4;
private: System::Windows::Forms::Label^ label3;
private: System::Windows::Forms::Label^ label2;
private: System::Windows::Forms::ErrorProvider^ errorProvider1;
private: System::Windows::Forms::Button^ bDefaultValues;

private: System::Windows::Forms::Button^ bClearBIC;
private: System::Windows::Forms::Button^ bSaveFileBIC;

private: System::Windows::Forms::Label^ label10;
private: System::Windows::Forms::Label^ label20;
private: System::Windows::Forms::Label^ label19;
private: System::Windows::Forms::Label^ label18;
private: System::Windows::Forms::Label^ label17;
private: System::Windows::Forms::Label^ label16;
private: System::Windows::Forms::Label^ label15;
private: System::Windows::Forms::Label^ label14;
private: System::Windows::Forms::Label^ label13;
private: System::Windows::Forms::Label^ label12;
private: System::Windows::Forms::Label^ label11;
private: System::Windows::Forms::ColorDialog^ cdImage;

private: System::Windows::Forms::GroupBox^ groupBox4;
private: System::Windows::Forms::GroupBox^ groupBox5;
private: System::Windows::Forms::GroupBox^ groupBox3;
private: System::Windows::Forms::GroupBox^ Coordinates;
private: System::Windows::Forms::TextBox^ eXArr;

private: System::Windows::Forms::TextBox^ eXDep;
private: System::Windows::Forms::TextBox^ eYArr;

private: System::Windows::Forms::TextBox^ eYDep;
private: System::Windows::Forms::Button^ bStep;
private: System::Windows::Forms::Label^ lPasV;
private: System::Windows::Forms::Label^ lPasH;

```

```

private: System::Windows::Forms::TextBox^ ePasV;

private: System::Windows::Forms::TextBox^ ePasH;
private: System::Windows::Forms::Label^ lYArr;

private: System::Windows::Forms::Label^ lXArr;
private: System::Windows::Forms::Label^ lYDep;

private: System::Windows::Forms::Label^ lXDep;

private: System::Windows::Forms::Label^ lcoordarr;

private: System::Windows::Forms::Label^ lcoorddep;
private: System::Windows::Forms::Button^ bClearPLT;

private: System::Windows::Forms::Button^ bCreerPLT;

private: System::Windows::Forms::Button^ bNumberPoints;
private: System::Windows::Forms::Label^ lNoValuePointsRow;

private: System::Windows::Forms::Label^ lNbPtsV;

private: System::Windows::Forms::Label^ lNbPtsH;

private: System::Windows::Forms::TextBox^ eNbPtsV;

private: System::Windows::Forms::TextBox^ eNbPtsH;
private: System::Windows::Forms::RadioButton^ rbComplexTrajectory;

private: System::Windows::Forms::RadioButton^ rbSimpleTrajectory;
private: System::Windows::Forms::Label^ lNoValueVerticalStep;
private: System::Windows::Forms::TextBox^ eNbPtsHbis;

private: System::Windows::Forms::TextBox^ ePasVbis;

private: System::Windows::Forms::TextBox^ ePsHbis;
private: System::Windows::Forms::TextBox^ eNbPtsVbis;
private: System::Windows::Forms::PictureBox^ iPhoto;

private: System::Windows::Forms::GroupBox^ groupBox2;
private: System::Windows::Forms::Button^ bPhoto;
private: System::Windows::Forms::TextBox^ ePhoto;

private: System::Windows::Forms::Label^ label25;
private: System::Windows::Forms::Label^ label24;
private: System::Windows::Forms::Label^ label23;
private: System::Windows::Forms::Label^ label22;
private: System::Windows::Forms::Label^ label21;
private: System::Windows::Forms::OpenFileDialog^ opSequenceType;

private: System::Windows::Forms::GroupBox^ bFurther;

private: System::Windows::Forms::GroupBox^ bMacros;

private: System::Windows::Forms::GroupBox^ bFiles;
private: System::Windows::Forms::TextBox^ eDataFile;
private: System::Windows::Forms::TextBox^ eInstrumentSettingFile;
private: System::Windows::Forms::TextBox^ eSequenceType;

private: System::Windows::Forms::TextBox^ eNbLignes;

private: System::Windows::Forms::Label^ lNumberLines;
private: System::Windows::Forms::Label^ lInstrumentSettingFile;

```

```

private: System::Windows::Forms::Label^ lDataFile;

private: System::Windows::Forms::Label^ lSequenceType;
private: System::Windows::Forms::Button^ bParcourirISF;

private: System::Windows::Forms::Button^ bParcourirDF;

private: System::Windows::Forms::Button^ bParcourirST;
private: System::Windows::Forms::Button^ bCreerFichierSEQ;
private: System::Windows::Forms::Button^ bClearSEQ;

private: System::Windows::Forms::Label^ lComments;

private: System::Windows::Forms::Label^ lSampleDescription;
private: System::Windows::Forms::TextBox^ eComments;

private: System::Windows::Forms::TextBox^ eSampleDescription;

private: System::Windows::Forms::Label^ lPostMacro;

private: System::Windows::Forms::Label^ lPreMacro;
private: System::Windows::Forms::ComboBox^ cbPostMacro;

private: System::Windows::Forms::ComboBox^ cbPreMacro;
private: System::Windows::Forms::Button^ bConvert;

private: System::Windows::Forms::Button^ bGlobalFile;

private: System::Windows::Forms::Button^ bFolderOfTheFiles;

private: System::Windows::Forms::Button^ bPointsCoordinatesFile;
private: System::Windows::Forms::TextBox^ eGlobalFile;

private: System::Windows::Forms::TextBox^ eFolderOfTheFiles;

private: System::Windows::Forms::TextBox^ ePointsCoordinatesFile;
private: System::Windows::Forms::Label^ lGlobalFile;

private: System::Windows::Forms::Label^ lFolderOfTheFiles;

private: System::Windows::Forms::Label^ lPointsCoordinatesFile;
private: System::Windows::Forms::GroupBox^ Bevell;
private: System::Windows::Forms::RadioButton^ rbIntensiteMax;

private: System::Windows::Forms::RadioButton^ rbAire;

private: System::Windows::Forms::TextBox^ eMasseSurCharge;

private: System::Windows::Forms::Label^ lMasseSurCharge;
private: System::Windows::Forms::Button^ bColors;
private: System::Windows::Forms::SaveFileDialog^ sdGlobalFile;

private: System::Windows::Forms::OpenFileDialog^ opDataFile;
private: System::Windows::Forms::OpenFileDialog^ opInstrumentSettingFile;
private: System::Windows::Forms::OpenFileDialog^ opPhoto;
private: System::Windows::Forms::OpenFileDialog^ odPointsCoordinatesFile;

private: System::Windows::Forms::OpenFileDialog^ odFolderOfTheFiles;
private: System::Windows::Forms::SaveFileDialog^ sdBIC;
private: System::Windows::Forms::SaveFileDialog^ sdPLT;
private: System::Windows::Forms::SaveFileDialog^ sdSEQ;

private: System::ComponentModel::IContainer^ components;

private:
    /// <summary>
    /// Variable nécessaire au concepteur.

```

```

    /// </summary>

public:
    FichierBIC *NomFichier;
    FichierPLT *NomFichierPLT;
    Coordonnees *point_dep;
    Coordonnees *point_arr;
    FichierSEQ *NomFichierSEQ;
    bool Dragging;
    double rapportH;
    double rapportV;
    bool bHaveMouse;
    bool bHaveMouseCircle1;
    System::Drawing::Point ptOriginal;
private: System::Windows::Forms::PictureBox^ pictureBoxCircle1;
public:

public:

public:
    System::Drawing::Point ptLast;

#pragma region Windows Form Designer generated code
    /// <summary>
    /// Méthode requise pour la prise en charge du concepteur - ne modifiez pas
    /// le contenu de cette méthode avec l'éditeur de code.
    /// </summary>
    void InitializeComponent(void)
    {
        this->components = (gcnew System::ComponentModel::Container());
        System::ComponentModel::ComponentResourceManager^ resources = (gcnew
System::ComponentModel::ComponentResourceManager(Form1::typeid));
        this->pcLogiciel = (gcnew System::Windows::Forms::TabControl());
        this->tsWelcome = (gcnew System::Windows::Forms::TabPage());
        this->label7 = (gcnew System::Windows::Forms::Label());
        this->pictureBox4 = (gcnew System::Windows::Forms::PictureBox());
        this->pictureBox3 = (gcnew System::Windows::Forms::PictureBox());
        this->tsBIC = (gcnew System::Windows::Forms::TabPage());
        this->label20 = (gcnew System::Windows::Forms::Label());
        this->label19 = (gcnew System::Windows::Forms::Label());
        this->label18 = (gcnew System::Windows::Forms::Label());
        this->label17 = (gcnew System::Windows::Forms::Label());
        this->label16 = (gcnew System::Windows::Forms::Label());
        this->label15 = (gcnew System::Windows::Forms::Label());
        this->label14 = (gcnew System::Windows::Forms::Label());
        this->label13 = (gcnew System::Windows::Forms::Label());
        this->label12 = (gcnew System::Windows::Forms::Label());
        this->label11 = (gcnew System::Windows::Forms::Label());
        this->label10 = (gcnew System::Windows::Forms::Label());
        this->bDefaultValues = (gcnew System::Windows::Forms::Button());
        this->bClearBIC = (gcnew System::Windows::Forms::Button());
        this->bSaveFileBIC = (gcnew System::Windows::Forms::Button());
        this->cbLowMassGate = (gcnew System::Windows::Forms::CheckBox());
        this->label9 = (gcnew System::Windows::Forms::Label());
        this->label8 = (gcnew System::Windows::Forms::Label());
        this->label6 = (gcnew System::Windows::Forms::Label());
        this->label5 = (gcnew System::Windows::Forms::Label());
        this->label4 = (gcnew System::Windows::Forms::Label());
        this->label3 = (gcnew System::Windows::Forms::Label());
        this->label2 = (gcnew System::Windows::Forms::Label());
        this->tbLaserIntensity = (gcnew System::Windows::Forms::TrackBar());
        this->label1 = (gcnew System::Windows::Forms::Label());
        this->edLowMassGateMass = (gcnew System::Windows::Forms::TextBox());
        this->edAcqMassMin = (gcnew System::Windows::Forms::TextBox());
        this->edAcqMassMax = (gcnew System::Windows::Forms::TextBox());
        this->edTotalLaserShots = (gcnew System::Windows::Forms::TextBox());
        this->edDelayTime = (gcnew System::Windows::Forms::TextBox());
        this->edGuideWireVoltage0 = (gcnew System::Windows::Forms::TextBox());
        this->edGridVoltage = (gcnew System::Windows::Forms::TextBox());
        this->edAcceleratingVoltage = (gcnew System::Windows::Forms::TextBox());
        this->edLaserIntensity = (gcnew System::Windows::Forms::TextBox());
    }

```

```

this->tsPLT = (gcnew System::Windows::Forms::TabPage());
this->bClearPLT = (gcnew System::Windows::Forms::Button());
this->bCreerPLT = (gcnew System::Windows::Forms::Button());
this->groupBox4 = (gcnew System::Windows::Forms::GroupBox());
this->bNumberPoints = (gcnew System::Windows::Forms::Button());
this->lNoValuePointsRow = (gcnew System::Windows::Forms::Label());
this->lNbPtsV = (gcnew System::Windows::Forms::Label());
this->lNbPtsH = (gcnew System::Windows::Forms::Label());
this->eNbPtsV = (gcnew System::Windows::Forms::TextBox());
this->eNbPtsH = (gcnew System::Windows::Forms::TextBox());
this->groupBox5 = (gcnew System::Windows::Forms::GroupBox());
this->rbComplexTrajectory = (gcnew System::Windows::Forms::RadioButton());
this->rbSimpleTrajectory = (gcnew System::Windows::Forms::RadioButton());
this->groupBox3 = (gcnew System::Windows::Forms::GroupBox());
this->lNoValueVerticalStep = (gcnew System::Windows::Forms::Label());
this->bStep = (gcnew System::Windows::Forms::Button());
this->lPasV = (gcnew System::Windows::Forms::Label());
this->lPasH = (gcnew System::Windows::Forms::Label());
this->ePasV = (gcnew System::Windows::Forms::TextBox());
this->ePasH = (gcnew System::Windows::Forms::TextBox());
this->Coordinates = (gcnew System::Windows::Forms::GroupBox());
this->lYArr = (gcnew System::Windows::Forms::Label());
this->lXArr = (gcnew System::Windows::Forms::Label());
this->lYDep = (gcnew System::Windows::Forms::Label());
this->lXDep = (gcnew System::Windows::Forms::Label());
this->lcoordarr = (gcnew System::Windows::Forms::Label());
this->lcoorddep = (gcnew System::Windows::Forms::Label());
this->eYArr = (gcnew System::Windows::Forms::TextBox());
this->eYDep = (gcnew System::Windows::Forms::TextBox());
this->eXArr = (gcnew System::Windows::Forms::TextBox());
this->eXDep = (gcnew System::Windows::Forms::TextBox());
this->tsPLTbis = (gcnew System::Windows::Forms::TabPage());
this->pictureBoxCircle1 = (gcnew System::Windows::Forms::PictureBox());
this->label25 = (gcnew System::Windows::Forms::Label());
this->label24 = (gcnew System::Windows::Forms::Label());
this->label23 = (gcnew System::Windows::Forms::Label());
this->label22 = (gcnew System::Windows::Forms::Label());
this->label21 = (gcnew System::Windows::Forms::Label());
this->eNbPtsHbis = (gcnew System::Windows::Forms::TextBox());
this->ePasVbis = (gcnew System::Windows::Forms::TextBox());
this->ePsHbis = (gcnew System::Windows::Forms::TextBox());
this->eNbPtsVbis = (gcnew System::Windows::Forms::TextBox());
this->iPhoto = (gcnew System::Windows::Forms::PictureBox());
this->groupBox2 = (gcnew System::Windows::Forms::GroupBox());
this->bPhoto = (gcnew System::Windows::Forms::Button());
this->ePhoto = (gcnew System::Windows::Forms::TextBox());
this->tsSEQ = (gcnew System::Windows::Forms::TabPage());
this->bCreerFichierSEQ = (gcnew System::Windows::Forms::Button());
this->bClearSEQ = (gcnew System::Windows::Forms::Button());
this->bFurther = (gcnew System::Windows::Forms::GroupBox());
this->lComments = (gcnew System::Windows::Forms::Label());
this->lSampleDescription = (gcnew System::Windows::Forms::Label());
this->eComments = (gcnew System::Windows::Forms::TextBox());
this->eSampleDescription = (gcnew System::Windows::Forms::TextBox());
this->bMacros = (gcnew System::Windows::Forms::GroupBox());
this->lPostMacro = (gcnew System::Windows::Forms::Label());
this->lPreMacro = (gcnew System::Windows::Forms::Label());
this->cbPostMacro = (gcnew System::Windows::Forms::ComboBox());
this->cbPreMacro = (gcnew System::Windows::Forms::ComboBox());
this->bFiles = (gcnew System::Windows::Forms::GroupBox());
this->eNbLignes = (gcnew System::Windows::Forms::TextBox());
this->lNumberLines = (gcnew System::Windows::Forms::Label());
this->lInstrumentSettingFile = (gcnew System::Windows::Forms::Label());
this->lDataFile = (gcnew System::Windows::Forms::Label());
this->lSequenceType = (gcnew System::Windows::Forms::Label());
this->bParcourirISF = (gcnew System::Windows::Forms::Button());
this->bParcourirDF = (gcnew System::Windows::Forms::Button());
this->bParcourirST = (gcnew System::Windows::Forms::Button());
this->eDataFile = (gcnew System::Windows::Forms::TextBox());
this->eInstrumentSettingFile = (gcnew System::Windows::Forms::TextBox());
this->eSequenceType = (gcnew System::Windows::Forms::TextBox());

```

```

this->tsConvert = (gcnew System::Windows::Forms::TabPage());
this->bConvert = (gcnew System::Windows::Forms::Button());
this->bGlobalFile = (gcnew System::Windows::Forms::Button());
this->bFolderOfTheFiles = (gcnew System::Windows::Forms::Button());
this->bPointsCoordinatesFile = (gcnew System::Windows::Forms::Button());
this->eGlobalFile = (gcnew System::Windows::Forms::TextBox());
this->eFolderOfTheFiles = (gcnew System::Windows::Forms::TextBox());
this->ePointsCoordinatesFile = (gcnew System::Windows::Forms::TextBox());
this->lGlobalFile = (gcnew System::Windows::Forms::Label());
this->lFoderOfTheFiles = (gcnew System::Windows::Forms::Label());
this->lPointsCoordinatesFile = (gcnew System::Windows::Forms::Label());
this->tsImageReconstruction = (gcnew System::Windows::Forms::TabPage());
this->Bevell = (gcnew System::Windows::Forms::GroupBox());
this->rbIntensiteMax = (gcnew System::Windows::Forms::RadioButton());
this->rbAire = (gcnew System::Windows::Forms::RadioButton());
this->eMasseSurCharge = (gcnew System::Windows::Forms::TextBox());
this->lMasseSurCharge = (gcnew System::Windows::Forms::Label());
this->bColors = (gcnew System::Windows::Forms::Button());
this->pictureBox1 = (gcnew System::Windows::Forms::PictureBox());
this->pictureBox2 = (gcnew System::Windows::Forms::PictureBox());
this->lFilesCreated = (gcnew System::Windows::Forms::Label());
this->lRed = (gcnew System::Windows::Forms::Label());
this->lGreen = (gcnew System::Windows::Forms::Label());
this->groupBox1 = (gcnew System::Windows::Forms::GroupBox());
this->lAF = (gcnew System::Windows::Forms::Label());
this->lPCF = (gcnew System::Windows::Forms::Label());
this->lISF = (gcnew System::Windows::Forms::Label());
this->cbFichierSEQ = (gcnew System::Windows::Forms::CheckBox());
this->cbFichierPLT = (gcnew System::Windows::Forms::CheckBox());
this->cbFichierBIC = (gcnew System::Windows::Forms::CheckBox());
this->cClearAll = (gcnew System::Windows::Forms::Button());
this->bClose = (gcnew System::Windows::Forms::Button());
this->pbLogiciel = (gcnew System::Windows::Forms::ProgressBar());
this->errorProvider1 = (gcnew System::Windows::Forms::ErrorProvider(this-
>components));

this->cdImage = (gcnew System::Windows::Forms::ColorDialog());
this->opSequenceType = (gcnew System::Windows::Forms::OpenFileDialog());
this->sdGlobalFile = (gcnew System::Windows::Forms::SaveFileDialog());
this->opDataFile = (gcnew System::Windows::Forms::OpenFileDialog());
this->opInstrumentSettingFile = (gcnew
System::Windows::Forms::OpenFileDialog());
this->opPhoto = (gcnew System::Windows::Forms::OpenFileDialog());
this->odPointsCoordinatesFile = (gcnew
System::Windows::Forms::OpenFileDialog());
this->odFolderOfTheFiles = (gcnew
System::Windows::Forms::OpenFileDialog());
this->sdBIC = (gcnew System::Windows::Forms::SaveFileDialog());
this->sdPLT = (gcnew System::Windows::Forms::SaveFileDialog());
this->sdSEQ = (gcnew System::Windows::Forms::SaveFileDialog());
this->pcLogiciel->SuspendLayout();
this->tsWelcome->SuspendLayout();
(cli::safe_cast<System::ComponentModel::ISupportInitialize^ >(this-
>pictureBox4))->BeginInit();
(cli::safe_cast<System::ComponentModel::ISupportInitialize^ >(this-
>pictureBox3))->BeginInit();
this->tsBIC->SuspendLayout();
(cli::safe_cast<System::ComponentModel::ISupportInitialize^ >(this-
>tbLaserIntensity))->BeginInit();
this->tsPLT->SuspendLayout();
this->groupBox4->SuspendLayout();
this->groupBox5->SuspendLayout();
this->groupBox3->SuspendLayout();
this->Coordinates->SuspendLayout();
this->tsPLTbis->SuspendLayout();
(cli::safe_cast<System::ComponentModel::ISupportInitialize^ >(this-
>pictureBoxCircle1))->BeginInit();
(cli::safe_cast<System::ComponentModel::ISupportInitialize^ >(this-
>iPhoto))->BeginInit();
this->groupBox2->SuspendLayout();
this->tsSEQ->SuspendLayout();
this->bFurther->SuspendLayout();

```

```

        this->bMacros->SuspendLayout();
        this->bFiles->SuspendLayout();
        this->tsConvert->SuspendLayout();
        this->tsImageReconstruction->SuspendLayout();
        this->Bevell->SuspendLayout();
        (cli::safe_cast<System::ComponentModel::ISupportInitialize^ >(this->
pictureBox1))->BeginInit();
        (cli::safe_cast<System::ComponentModel::ISupportInitialize^ >(this->
pictureBox2))->BeginInit();
        this->groupBox1->SuspendLayout();
        (cli::safe_cast<System::ComponentModel::ISupportInitialize^ >(this->
errorProvider1))->BeginInit();
        this->SuspendLayout();
        //
        // pcLogiciel
        //
        this->pcLogiciel->Controls->Add(this->tsWelcome);
        this->pcLogiciel->Controls->Add(this->tsBIC);
        this->pcLogiciel->Controls->Add(this->tsPLT);
        this->pcLogiciel->Controls->Add(this->tsPLTbis);
        this->pcLogiciel->Controls->Add(this->tsSEQ);
        this->pcLogiciel->Controls->Add(this->tsConvert);
        this->pcLogiciel->Controls->Add(this->tsImageReconstruction);
        this->pcLogiciel->ItemSize = System::Drawing::Size(57, 18);
        this->pcLogiciel->Location = System::Drawing::Point(129, 12);
        this->pcLogiciel->Name = L"pcLogiciel";
        this->pcLogiciel->SelectedIndex = 0;
        this->pcLogiciel->Size = System::Drawing::Size(628, 452);
        this->pcLogiciel->TabIndex = 0;
        //
        // tsWelcome
        //
        this->tsWelcome->BackColor = System::Drawing::Color::LightBlue;
        this->tsWelcome->BackgroundImageLayout =
System::Windows::Forms::ImageLayout::Stretch;
        this->tsWelcome->Controls->Add(this->label7);
        this->tsWelcome->Controls->Add(this->pictureBox4);
        this->tsWelcome->Controls->Add(this->pictureBox3);
        this->tsWelcome->Location = System::Drawing::Point(4, 22);
        this->tsWelcome->Name = L"tsWelcome";
        this->tsWelcome->Padding = System::Windows::Forms::Padding(3);
        this->tsWelcome->Size = System::Drawing::Size(620, 426);
        this->tsWelcome->TabIndex = 0;
        this->tsWelcome->Text = L"Welcome";
        this->tsWelcome->UseVisualStyleBackColor = true;
        //
        // label7
        //
        this->label7->BorderStyle = System::Windows::Forms::BorderStyle::Fixed3D;
        this->label7->Font = (gcnew System::Drawing::Font(L"Verdana", 14.25F,
System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(0)));
        this->label7->Location = System::Drawing::Point(6, 154);
        this->label7->Name = L"label7";
        this->label7->Size = System::Drawing::Size(608, 127);
        this->label7->TabIndex = 2;
        this->label7->Text = L"Welcome in the IMAGING SOFTWARE";
        this->label7->TextAlign = System::Drawing::ContentAlignment::MiddleCenter;
        //
        // pictureBox4
        //
        this->pictureBox4->BackgroundImage =
(cli::safe_cast<System::Drawing::Image^ >(resources->
GetObject(L"pictureBox4.BackgroundImage")));
        this->pictureBox4->BackgroundImageLayout =
System::Windows::Forms::ImageLayout::Stretch;
        this->pictureBox4->Location = System::Drawing::Point(228, 327);
        this->pictureBox4->Name = L"pictureBox4";
        this->pictureBox4->Size = System::Drawing::Size(176, 93);
        this->pictureBox4->TabIndex = 1;
        this->pictureBox4->TabStop = false;

```

```

//
// pictureBox3
//
this->pictureBox3->BackgroundImage =
(cli::safe_cast<System::Drawing::Image^ >(resources-
>GetObject(L"pictureBox3.BackgroundImage")));
this->pictureBox3->BackgroundImageLayout =
System::Windows::Forms::ImageLayout::Center;
this->pictureBox3->Location = System::Drawing::Point(6, 25);
this->pictureBox3->Name = L"pictureBox3";
this->pictureBox3->Size = System::Drawing::Size(608, 58);
this->pictureBox3->TabIndex = 0;
this->pictureBox3->TabStop = false;
//
// tsBIC
//
this->tsBIC->BackColor = System::Drawing::Color::LightBlue;
this->tsBIC->Controls->Add(this->label20);
this->tsBIC->Controls->Add(this->label19);
this->tsBIC->Controls->Add(this->label18);
this->tsBIC->Controls->Add(this->label17);
this->tsBIC->Controls->Add(this->label16);
this->tsBIC->Controls->Add(this->label15);
this->tsBIC->Controls->Add(this->label14);
this->tsBIC->Controls->Add(this->label13);
this->tsBIC->Controls->Add(this->label12);
this->tsBIC->Controls->Add(this->label11);
this->tsBIC->Controls->Add(this->label10);
this->tsBIC->Controls->Add(this->bDefaultValues);
this->tsBIC->Controls->Add(this->bClearBIC);
this->tsBIC->Controls->Add(this->bSaveFileBIC);
this->tsBIC->Controls->Add(this->cbLowMassGate);
this->tsBIC->Controls->Add(this->label9);
this->tsBIC->Controls->Add(this->label8);
this->tsBIC->Controls->Add(this->label6);
this->tsBIC->Controls->Add(this->label5);
this->tsBIC->Controls->Add(this->label4);
this->tsBIC->Controls->Add(this->label3);
this->tsBIC->Controls->Add(this->label2);
this->tsBIC->Controls->Add(this->tbLaserIntensity);
this->tsBIC->Controls->Add(this->label1);
this->tsBIC->Controls->Add(this->edLowMassGateMass);
this->tsBIC->Controls->Add(this->edAcqMassMin);
this->tsBIC->Controls->Add(this->edAcqMassMax);
this->tsBIC->Controls->Add(this->edTotalLaserShots);
this->tsBIC->Controls->Add(this->edDelayTime);
this->tsBIC->Controls->Add(this->edGuideWireVoltage0);
this->tsBIC->Controls->Add(this->edGridVoltage);
this->tsBIC->Controls->Add(this->edAcceleratingVoltage);
this->tsBIC->Controls->Add(this->edLaserIntensity);
this->tsBIC->Location = System::Drawing::Point(4, 22);
this->tsBIC->Name = L"tsBIC";
this->tsBIC->Padding = System::Windows::Forms::Padding(3);
this->tsBIC->Size = System::Drawing::Size(620, 426);
this->tsBIC->TabIndex = 1;
this->tsBIC->Text = L"BIC File";
this->tsBIC->UseVisualStyleBackColor = true;
//
// label20
//
this->label20->AutoSize = true;
this->label20->Location = System::Drawing::Point(315, 231);
this->label20->Name = L"label20";
this->label20->Size = System::Drawing::Size(49, 13);
this->label20->TabIndex = 32;
this->label20->Text = L"< 10 000";
//
// label19
//
this->label19->AutoSize = true;
this->label19->Location = System::Drawing::Point(312, 208);

```

```

this->label19->Name = L"label19";
this->label19->Size = System::Drawing::Size(78, 13);
this->label19->TabIndex = 31;
this->label19->Text = L"0 - 3 000 nsec ";
//
// label18
//
this->label18->AutoSize = true;
this->label18->Location = System::Drawing::Point(315, 182);
this->label18->Name = L"label18";
this->label18->Size = System::Drawing::Size(69, 13);
this->label18->TabIndex = 30;
this->label18->Text = L"50.0 - 99.9 %";
//
// label17
//
this->label17->AutoSize = true;
this->label17->Location = System::Drawing::Point(315, 156);
this->label17->Name = L"label17";
this->label17->Size = System::Drawing::Size(69, 13);
this->label17->TabIndex = 29;
this->label17->Text = L"50.0 - 99.9 %";
//
// label16
//
this->label16->AutoSize = true;
this->label16->Location = System::Drawing::Point(315, 130);
this->label16->Name = L"label16";
this->label16->Size = System::Drawing::Size(65, 13);
this->label16->TabIndex = 28;
this->label16->Text = L"0 - 25 000 V";
//
// label15
//
this->label15->AutoSize = true;
this->label15->Location = System::Drawing::Point(536, 33);
this->label15->Name = L"label15";
this->label15->Size = System::Drawing::Size(31, 13);
this->label15->TabIndex = 27;
this->label15->Text = L"4000";
//
// label14
//
this->label14->AutoSize = true;
this->label14->Location = System::Drawing::Point(583, 33);
this->label14->Name = L"label14";
this->label14->Size = System::Drawing::Size(31, 13);
this->label14->TabIndex = 26;
this->label14->Text = L"4600";
//
// label13
//
this->label13->AutoSize = true;
this->label13->Location = System::Drawing::Point(485, 33);
this->label13->Name = L"label13";
this->label13->Size = System::Drawing::Size(31, 13);
this->label13->TabIndex = 25;
this->label13->Text = L"3000";
//
// label12
//
this->label12->AutoSize = true;
this->label12->Location = System::Drawing::Point(428, 33);
this->label12->Name = L"label12";
this->label12->Size = System::Drawing::Size(31, 13);
this->label12->TabIndex = 24;
this->label12->Text = L"2000";
//
// label11
//
this->label11->AutoSize = true;

```

```

this->label11->Location = System::Drawing::Point(366, 33);
this->label11->Name = L"label11";
this->label11->Size = System::Drawing::Size(31, 13);
this->label11->TabIndex = 23;
this->label11->Text = L"1000";
//
// label10
//
this->label10->AutoSize = true;
this->label10->Location = System::Drawing::Point(315, 33);
this->label10->Name = L"label10";
this->label10->Size = System::Drawing::Size(13, 13);
this->label10->TabIndex = 22;
this->label10->Text = L"0";
//
// bDefaultValues
//
this->bDefaultValues->Location = System::Drawing::Point(499, 379);
this->bDefaultValues->Name = L"bDefaultValues";
this->bDefaultValues->Size = System::Drawing::Size(102, 23);
this->bDefaultValues->TabIndex = 21;
this->bDefaultValues->Text = L"Default Values";
this->bDefaultValues->UseVisualStyleBackColor = true;
this->bDefaultValues->Click += gcnew System::EventHandler(this,
&Form1::bDefaultValues_Click);
//
// bClearBIC
//
this->bClearBIC->Location = System::Drawing::Point(341, 379);
this->bClearBIC->Name = L"bClearBIC";
this->bClearBIC->Size = System::Drawing::Size(75, 23);
this->bClearBIC->TabIndex = 20;
this->bClearBIC->Text = L"Clear BIC";
this->bClearBIC->UseVisualStyleBackColor = true;
this->bClearBIC->Click += gcnew System::EventHandler(this,
&Form1::bClearBIC_Click);
//
// bSaveFileBIC
//
this->bSaveFileBIC->Location = System::Drawing::Point(150, 380);
this->bSaveFileBIC->Name = L"bSaveFileBIC";
this->bSaveFileBIC->Size = System::Drawing::Size(106, 23);
this->bSaveFileBIC->TabIndex = 19;
this->bSaveFileBIC->Text = L"Create the file";
this->bSaveFileBIC->UseVisualStyleBackColor = true;
this->bSaveFileBIC->Click += gcnew System::EventHandler(this,
&Form1::bSaveFileBIC_Click);
//
// cbLowMassGate
//
this->cbLowMassGate->AutoSize = true;
this->cbLowMassGate->Location = System::Drawing::Point(134, 315);
this->cbLowMassGate->Name = L"cbLowMassGate";
this->cbLowMassGate->Size = System::Drawing::Size(100, 17);
this->cbLowMassGate->TabIndex = 18;
this->cbLowMassGate->Text = L"Low Mass Gate";
this->cbLowMassGate->UseVisualStyleBackColor = true;
//
// label9
//
this->label9->AutoSize = true;
this->label9->Location = System::Drawing::Point(315, 260);
this->label9->Name = L"label9";
this->label9->Size = System::Drawing::Size(20, 13);
this->label9->TabIndex = 17;
this->label9->Text = L"To";
//
// label8
//
this->label8->AutoSize = true;
this->label8->Location = System::Drawing::Point(6, 260);

```

```

this->label8->Name = L"label8";
this->label8->Size = System::Drawing::Size(90, 13);
this->label8->TabIndex = 16;
this->label8->Text = L"Mass Range from";
//
// label6
//
this->label6->AutoSize = true;
this->label6->Location = System::Drawing::Point(6, 234);
this->label6->Name = L"label6";
this->label6->Size = System::Drawing::Size(90, 13);
this->label6->TabIndex = 15;
this->label6->Text = L"Total Laser Shots";
//
// label5
//
this->label5->AutoSize = true;
this->label5->Location = System::Drawing::Point(6, 208);
this->label5->Name = L"label5";
this->label5->Size = System::Drawing::Size(60, 13);
this->label5->TabIndex = 14;
this->label5->Text = L"Delay Time";
//
// label4
//
this->label4->AutoSize = true;
this->label4->Location = System::Drawing::Point(6, 182);
this->label4->Name = L"label4";
this->label4->Size = System::Drawing::Size(60, 13);
this->label4->TabIndex = 13;
this->label4->Text = L"Guide Wire";
//
// label3
//
this->label3->AutoSize = true;
this->label3->Location = System::Drawing::Point(6, 156);
this->label3->Name = L"label3";
this->label3->Size = System::Drawing::Size(65, 13);
this->label3->TabIndex = 12;
this->label3->Text = L"Grid Voltage";
//
// label2
//
this->label2->AutoSize = true;
this->label2->Location = System::Drawing::Point(6, 130);
this->label2->Name = L"label2";
this->label2->Size = System::Drawing::Size(105, 13);
this->label2->TabIndex = 11;
this->label2->Text = L"Accelerating Voltage";
//
// tbLaserIntensity
//
this->tbLaserIntensity->Location = System::Drawing::Point(315, 49);
this->tbLaserIntensity->Maximum = 4600;
this->tbLaserIntensity->Name = L"tbLaserIntensity";
this->tbLaserIntensity->Size = System::Drawing::Size(299, 45);
this->tbLaserIntensity->TabIndex = 10;
this->tbLaserIntensity->TickFrequency = 100;
this->tbLaserIntensity->Scroll += gcnew System::EventHandler(this,
&Form1::tbLaserIntensity_Scroll);
//
// label1
//
this->label1->AutoSize = true;
this->label1->Location = System::Drawing::Point(6, 33);
this->label1->Name = L"label1";
this->label1->Size = System::Drawing::Size(75, 13);
this->label1->TabIndex = 9;
this->label1->Text = L"Laser Intensity";
//
// edLowMassGateMass

```

```

//
this->edLowMassGateMass->Location = System::Drawing::Point(268, 313);
this->edLowMassGateMass->Name = L"edLowMassGateMass";
this->edLowMassGateMass->Size = System::Drawing::Size(175, 20);
this->edLowMassGateMass->TabIndex = 8;
//
// edAcqMassMin
//
this->edAcqMassMin->Location = System::Drawing::Point(134, 257);
this->edAcqMassMin->Name = L"edAcqMassMin";
this->edAcqMassMin->Size = System::Drawing::Size(175, 20);
this->edAcqMassMin->TabIndex = 7;
//
// edAcqMassMax
//
this->edAcqMassMax->Location = System::Drawing::Point(341, 257);
this->edAcqMassMax->Name = L"edAcqMassMax";
this->edAcqMassMax->Size = System::Drawing::Size(175, 20);
this->edAcqMassMax->TabIndex = 6;
//
// edTotalLaserShots
//
this->edTotalLaserShots->Location = System::Drawing::Point(134, 231);
this->edTotalLaserShots->Name = L"edTotalLaserShots";
this->edTotalLaserShots->Size = System::Drawing::Size(175, 20);
this->edTotalLaserShots->TabIndex = 5;
//
// edDelayTime
//
this->edDelayTime->Location = System::Drawing::Point(134, 205);
this->edDelayTime->Name = L"edDelayTime";
this->edDelayTime->Size = System::Drawing::Size(175, 20);
this->edDelayTime->TabIndex = 4;
//
// edGuideWireVoltage0
//
this->edGuideWireVoltage0->Location = System::Drawing::Point(134, 179);
this->edGuideWireVoltage0->Name = L"edGuideWireVoltage0";
this->edGuideWireVoltage0->Size = System::Drawing::Size(175, 20);
this->edGuideWireVoltage0->TabIndex = 3;
//
// edGridVoltage
//
this->edGridVoltage->Location = System::Drawing::Point(134, 153);
this->edGridVoltage->Name = L"edGridVoltage";
this->edGridVoltage->Size = System::Drawing::Size(175, 20);
this->edGridVoltage->TabIndex = 2;
//
// edAcceleratingVoltage
//
this->edAcceleratingVoltage->Location = System::Drawing::Point(134, 127);
this->edAcceleratingVoltage->Name = L"edAcceleratingVoltage";
this->edAcceleratingVoltage->Size = System::Drawing::Size(175, 20);
this->edAcceleratingVoltage->TabIndex = 1;
//
// edLaserIntensity
//
this->edLaserIntensity->Location = System::Drawing::Point(134, 30);
this->edLaserIntensity->Name = L"edLaserIntensity";
this->edLaserIntensity->Size = System::Drawing::Size(175, 20);
this->edLaserIntensity->TabIndex = 0;
//
// tsPLT
//
this->tsPLT->BackColor = System::Drawing::Color::LightBlue;
this->tsPLT->Controls->Add(this->bClearPLT);
this->tsPLT->Controls->Add(this->bCreerPLT);
this->tsPLT->Controls->Add(this->groupBox4);
this->tsPLT->Controls->Add(this->groupBox5);
this->tsPLT->Controls->Add(this->groupBox3);
this->tsPLT->Controls->Add(this->Coordinates);

```

```

this->tsPLT->Location = System::Drawing::Point(4, 22);
this->tsPLT->Name = L"tsPLT";
this->tsPLT->Padding = System::Windows::Forms::Padding(3);
this->tsPLT->Size = System::Drawing::Size(620, 426);
this->tsPLT->TabIndex = 2;
this->tsPLT->Text = L"PLT File";
this->tsPLT->UseVisualStyleBackColor = true;
//
// bClearPLT
//
this->bClearPLT->Location = System::Drawing::Point(312, 397);
this->bClearPLT->Name = L"bClearPLT";
this->bClearPLT->Size = System::Drawing::Size(75, 23);
this->bClearPLT->TabIndex = 3;
this->bClearPLT->Text = L"Clear PLT";
this->bClearPLT->UseVisualStyleBackColor = true;
this->bClearPLT->Click += gcnew System::EventHandler(this,
&Form1::bClearPLT_Click);
//
// bCreerPLT
//
this->bCreerPLT->Location = System::Drawing::Point(169, 397);
this->bCreerPLT->Name = L"bCreerPLT";
this->bCreerPLT->Size = System::Drawing::Size(93, 23);
this->bCreerPLT->TabIndex = 2;
this->bCreerPLT->Text = L"Create the file";
this->bCreerPLT->UseVisualStyleBackColor = true;
this->bCreerPLT->Click += gcnew System::EventHandler(this,
&Form1::bCreerPLT_Click);
//
// groupBox4
//
this->groupBox4->Controls->Add(this->bNumberPoints);
this->groupBox4->Controls->Add(this->lNoValuePointsRow);
this->groupBox4->Controls->Add(this->lNbPtsV);
this->groupBox4->Controls->Add(this->lNbPtsH);
this->groupBox4->Controls->Add(this->eNbPtsV);
this->groupBox4->Controls->Add(this->eNbPtsH);
this->groupBox4->Location = System::Drawing::Point(9, 214);
this->groupBox4->Name = L"groupBox4";
this->groupBox4->Size = System::Drawing::Size(608, 100);
this->groupBox4->TabIndex = 1;
this->groupBox4->TabStop = false;
this->groupBox4->Text = L"NUMBER OF POINTS";
//
// bNumberPoints
//
this->bNumberPoints->Location = System::Drawing::Point(127, 0);
this->bNumberPoints->Name = L"bNumberPoints";
this->bNumberPoints->Size = System::Drawing::Size(60, 21);
this->bNumberPoints->TabIndex = 10;
this->bNumberPoints->Text = L"OK";
this->bNumberPoints->UseVisualStyleBackColor = true;
this->bNumberPoints->Click += gcnew System::EventHandler(this,
&Form1::bNumberPoints_Click);
//
// lNoValuePointsRow
//
this->lNoValuePointsRow->AutoSize = true;
this->lNoValuePointsRow->Location = System::Drawing::Point(200, 75);
this->lNoValuePointsRow->Name = L"lNoValuePointsRow";
this->lNoValuePointsRow->Size = System::Drawing::Size(399, 13);
this->lNoValuePointsRow->TabIndex = 9;
this->lNoValuePointsRow->Text = L"(if you don't give this value it will
be the same that the number of points for "
L"one line)";
//
// lNbPtsV
//
this->lNbPtsV->AutoSize = true;
this->lNbPtsV->Location = System::Drawing::Point(157, 51);

```

```

this->lNbPtsV->Name = L"lNbPtsV";
this->lNbPtsV->Size = System::Drawing::Size(171, 13);
this->lNbPtsV->TabIndex = 8;
this->lNbPtsV->Text = L"    and number of points for one row";
//
// lNbPtsH
//
this->lNbPtsH->AutoSize = true;
this->lNbPtsH->Location = System::Drawing::Point(186, 23);
this->lNbPtsH->Name = L"lNbPtsH";
this->lNbPtsH->Size = System::Drawing::Size(142, 13);
this->lNbPtsH->TabIndex = 7;
this->lNbPtsH->Text = L"Number of points for one line";
//
// eNbPtsV
//
this->eNbPtsV->Location = System::Drawing::Point(334, 48);
this->eNbPtsV->Name = L"eNbPtsV";
this->eNbPtsV->Size = System::Drawing::Size(152, 20);
this->eNbPtsV->TabIndex = 6;
//
// eNbPtsH
//
this->eNbPtsH->Location = System::Drawing::Point(334, 20);
this->eNbPtsH->Name = L"eNbPtsH";
this->eNbPtsH->Size = System::Drawing::Size(152, 20);
this->eNbPtsH->TabIndex = 5;
//
// groupBox5
//
this->groupBox5->Controls->Add(this->rbComplexTrajectory);
this->groupBox5->Controls->Add(this->rbSimpleTrajectory);
this->groupBox5->Location = System::Drawing::Point(9, 320);
this->groupBox5->Name = L"groupBox5";
this->groupBox5->Size = System::Drawing::Size(608, 69);
this->groupBox5->TabIndex = 1;
this->groupBox5->TabStop = false;
this->groupBox5->Text = L"TRAJECTORY OF THE LASER";
//
// rbComplexTrajectory
//
this->rbComplexTrajectory->AutoSize = true;
this->rbComplexTrajectory->Location = System::Drawing::Point(193, 38);
this->rbComplexTrajectory->Name = L"rbComplexTrajectory";
this->rbComplexTrajectory->Size = System::Drawing::Size(322, 17);
this->rbComplexTrajectory->TabIndex = 1;
this->rbComplexTrajectory->TabStop = true;
this->rbComplexTrajectory->Text = L"From left to right for odd lines and
from right to left for even lines";
this->rbComplexTrajectory->UseVisualStyleBackColor = true;
//
// rbSimpleTrajectory
//
this->rbSimpleTrajectory->AutoSize = true;
this->rbSimpleTrajectory->Location = System::Drawing::Point(193, 15);
this->rbSimpleTrajectory->Name = L"rbSimpleTrajectory";
this->rbSimpleTrajectory->Size = System::Drawing::Size(170, 17);
this->rbSimpleTrajectory->TabIndex = 0;
this->rbSimpleTrajectory->TabStop = true;
this->rbSimpleTrajectory->Text = L"From left to right for all the lines";
this->rbSimpleTrajectory->UseVisualStyleBackColor = true;
//
// groupBox3
//
this->groupBox3->Controls->Add(this->lNoValueVerticalStep);
this->groupBox3->Controls->Add(this->bStep);
this->groupBox3->Controls->Add(this->lPasV);
this->groupBox3->Controls->Add(this->lPasH);
this->groupBox3->Controls->Add(this->ePasV);
this->groupBox3->Controls->Add(this->ePasH);
this->groupBox3->Location = System::Drawing::Point(6, 108);

```

```

this->groupBox3->Name = L"groupBox3";
this->groupBox3->Size = System::Drawing::Size(608, 100);
this->groupBox3->TabIndex = 1;
this->groupBox3->TabStop = false;
this->groupBox3->Text = L"STEP";
//
// lNoValueVerticalStep
//
this->lNoValueVerticalStep->AutoSize = true;
this->lNoValueVerticalStep->Location = System::Drawing::Point(286, 70);
this->lNoValueVerticalStep->Name = L"lNoValueVerticalStep";
this->lNoValueVerticalStep->Size = System::Drawing::Size(316, 13);
this->lNoValueVerticalStep->TabIndex = 6;
this->lNoValueVerticalStep->Text = L"(if you don't give this value it
will be the same that horizontal step)";
//
// bStep
//
this->bStep->Location = System::Drawing::Point(53, 0);
this->bStep->Name = L"bStep";
this->bStep->Size = System::Drawing::Size(60, 21);
this->bStep->TabIndex = 5;
this->bStep->Text = L"OK";
this->bStep->UseVisualStyleBackColor = true;
this->bStep->Click += gcnew System::EventHandler(this,
&Form1::bStep_Click);
//
// lPasV
//
this->lPasV->AutoSize = true;
this->lPasV->Location = System::Drawing::Point(303, 46);
this->lPasV->Name = L"lPasV";
this->lPasV->Size = System::Drawing::Size(65, 13);
this->lPasV->TabIndex = 4;
this->lPasV->Text = L"Vertical step";
//
// lPasH
//
this->lPasH->AutoSize = true;
this->lPasH->Location = System::Drawing::Point(6, 42);
this->lPasH->Name = L"lPasH";
this->lPasH->Size = System::Drawing::Size(77, 13);
this->lPasH->TabIndex = 3;
this->lPasH->Text = L"Horizontal step";
//
// ePasV
//
this->ePasV->Location = System::Drawing::Point(374, 39);
this->ePasV->Name = L"ePasV";
this->ePasV->Size = System::Drawing::Size(152, 20);
this->ePasV->TabIndex = 2;
//
// ePasH
//
this->ePasH->Location = System::Drawing::Point(89, 39);
this->ePasH->Name = L"ePasH";
this->ePasH->Size = System::Drawing::Size(152, 20);
this->ePasH->TabIndex = 1;
//
// Coordinates
//
this->Coordinates->Controls->Add(this->lYArr);
this->Coordinates->Controls->Add(this->lXArr);
this->Coordinates->Controls->Add(this->lYDep);
this->Coordinates->Controls->Add(this->lXDep);
this->Coordinates->Controls->Add(this->lcoordarr);
this->Coordinates->Controls->Add(this->lcoorddep);
this->Coordinates->Controls->Add(this->eYArr);
this->Coordinates->Controls->Add(this->eYDep);
this->Coordinates->Controls->Add(this->eXArr);
this->Coordinates->Controls->Add(this->eXDep);

```

```

this->Coordinates->Location = System::Drawing::Point(6, 14);
this->Coordinates->Name = L"Coordinates";
this->Coordinates->Size = System::Drawing::Size(608, 88);
this->Coordinates->TabIndex = 0;
this->Coordinates->TabStop = false;
this->Coordinates->Text = L"COORDINATES";
//
// lYArr
//
this->lYArr->AutoSize = true;
this->lYArr->Location = System::Drawing::Point(401, 60);
this->lYArr->Name = L"lYArr";
this->lYArr->Size = System::Drawing::Size(14, 13);
this->lYArr->TabIndex = 9;
this->lYArr->Text = L"Y";
//
// lXArr
//
this->lXArr->AutoSize = true;
this->lXArr->Location = System::Drawing::Point(176, 56);
this->lXArr->Name = L"lXArr";
this->lXArr->Size = System::Drawing::Size(14, 13);
this->lXArr->TabIndex = 8;
this->lXArr->Text = L"X";
//
// lYDep
//
this->lYDep->AutoSize = true;
this->lYDep->Location = System::Drawing::Point(401, 30);
this->lYDep->Name = L"lYDep";
this->lYDep->Size = System::Drawing::Size(14, 13);
this->lYDep->TabIndex = 7;
this->lYDep->Text = L"Y";
//
// lXDep
//
this->lXDep->AutoSize = true;
this->lXDep->Location = System::Drawing::Point(176, 30);
this->lXDep->Name = L"lXDep";
this->lXDep->Size = System::Drawing::Size(14, 13);
this->lXDep->TabIndex = 6;
this->lXDep->Text = L"X";
//
// lcoordarr
//
this->lcoordarr->AutoSize = true;
this->lcoordarr->Location = System::Drawing::Point(6, 56);
this->lcoordarr->Name = L"lcoordarr";
this->lcoordarr->Size = System::Drawing::Size(127, 13);
this->lcoordarr->TabIndex = 5;
this->lcoordarr->Text = L" Ending point coordinates";
//
// lcoorddep
//
this->lcoorddep->AutoSize = true;
this->lcoorddep->Location = System::Drawing::Point(6, 30);
this->lcoorddep->Name = L"lcoorddep";
this->lcoorddep->Size = System::Drawing::Size(127, 13);
this->lcoorddep->TabIndex = 4;
this->lcoorddep->Text = L"Starting point coordinates";
//
// eYArr
//
this->eYArr->Location = System::Drawing::Point(421, 53);
this->eYArr->Name = L"eYArr";
this->eYArr->Size = System::Drawing::Size(152, 20);
this->eYArr->TabIndex = 3;
//
// eYDep
//
this->eYDep->Location = System::Drawing::Point(421, 27);

```

```

this->eYDep->Name = L"eYDep";
this->eYDep->Size = System::Drawing::Size(152, 20);
this->eYDep->TabIndex = 2;
//
// eXArr
//
this->eXArr->Location = System::Drawing::Point(196, 53);
this->eXArr->Name = L"eXArr";
this->eXArr->Size = System::Drawing::Size(152, 20);
this->eXArr->TabIndex = 1;
//
// eXDep
//
this->eXDep->Location = System::Drawing::Point(196, 27);
this->eXDep->Name = L"eXDep";
this->eXDep->Size = System::Drawing::Size(152, 20);
this->eXDep->TabIndex = 0;
//
// tsPLTbis
//
this->tsPLTbis->BackColor = System::Drawing::Color::LightBlue;
this->tsPLTbis->Controls->Add(this->pictureBoxCircle1);
this->tsPLTbis->Controls->Add(this->label25);
this->tsPLTbis->Controls->Add(this->label24);
this->tsPLTbis->Controls->Add(this->label23);
this->tsPLTbis->Controls->Add(this->label22);
this->tsPLTbis->Controls->Add(this->label21);
this->tsPLTbis->Controls->Add(this->eNbPtsHbis);
this->tsPLTbis->Controls->Add(this->ePasVbis);
this->tsPLTbis->Controls->Add(this->ePsHbis);
this->tsPLTbis->Controls->Add(this->eNbPtsVbis);
this->tsPLTbis->Controls->Add(this->iPhoto);
this->tsPLTbis->Controls->Add(this->groupBox2);
this->tsPLTbis->Location = System::Drawing::Point(4, 22);
this->tsPLTbis->Name = L"tsPLTbis";
this->tsPLTbis->Padding = System::Windows::Forms::Padding(3);
this->tsPLTbis->Size = System::Drawing::Size(620, 426);
this->tsPLTbis->TabIndex = 3;
this->tsPLTbis->Text = L"PLT file by means of a picture";
this->tsPLTbis->UseVisualStyleBackColor = true;
//
// pictureBoxCircle1
//
this->pictureBoxCircle1->BorderStyle =
System::Windows::Forms::BorderStyle::FixedSingle;
this->pictureBoxCircle1->Image = (cli::safe_cast<System::Drawing::Image^
>(resources->GetObject(L"pictureBoxCircle1.Image")));
this->pictureBoxCircle1->Location = System::Drawing::Point(169, 117);
this->pictureBoxCircle1->Name = L"pictureBoxCircle1";
this->pictureBoxCircle1->Size = System::Drawing::Size(22, 22);
this->pictureBoxCircle1->TabIndex = 11;
this->pictureBoxCircle1->TabStop = false;
this->pictureBoxCircle1->MouseMove += gcnew
System::Windows::Forms::EventHandler(this, &Form1::pictureBoxCircle1_MouseMove);
this->pictureBoxCircle1->MouseDown += gcnew
System::Windows::Forms::EventHandler(this, &Form1::pictureBoxCircle1_MouseDown);
this->pictureBoxCircle1->MouseUp += gcnew
System::Windows::Forms::EventHandler(this, &Form1::pictureBoxCircle1_MouseUp);
//
// label25
//
this->label25->AutoSize = true;
this->label25->Location = System::Drawing::Point(6, 284);
this->label25->Name = L"label25";
this->label25->Size = System::Drawing::Size(87, 13);
this->label25->TabIndex = 10;
this->label25->Text = L"Number of points";
//
// label24
//
this->label24->AutoSize = true;

```

```

this->label24->Location = System::Drawing::Point(6, 376);
this->label24->Name = L"label24";
this->label24->Size = System::Drawing::Size(60, 13);
this->label24->TabIndex = 9;
this->label24->Text = L"on one row";
//
// label23
//
this->label23->AutoSize = true;
this->label23->Location = System::Drawing::Point(6, 316);
this->label23->Name = L"label23";
this->label23->Size = System::Drawing::Size(59, 13);
this->label23->TabIndex = 8;
this->label23->Text = L"on one line";
//
// label22
//
this->label22->AutoSize = true;
this->label22->Location = System::Drawing::Point(6, 195);
this->label22->Name = L"label22";
this->label22->Size = System::Drawing::Size(65, 13);
this->label22->TabIndex = 7;
this->label22->Text = L"Vertical step";
//
// label21
//
this->label21->AutoSize = true;
this->label21->Location = System::Drawing::Point(6, 130);
this->label21->Name = L"label21";
this->label21->Size = System::Drawing::Size(77, 13);
this->label21->TabIndex = 6;
this->label21->Text = L"Horizontal step";
//
// eNbPtsHbis
//
this->eNbPtsHbis->Location = System::Drawing::Point(6, 332);
this->eNbPtsHbis->Name = L"eNbPtsHbis";
this->eNbPtsHbis->Size = System::Drawing::Size(100, 20);
this->eNbPtsHbis->TabIndex = 5;
//
// ePasVbis
//
this->ePasVbis->Location = System::Drawing::Point(6, 211);
this->ePasVbis->Name = L"ePasVbis";
this->ePasVbis->Size = System::Drawing::Size(100, 20);
this->ePasVbis->TabIndex = 4;
//
// ePsHbis
//
this->ePsHbis->Location = System::Drawing::Point(6, 147);
this->ePsHbis->Name = L"ePsHbis";
this->ePsHbis->Size = System::Drawing::Size(100, 20);
this->ePsHbis->TabIndex = 3;
//
// eNbPtsVbis
//
this->eNbPtsVbis->Location = System::Drawing::Point(6, 392);
this->eNbPtsVbis->Name = L"eNbPtsVbis";
this->eNbPtsVbis->Size = System::Drawing::Size(100, 20);
this->eNbPtsVbis->TabIndex = 2;
//
// iPhoto
//
this->iPhoto->Location = System::Drawing::Point(112, 87);
this->iPhoto->Name = L"iPhoto";
this->iPhoto->Size = System::Drawing::Size(502, 333);
this->iPhoto->SizeMode = System::Windows::Forms::PictureBoxSizeMode::Zoom;
this->iPhoto->TabIndex = 1;
this->iPhoto->TabStop = false;
this->iPhoto->MouseMove += gcnew
System::Windows::Forms::EventHandler(this, &Form1::iPhoto_MouseMove);

```

```

        this->iPhoto->MouseDown += gcnew
System::Windows::Forms::EventHandler(this, &Form1::iPhoto_MouseDown);
        this->iPhoto->MouseUp += gcnew
System::Windows::Forms::EventHandler(this, &Form1::iPhoto_MouseUp);
        //
        // groupBox2
        //
        this->groupBox2->Controls->Add(this->bPhoto);
        this->groupBox2->Controls->Add(this->ePhoto);
        this->groupBox2->Location = System::Drawing::Point(112, 14);
        this->groupBox2->Name = L"groupBox2";
        this->groupBox2->Size = System::Drawing::Size(502, 67);
        this->groupBox2->TabIndex = 0;
        this->groupBox2->TabStop = false;
        this->groupBox2->Text = L"File of te sample picture";
        //
        // bPhoto
        //
        this->bPhoto->Location = System::Drawing::Point(421, 25);
        this->bPhoto->Name = L"bPhoto";
        this->bPhoto->Size = System::Drawing::Size(75, 23);
        this->bPhoto->TabIndex = 5;
        this->bPhoto->Text = L"Browse";
        this->bPhoto->UseVisualStyleBackColor = true;
        this->bPhoto->Click += gcnew System::EventHandler(this,
&Form1::bPhoto_Click);
        //
        // ePhoto
        //
        this->ePhoto->Location = System::Drawing::Point(6, 27);
        this->ePhoto->Name = L"ePhoto";
        this->ePhoto->Size = System::Drawing::Size(409, 20);
        this->ePhoto->TabIndex = 4;
        //
        // tsSEQ
        //
        this->tsSEQ->BackColor = System::Drawing::Color::LightBlue;
        this->tsSEQ->Controls->Add(this->bCreerFichierSEQ);
        this->tsSEQ->Controls->Add(this->bClearSEQ);
        this->tsSEQ->Controls->Add(this->bFurther);
        this->tsSEQ->Controls->Add(this->bMacros);
        this->tsSEQ->Controls->Add(this->bFiles);
        this->tsSEQ->Location = System::Drawing::Point(4, 22);
        this->tsSEQ->Name = L"tsSEQ";
        this->tsSEQ->Padding = System::Windows::Forms::Padding(3);
        this->tsSEQ->Size = System::Drawing::Size(620, 426);
        this->tsSEQ->TabIndex = 4;
        this->tsSEQ->Text = L"SEQ file";
        this->tsSEQ->UseVisualStyleBackColor = true;
        //
        // bCreerFichierSEQ
        //
        this->bCreerFichierSEQ->Location = System::Drawing::Point(131, 390);
        this->bCreerFichierSEQ->Name = L"bCreerFichierSEQ";
        this->bCreerFichierSEQ->Size = System::Drawing::Size(102, 23);
        this->bCreerFichierSEQ->TabIndex = 4;
        this->bCreerFichierSEQ->Text = L"Create the file";
        this->bCreerFichierSEQ->UseVisualStyleBackColor = true;
        this->bCreerFichierSEQ->Click += gcnew System::EventHandler(this,
&Form1::bCreerFichierSEQ_Click);
        //
        // bClearSEQ
        //
        this->bClearSEQ->Location = System::Drawing::Point(379, 390);
        this->bClearSEQ->Name = L"bClearSEQ";
        this->bClearSEQ->Size = System::Drawing::Size(75, 23);
        this->bClearSEQ->TabIndex = 3;
        this->bClearSEQ->Text = L"Clear SEQ";
        this->bClearSEQ->UseVisualStyleBackColor = true;
        this->bClearSEQ->Click += gcnew System::EventHandler(this,
&Form1::bClearSEQ_Click);

```

```

//
// bFurther
//
this->bFurther->Controls->Add(this->lComments);
this->bFurther->Controls->Add(this->lSampleDescription);
this->bFurther->Controls->Add(this->eComments);
this->bFurther->Controls->Add(this->eSampleDescription);
this->bFurther->Location = System::Drawing::Point(6, 285);
this->bFurther->Name = L"bFurther";
this->bFurther->Size = System::Drawing::Size(608, 91);
this->bFurther->TabIndex = 2;
this->bFurther->TabStop = false;
this->bFurther->Text = L"FURTHER";
//
// lComments
//
this->lComments->AutoSize = true;
this->lComments->Location = System::Drawing::Point(11, 58);
this->lComments->Name = L"lComments";
this->lComments->Size = System::Drawing::Size(56, 13);
this->lComments->TabIndex = 3;
this->lComments->Text = L"Comments";
//
// lSampleDescription
//
this->lSampleDescription->AutoSize = true;
this->lSampleDescription->Location = System::Drawing::Point(11, 31);
this->lSampleDescription->Name = L"lSampleDescription";
this->lSampleDescription->Size = System::Drawing::Size(96, 13);
this->lSampleDescription->TabIndex = 2;
this->lSampleDescription->Text = L"Sample description";
//
// eComments
//
this->eComments->Location = System::Drawing::Point(125, 55);
this->eComments->Name = L"eComments";
this->eComments->Size = System::Drawing::Size(477, 20);
this->eComments->TabIndex = 1;
//
// eSampleDescription
//
this->eSampleDescription->Location = System::Drawing::Point(125, 28);
this->eSampleDescription->Name = L"eSampleDescription";
this->eSampleDescription->Size = System::Drawing::Size(477, 20);
this->eSampleDescription->TabIndex = 0;
//
// bMacros
//
this->bMacros->Controls->Add(this->lPostMacro);
this->bMacros->Controls->Add(this->lPreMacro);
this->bMacros->Controls->Add(this->cbPostMacro);
this->bMacros->Controls->Add(this->cbPreMacro);
this->bMacros->Location = System::Drawing::Point(6, 185);
this->bMacros->Name = L"bMacros";
this->bMacros->Size = System::Drawing::Size(608, 94);
this->bMacros->TabIndex = 1;
this->bMacros->TabStop = false;
this->bMacros->Text = L"Macros";
//
// lPostMacro
//
this->lPostMacro->AutoSize = true;
this->lPostMacro->Location = System::Drawing::Point(11, 59);
this->lPostMacro->Name = L"lPostMacro";
this->lPostMacro->Size = System::Drawing::Size(60, 13);
this->lPostMacro->TabIndex = 3;
this->lPostMacro->Text = L"Post macro";
//
// lPreMacro
//
this->lPreMacro->AutoSize = true;

```

```

this->lPreMacro->Location = System::Drawing::Point(11, 32);
this->lPreMacro->Name = L"lPreMacro";
this->lPreMacro->Size = System::Drawing::Size(55, 13);
this->lPreMacro->TabIndex = 2;
this->lPreMacro->Text = L"Pre macro";
//
// cbPostMacro
//
this->cbPostMacro->FormattingEnabled = true;
this->cbPostMacro->Location = System::Drawing::Point(125, 56);
this->cbPostMacro->Name = L"cbPostMacro";
this->cbPostMacro->Size = System::Drawing::Size(477, 21);
this->cbPostMacro->TabIndex = 1;
//
// cbPreMacro
//
this->cbPreMacro->FormattingEnabled = true;
this->cbPreMacro->Location = System::Drawing::Point(125, 29);
this->cbPreMacro->Name = L"cbPreMacro";
this->cbPreMacro->Size = System::Drawing::Size(477, 21);
this->cbPreMacro->TabIndex = 0;
//
// bFiles
//
this->bFiles->Controls->Add(this->eNbLignes);
this->bFiles->Controls->Add(this->lNumberLines);
this->bFiles->Controls->Add(this->lInstrumentSettingFile);
this->bFiles->Controls->Add(this->lDataFile);
this->bFiles->Controls->Add(this->lSequenceType);
this->bFiles->Controls->Add(this->bParcourirISF);
this->bFiles->Controls->Add(this->bParcourirDF);
this->bFiles->Controls->Add(this->bParcourirST);
this->bFiles->Controls->Add(this->eDataFile);
this->bFiles->Controls->Add(this->eInstrumentSettingFile);
this->bFiles->Controls->Add(this->eSequenceType);
this->bFiles->Location = System::Drawing::Point(6, 6);
this->bFiles->Name = L"bFiles";
this->bFiles->Size = System::Drawing::Size(608, 173);
this->bFiles->TabIndex = 0;
this->bFiles->TabStop = false;
this->bFiles->Text = L"Files";
//
// eNbLignes
//
this->eNbLignes->Enabled = false;
this->eNbLignes->Location = System::Drawing::Point(314, 13);
this->eNbLignes->Name = L"eNbLignes";
this->eNbLignes->Size = System::Drawing::Size(122, 20);
this->eNbLignes->TabIndex = 10;
//
// lNumberLines
//
this->lNumberLines->AutoSize = true;
this->lNumberLines->ForeColor = System::Drawing::Color::DarkGoldenrod;
this->lNumberLines->Location = System::Drawing::Point(182, 16);
this->lNumberLines->Name = L"lNumberLines";
this->lNumberLines->Size = System::Drawing::Size(126, 13);
this->lNumberLines->TabIndex = 9;
this->lNumberLines->Text = L"Number of lines in this file";
//
// lInstrumentSettingFile
//
this->lInstrumentSettingFile->AutoSize = true;
this->lInstrumentSettingFile->Location = System::Drawing::Point(6, 114);
this->lInstrumentSettingFile->Name = L"lInstrumentSettingFile";
this->lInstrumentSettingFile->Size = System::Drawing::Size(106, 13);
this->lInstrumentSettingFile->TabIndex = 8;
this->lInstrumentSettingFile->Text = L"Instrument setting file";
//
// lDataFile
//

```

```

this->lDataFile->AutoSize = true;
this->lDataFile->Location = System::Drawing::Point(6, 75);
this->lDataFile->Name = L"lDataFile";
this->lDataFile->Size = System::Drawing::Size(46, 13);
this->lDataFile->TabIndex = 7;
this->lDataFile->Text = L"Data file";
//
// lSequenceType
//
this->lSequenceType->AutoSize = true;
this->lSequenceType->Location = System::Drawing::Point(6, 35);
this->lSequenceType->Name = L"lSequenceType";
this->lSequenceType->Size = System::Drawing::Size(79, 13);
this->lSequenceType->TabIndex = 6;
this->lSequenceType->Text = L"Sequence type";
//
// bParcourirISF
//
this->bParcourirISF->Location = System::Drawing::Point(527, 128);
this->bParcourirISF->Name = L"bParcourirISF";
this->bParcourirISF->Size = System::Drawing::Size(75, 23);
this->bParcourirISF->TabIndex = 5;
this->bParcourirISF->Text = L"Browse";
this->bParcourirISF->UseVisualStyleBackColor = true;
this->bParcourirISF->Click += gcnew System::EventHandler(this,
&Form1::bParcourirISF_Click);
//
// bParcourirDF
//
this->bParcourirDF->Location = System::Drawing::Point(527, 89);
this->bParcourirDF->Name = L"bParcourirDF";
this->bParcourirDF->Size = System::Drawing::Size(75, 23);
this->bParcourirDF->TabIndex = 4;
this->bParcourirDF->Text = L"Browse";
this->bParcourirDF->UseVisualStyleBackColor = true;
this->bParcourirDF->Click += gcnew System::EventHandler(this,
&Form1::bParcourirDF_Click);
//
// bParcourirST
//
this->bParcourirST->Location = System::Drawing::Point(527, 49);
this->bParcourirST->Name = L"bParcourirST";
this->bParcourirST->Size = System::Drawing::Size(75, 23);
this->bParcourirST->TabIndex = 3;
this->bParcourirST->Text = L"Browse";
this->bParcourirST->UseVisualStyleBackColor = true;
this->bParcourirST->Click += gcnew System::EventHandler(this,
&Form1::bParcourirST_Click);
//
// eDataFile
//
this->eDataFile->Location = System::Drawing::Point(6, 91);
this->eDataFile->Name = L"eDataFile";
this->eDataFile->Size = System::Drawing::Size(515, 20);
this->eDataFile->TabIndex = 2;
//
// eInstrumentSettingFile
//
this->eInstrumentSettingFile->Location = System::Drawing::Point(6, 130);
this->eInstrumentSettingFile->Name = L"eInstrumentSettingFile";
this->eInstrumentSettingFile->Size = System::Drawing::Size(515, 20);
this->eInstrumentSettingFile->TabIndex = 1;
//
// eSequenceType
//
this->eSequenceType->Location = System::Drawing::Point(6, 51);
this->eSequenceType->Name = L"eSequenceType";
this->eSequenceType->Size = System::Drawing::Size(515, 20);
this->eSequenceType->TabIndex = 0;
//
// tsConvert

```

```

//
this->tsConvert->BackColor = System::Drawing::Color::LightBlue;
this->tsConvert->Controls->Add(this->bConvert);
this->tsConvert->Controls->Add(this->bGlobalFile);
this->tsConvert->Controls->Add(this->bFolderOfTheFiles);
this->tsConvert->Controls->Add(this->bPointsCoordinatesFile);
this->tsConvert->Controls->Add(this->eGlobalFile);
this->tsConvert->Controls->Add(this->eFolderOfTheFiles);
this->tsConvert->Controls->Add(this->ePointsCoordinatesFile);
this->tsConvert->Controls->Add(this->lGlobalFile);
this->tsConvert->Controls->Add(this->lFolderOfTheFiles);
this->tsConvert->Controls->Add(this->lPointsCoordinatesFile);
this->tsConvert->Location = System::Drawing::Point(4, 22);
this->tsConvert->Name = L"tsConvert";
this->tsConvert->Padding = System::Windows::Forms::Padding(3);
this->tsConvert->Size = System::Drawing::Size(620, 426);
this->tsConvert->TabIndex = 5;
this->tsConvert->Text = L"Convert acquisition files";
this->tsConvert->UseVisualStyleBackColor = true;
//
// bConvert
//
this->bConvert->Location = System::Drawing::Point(202, 332);
this->bConvert->Name = L"bConvert";
this->bConvert->Size = System::Drawing::Size(188, 23);
this->bConvert->TabIndex = 9;
this->bConvert->Text = L"Create the file";
this->bConvert->UseVisualStyleBackColor = true;
this->bConvert->Click += gcnew System::EventHandler(this,
&Form1::bConvert_Click);
//
// bGlobalFile
//
this->bGlobalFile->Location = System::Drawing::Point(495, 181);
this->bGlobalFile->Name = L"bGlobalFile";
this->bGlobalFile->Size = System::Drawing::Size(75, 23);
this->bGlobalFile->TabIndex = 8;
this->bGlobalFile->Text = L"Browse";
this->bGlobalFile->UseVisualStyleBackColor = true;
this->bGlobalFile->Click += gcnew System::EventHandler(this,
&Form1::bGlobalFile_Click);
//
// bFolderOfTheFiles
//
this->bFolderOfTheFiles->Location = System::Drawing::Point(495, 115);
this->bFolderOfTheFiles->Name = L"bFolderOfTheFiles";
this->bFolderOfTheFiles->Size = System::Drawing::Size(75, 23);
this->bFolderOfTheFiles->TabIndex = 7;
this->bFolderOfTheFiles->Text = L"Browse";
this->bFolderOfTheFiles->UseVisualStyleBackColor = true;
this->bFolderOfTheFiles->Click += gcnew System::EventHandler(this,
&Form1::bFolderOfTheFiles_Click);
//
// bPointsCoordinatesFile
//
this->bPointsCoordinatesFile->Location = System::Drawing::Point(495, 54);
this->bPointsCoordinatesFile->Name = L"bPointsCoordinatesFile";
this->bPointsCoordinatesFile->Size = System::Drawing::Size(75, 23);
this->bPointsCoordinatesFile->TabIndex = 6;
this->bPointsCoordinatesFile->Text = L"Browse";
this->bPointsCoordinatesFile->UseVisualStyleBackColor = true;
this->bPointsCoordinatesFile->Click += gcnew System::EventHandler(this,
&Form1::bPointsCoordinatesFile_Click);
//
// eGlobalFile
//
this->eGlobalFile->Location = System::Drawing::Point(67, 184);
this->eGlobalFile->Name = L"eGlobalFile";
this->eGlobalFile->Size = System::Drawing::Size(404, 20);
this->eGlobalFile->TabIndex = 5;
//

```

```

// eFolderOfTheFiles
//
this->eFolderOfTheFiles->Location = System::Drawing::Point(67, 117);
this->eFolderOfTheFiles->Name = L"eFolderOfTheFiles";
this->eFolderOfTheFiles->Size = System::Drawing::Size(404, 20);
this->eFolderOfTheFiles->TabIndex = 4;
//
// ePointsCoordinatesFile
//
this->ePointsCoordinatesFile->Location = System::Drawing::Point(67, 57);
this->ePointsCoordinatesFile->Name = L"ePointsCoordinatesFile";
this->ePointsCoordinatesFile->Size = System::Drawing::Size(404, 20);
this->ePointsCoordinatesFile->TabIndex = 3;
//
// lGlobalFile
//
this->lGlobalFile->AutoSize = true;
this->lGlobalFile->Location = System::Drawing::Point(64, 167);
this->lGlobalFile->Name = L"lGlobalFile";
this->lGlobalFile->Size = System::Drawing::Size(53, 13);
this->lGlobalFile->TabIndex = 2;
this->lGlobalFile->Text = L"Global file";
//
// lFoderOfTheFiles
//
this->lFoderOfTheFiles->AutoSize = true;
this->lFoderOfTheFiles->Location = System::Drawing::Point(64, 101);
this->lFoderOfTheFiles->Name = L"lFoderOfTheFiles";
this->lFoderOfTheFiles->Size = System::Drawing::Size(140, 13);
this->lFoderOfTheFiles->TabIndex = 1;
this->lFoderOfTheFiles->Text = L"Folder of the acquisition files";
//
// lPointsCoordinatesFile
//
this->lPointsCoordinatesFile->AutoSize = true;
this->lPointsCoordinatesFile->Location = System::Drawing::Point(64, 41);
this->lPointsCoordinatesFile->Name = L"lPointsCoordinatesFile";
this->lPointsCoordinatesFile->Size = System::Drawing::Size(110, 13);
this->lPointsCoordinatesFile->TabIndex = 0;
this->lPointsCoordinatesFile->Text = L"Points coordinates file";
//
// tsImageReconstruction
//
this->tsImageReconstruction->BackColor = System::Drawing::Color::LightBlue;
this->tsImageReconstruction->Controls->Add(this->Bevell);
this->tsImageReconstruction->Controls->Add(this->eMasseSurCharge);
this->tsImageReconstruction->Controls->Add(this->lMasseSurCharge);
this->tsImageReconstruction->Controls->Add(this->bColors);
this->tsImageReconstruction->Location = System::Drawing::Point(4, 22);
this->tsImageReconstruction->Name = L"tsImageReconstruction";
this->tsImageReconstruction->Padding = System::Windows::Forms::Padding(3);
this->tsImageReconstruction->Size = System::Drawing::Size(620, 426);
this->tsImageReconstruction->TabIndex = 6;
this->tsImageReconstruction->Text = L"Image reconstruction";
this->tsImageReconstruction->UseVisualStyleBackColor = true;
//
// Bevell
//
this->Bevell->Controls->Add(this->rbIntensiteMax);
this->Bevell->Controls->Add(this->rbAire);
this->Bevell->Location = System::Drawing::Point(38, 184);
this->Bevell->Name = L"Bevell";
this->Bevell->Size = System::Drawing::Size(519, 143);
this->Bevell->TabIndex = 3;
this->Bevell->TabStop = false;
this->Bevell->Text = L"Image reconstruction method :";
//
// rbIntensiteMax
//
this->rbIntensiteMax->AutoSize = true;
this->rbIntensiteMax->Location = System::Drawing::Point(78, 82);

```

```

this->rbIntensiteMax->Name = L"rbIntensiteMax";
this->rbIntensiteMax->Size = System::Drawing::Size(206, 17);
this->rbIntensiteMax->TabIndex = 2;
this->rbIntensiteMax->TabStop = true;
this->rbIntensiteMax->Text = L"with the maximum intensity of the peak";
this->rbIntensiteMax->UseVisualStyleBackColor = true;
//
// rbAire
//
this->rbAire->AutoSize = true;
this->rbAire->Location = System::Drawing::Point(78, 34);
this->rbAire->Name = L"rbAire";
this->rbAire->Size = System::Drawing::Size(143, 17);
this->rbAire->TabIndex = 1;
this->rbAire->TabStop = true;
this->rbAire->Text = L"with the area of the peak";
this->rbAire->UseVisualStyleBackColor = true;
//
// eMasseSurCharge
//
this->eMasseSurCharge->Location = System::Drawing::Point(239, 78);
this->eMasseSurCharge->Name = L"eMasseSurCharge";
this->eMasseSurCharge->Size = System::Drawing::Size(151, 20);
this->eMasseSurCharge->TabIndex = 2;
//
// lMasseSurCharge
//
this->lMasseSurCharge->AutoSize = true;
this->lMasseSurCharge->Location = System::Drawing::Point(195, 78);
this->lMasseSurCharge->Name = L"lMasseSurCharge";
this->lMasseSurCharge->Size = System::Drawing::Size(25, 13);
this->lMasseSurCharge->TabIndex = 1;
this->lMasseSurCharge->Text = L"m/z";
//
// bColors
//
this->bColors->Location = System::Drawing::Point(198, 117);
this->bColors->Name = L"bColors";
this->bColors->Size = System::Drawing::Size(192, 23);
this->bColors->TabIndex = 0;
this->bColors->Text = L"Choose the colors";
this->bColors->UseVisualStyleBackColor = true;
//
// pictureBox1
//
this->pictureBox1->BackgroundImage =
(cli::safe_cast<System::Drawing::Image^ >(resources-
>GetObject(L"pictureBox1.BackgroundImage")));
this->pictureBox1->BackgroundImageLayout =
System::Windows::Forms::ImageLayout::Stretch;
this->pictureBox1->Location = System::Drawing::Point(21, 12);
this->pictureBox1->Name = L"pictureBox1";
this->pictureBox1->Size = System::Drawing::Size(96, 57);
this->pictureBox1->TabIndex = 1;
this->pictureBox1->TabStop = false;
//
// pictureBox2
//
this->pictureBox2->BackgroundImage =
(cli::safe_cast<System::Drawing::Image^ >(resources-
>GetObject(L"pictureBox2.BackgroundImage")));
this->pictureBox2->Location = System::Drawing::Point(26, 75);
this->pictureBox2->Name = L"pictureBox2";
this->pictureBox2->Size = System::Drawing::Size(87, 73);
this->pictureBox2->TabIndex = 2;
this->pictureBox2->TabStop = false;
//
// lFilesCreated
//
this->lFilesCreated->AutoSize = true;

```

```

        this->lFilesCreated->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 8.25F, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(0)));
        this->lFilesCreated->Location = System::Drawing::Point(12, 151);
        this->lFilesCreated->Name = L"lFilesCreated";
        this->lFilesCreated->Size = System::Drawing::Size(80, 13);
        this->lFilesCreated->TabIndex = 3;
        this->lFilesCreated->Text = L"Files created";
        //
        // lRed
        //
        this->lRed->Location = System::Drawing::Point(23, 164);
        this->lRed->Name = L"lRed";
        this->lRed->Size = System::Drawing::Size(85, 37);
        this->lRed->TabIndex = 4;
        this->lRed->Text = L"- files not created are in red";
        //
        // lGreen
        //
        this->lGreen->Location = System::Drawing::Point(23, 201);
        this->lGreen->Name = L"lGreen";
        this->lGreen->Size = System::Drawing::Size(85, 37);
        this->lGreen->TabIndex = 5;
        this->lGreen->Text = L"- files created are in green";
        //
        // groupBox1
        //
        this->groupBox1->Controls->Add(this->lAF);
        this->groupBox1->Controls->Add(this->lPCF);
        this->groupBox1->Controls->Add(this->lISF);
        this->groupBox1->Controls->Add(this->cbFichierSEQ);
        this->groupBox1->Controls->Add(this->cbFichierPLT);
        this->groupBox1->Controls->Add(this->cbFichierBIC);
        this->groupBox1->Location = System::Drawing::Point(12, 229);
        this->groupBox1->Name = L"groupBox1";
        this->groupBox1->Size = System::Drawing::Size(111, 235);
        this->groupBox1->TabIndex = 6;
        this->groupBox1->TabStop = false;
        //
        // lAF
        //
        this->lAF->ForeColor = System::Drawing::Color::Red;
        this->lAF->Location = System::Drawing::Point(30, 184);
        this->lAF->Name = L"lAF";
        this->lAF->Size = System::Drawing::Size(66, 45);
        this->lAF->TabIndex = 13;
        this->lAF->Text = L"Acquisition File";
        this->lAF->TextAlign = System::Drawing::ContentAlignment::MiddleCenter;
        //
        // lPCF
        //
        this->lPCF->ForeColor = System::Drawing::Color::Red;
        this->lPCF->Location = System::Drawing::Point(30, 105);
        this->lPCF->Name = L"lPCF";
        this->lPCF->Size = System::Drawing::Size(66, 45);
        this->lPCF->TabIndex = 12;
        this->lPCF->Text = L"Points Coordinates File";
        this->lPCF->TextAlign = System::Drawing::ContentAlignment::MiddleCenter;
        //
        // lISF
        //
        this->lISF->ForeColor = System::Drawing::Color::Red;
        this->lISF->Location = System::Drawing::Point(30, 39);
        this->lISF->Name = L"lISF";
        this->lISF->Size = System::Drawing::Size(66, 32);
        this->lISF->TabIndex = 11;
        this->lISF->Text = L" Instrument Setting File";
        this->lISF->TextAlign = System::Drawing::ContentAlignment::MiddleCenter;
        //
        // cbFichierSEQ
        //

```

```

this->cbFichierSEQ->AutoSize = true;
this->cbFichierSEQ->ForeColor = System::Drawing::Color::Red;
this->cbFichierSEQ->Location = System::Drawing::Point(14, 164);
this->cbFichierSEQ->Name = L"cbFichierSEQ";
this->cbFichierSEQ->Size = System::Drawing::Size(64, 17);
this->cbFichierSEQ->TabIndex = 10;
this->cbFichierSEQ->Text = L"SEQ file";
this->cbFichierSEQ->UseVisualStyleBackColor = true;
//
// cbFichierPLT
//
this->cbFichierPLT->AutoSize = true;
this->cbFichierPLT->ForeColor = System::Drawing::Color::Red;
this->cbFichierPLT->Location = System::Drawing::Point(14, 85);
this->cbFichierPLT->Name = L"cbFichierPLT";
this->cbFichierPLT->Size = System::Drawing::Size(62, 17);
this->cbFichierPLT->TabIndex = 9;
this->cbFichierPLT->Text = L"PLT file";
this->cbFichierPLT->UseVisualStyleBackColor = true;
//
// cbFichierBIC
//
this->cbFichierBIC->AutoSize = true;
this->cbFichierBIC->ForeColor = System::Drawing::Color::Red;
this->cbFichierBIC->Location = System::Drawing::Point(14, 19);
this->cbFichierBIC->Name = L"cbFichierBIC";
this->cbFichierBIC->Size = System::Drawing::Size(59, 17);
this->cbFichierBIC->TabIndex = 8;
this->cbFichierBIC->Text = L"BIC file";
this->cbFichierBIC->UseVisualStyleBackColor = true;
//
// cClearAll
//
this->cClearAll->Location = System::Drawing::Point(33, 470);
this->cClearAll->Name = L"cClearAll";
this->cClearAll->Size = System::Drawing::Size(75, 23);
this->cClearAll->TabIndex = 7;
this->cClearAll->Text = L"Clear All";
this->cClearAll->UseVisualStyleBackColor = true;
this->cClearAll->Click += gcnew System::EventHandler(this,
&Form1::cClearAll_Click);
//
// bClose
//
this->bClose->Location = System::Drawing::Point(33, 499);
this->bClose->Name = L"bClose";
this->bClose->Size = System::Drawing::Size(75, 23);
this->bClose->TabIndex = 8;
this->bClose->Text = L"Close";
this->bClose->UseVisualStyleBackColor = true;
this->bClose->Click += gcnew System::EventHandler(this,
&Form1::bClose_Click);
//
// pbLogiciel
//
this->pbLogiciel->Location = System::Drawing::Point(133, 482);
this->pbLogiciel->Name = L"pbLogiciel";
this->pbLogiciel->Size = System::Drawing::Size(624, 23);
this->pbLogiciel->TabIndex = 9;
//
// errorProvider1
//
this->errorProvider1->ContainerControl = this;
//
// opSequenceType
//
this->opSequenceType->Filter = L"Points Coordinates File (*.plt)|*.plt";
//
// sdGlobalFile
//
this->sdGlobalFile->DefaultExt = L".txt";

```

```

//
// opDataFile
//
this->opDataFile->CheckFileExists = false;
this->opDataFile->Filter = L"Data file (*.dat)|*.dat";
//
// opInstrumentSettingFile
//
this->opInstrumentSettingFile->Filter = L"Instrument setting file
(*.bic)|*.bic";
//
// opPhoto
//
this->opPhoto->Filter = L"Picture files (*.bmp)|*.bmp|Photo files
(*.jpg)|*.jpg";
//
// odPointsCoordinatesFile
//
this->odPointsCoordinatesFile->Filter = L"Points Coordinates File (
*.plt)|*.plt";
//
// sdBIC
//
this->sdBIC->DefaultExt = L".bic";
this->sdBIC->Filter = L"Instrument setting file (*.bic)|*.bic";
//
// sdPLT
//
this->sdPLT->DefaultExt = L".plt";
this->sdPLT->Filter = L"Points Coordinates File (*.plt)|*.plt";
//
// sdSEQ
//
this->sdSEQ->DefaultExt = L".seq";
this->sdSEQ->Filter = L"Acquisition File (*.seq)|*.seq";
//
// Form1
//
this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);
this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;
this->BackColor = System::Drawing::Color::LightBlue;
this->ClientSize = System::Drawing::Size(801, 533);
this->Controls->Add(this->pbLogiciel);
this->Controls->Add(this->bClose);
this->Controls->Add(this->cClearAll);
this->Controls->Add(this->groupBox1);
this->Controls->Add(this->lGreen);
this->Controls->Add(this->lRed);
this->Controls->Add(this->lFilesCreated);
this->Controls->Add(this->pictureBox2);
this->Controls->Add(this->pictureBox1);
this->Controls->Add(this->pcLogiciel);
this->Icon = (cli::safe_cast<System::Drawing::Icon^ >(resources-
>GetObject(L"$this.Icon"));
this->Name = L"Form1";
this->Text = L"Imaging";
this->Load += gcnew System::EventHandler(this, &Form1::Form1_Load);
this->pcLogiciel->ResumeLayout(false);
this->tsWelcome->ResumeLayout(false);
(cli::safe_cast<System::ComponentModel::ISupportInitialize^ >(this-
>pictureBox4))->EndInit();
(cli::safe_cast<System::ComponentModel::ISupportInitialize^ >(this-
>pictureBox3))->EndInit();
this->tsBIC->ResumeLayout(false);
this->tsBIC->PerformLayout();
(cli::safe_cast<System::ComponentModel::ISupportInitialize^ >(this-
>tbLaserIntensity))->EndInit();
this->tsPLT->ResumeLayout(false);
this->groupBox4->ResumeLayout(false);
this->groupBox4->PerformLayout();
this->groupBox5->ResumeLayout(false);

```

```

                this->groupBox5->PerformLayout();
                this->groupBox3->ResumeLayout(false);
                this->groupBox3->PerformLayout();
                this->Coordinates->ResumeLayout(false);
                this->Coordinates->PerformLayout();
                this->tsPLTbis->ResumeLayout(false);
                this->tsPLTbis->PerformLayout();
                (cli::safe_cast<System::ComponentModel::ISupportInitialize^ >(this-
>pictureBoxCircle1))->EndInit();
                (cli::safe_cast<System::ComponentModel::ISupportInitialize^ >(this-
>iPhoto))->EndInit();
                this->groupBox2->ResumeLayout(false);
                this->groupBox2->PerformLayout();
                this->tsSEQ->ResumeLayout(false);
                this->bFurther->ResumeLayout(false);
                this->bFurther->PerformLayout();
                this->bMacros->ResumeLayout(false);
                this->bMacros->PerformLayout();
                this->bFiles->ResumeLayout(false);
                this->bFiles->PerformLayout();
                this->tsConvert->ResumeLayout(false);
                this->tsConvert->PerformLayout();
                this->tsImageReconstruction->ResumeLayout(false);
                this->tsImageReconstruction->PerformLayout();
                this->Bevell->ResumeLayout(false);
                this->Bevell->PerformLayout();
                (cli::safe_cast<System::ComponentModel::ISupportInitialize^ >(this-
>pictureBox1))->EndInit();
                (cli::safe_cast<System::ComponentModel::ISupportInitialize^ >(this-
>pictureBox2))->EndInit();
                this->groupBox1->ResumeLayout(false);
                this->groupBox1->PerformLayout();
                (cli::safe_cast<System::ComponentModel::ISupportInitialize^ >(this-
>errorProvider1))->EndInit();
                this->ResumeLayout(false);
                this->PerformLayout();
            }
#pragma endregion

        private: System::Void Form1_Load(System::Object^ sender, System::EventArgs^ e) {
            this->tsWelcome->BackColor=System::Drawing::Color::LightBlue;
            this->tsBIC->BackColor=System::Drawing::Color::LightBlue;
            this->tsConvert->BackColor=System::Drawing::Color::LightBlue;
            this->tsImageReconstruction-
>BackColor=System::Drawing::Color::LightBlue;
            this->tsPLT->BackColor=System::Drawing::Color::LightBlue;
            this->tsPLTbis->BackColor=System::Drawing::Color::LightBlue;
            this->tsSEQ->BackColor=System::Drawing::Color::LightBlue;
            this->InitialiseBicValue ();

            NomFichier = new FichierBIC;
            NomFichierPLT = new FichierPLT;
            point_dep = new Coordonnees;
            point_arr = new Coordonnees;
            NomFichierSEQ = new FichierSEQ;
            bHaveMouse = false;
            bHaveMouseCircle1=false;
        }

        private: System::Void bDefaultValues_Click(System::Object^ sender, System::EventArgs^ e)
        {
            this->InitialiseBicValue ();
        }

        private: void InitialiseBicValue()
        {
            this->edLaserIntensity->Text = "1450";
        }

```

```

        this->tbLaserIntensity->Value=1450;
        this->edAcceleratingVoltage->Text = "20000";
        this->edGridVoltage->Text = "94";
        this->edGuideWireVoltage0->Text = " 0.05";
        this->edDelayTime->Text = "200";
        this->edTotalLaserShots->Text = "50";
        this->edAcqMassMin->Text = "500";
        this->edAcqMassMax->Text = "6000";
        this->cbLowMassGate->Checked = true;
        this->edLowMassGateMass->Text = "500";
    }

private: System::Void bSaveFileBIC_Click(System::Object^ sender, System::EventArgs^ e) {
    if ((this->edLaserIntensity->Text != "")
        & (this->edAcceleratingVoltage->Text != "")
        & (this->edGridVoltage->Text != "")
        & (this->edGuideWireVoltage0->Text != "")
        & (this->edDelayTime->Text != "")
        & (this->edTotalLaserShots->Text != "")
        & (this->edAcqMassMin->Text != "")
        & (this->edAcqMassMax->Text != "")
        & (this->edLowMassGateMass->Text != ""))
    {
        this->sdBIC->ShowDialog ();
        if (this->sdBIC->FileName != "") {
            NomFichier->Sauvegarder(StringTOSTring(sdBIC-
>FileName));
            this->cbFichierBIC->Checked = true;
            this->cbFichierBIC-
>ForeColor=System::Drawing::Color::Green ;
            System::Drawing::Color::Green;
            this->lISF->ForeColor =
            this->eInstrumentSettingFile->Text = this->sdBIC-
>FileName;
        } // ferme le if (this->sdBIC->FileName != "")
    } // ferme le if ((this->edLaserIntensity->Text == "") .....
    else
    {
        MessageBox::Show ("There is empty lines"
, "error", MessageBoxButtons::OK, MessageBoxIcon::Warning );
    }
};

private: string StringTOSTring(System::String^ MyString)
{
    char * chaineChar = static_cast<char
*>(System::Runtime::InteropServices::Marshal::StringToHGlobalAnsi(MyString).ToPointer());

    /*pin_ptr<const wchar_t> wch = PtrToStringChars(MyString);
    int taille = (MyString->Length+1) * 2;
    chaineChar = new char[taille];
    int t = WideCharToMultiByte(CP_ACP, 0, wch, taille, NULL, 0, NULL,
NULL);
    WideCharToMultiByte(CP_ACP, 0, wch, taille, chaineChar, t, NULL,
NULL); */

    std::string chaineSTL = chaineChar;
    return chaineSTL;
}

#pragma endregion

private: System::Void bClearBIC_Click(System::Object^ sender, System::EventArgs^ e) {
    this->edLaserIntensity->Clear();
    this->edAcceleratingVoltage->Clear();
}

```

```

        this->edGridVoltage->Clear();
        this->edGuideWireVoltage0->Clear();
        this->edDelayTime->Clear();
        this->edTotalLaserShots->Clear();
        this->edAcqMassMin->Clear();
        this->edAcqMassMax->Clear();
        this->edLowMassGateMass->Clear();
    }

e) {
    private: System::Void tbLaserIntensity_Scroll(System::Object^ sender, System::EventArgs^
        this->edLaserIntensity->Text=this->tbLaserIntensity->Value+" " ;
    }

private: System::Void cClearAll_Click(System::Object^ sender, System::EventArgs^ e) {
    // valeurs de BIC
    this->InitialiseBicValue();
    // valeurs de PLT
    this->eXDep->Clear();
    this->eXArr->Clear();
    this->eYDep->Clear();
    this->eYArr->Clear();
    this->ePasH->Clear();
    this->ePasV->Clear();
    this->eNbPtsH->Clear();
    this->eNbPtsV->Clear();
    // valeurs de SEQ
    this->eNbLignes->Clear();
    this->eDataFile->Clear();
    this->eInstrumentSettingFile->Clear();
    this->eSequenceType->Clear();
    this->eSampleDescription->Clear();
    this->eComments->Clear();
    this->cbPreMacro->Text = "";
    this->cbPostMacro->Text = "";
    // retour a la page d'accueil
    this->pcLogiciel->SelectedIndex=0;
    // les fichiers repassent en rouge, non cochés
    this->cbFichierSEQ->Checked = false;
    this->cbFichierSEQ->ForeColor = System::Drawing::Color::Red ;
    this->lAF->ForeColor = System::Drawing::Color::Red ;
    this->cbFichierPLT->Checked = false;
    this->cbFichierPLT->ForeColor = System::Drawing::Color::Red ;
    this->lPCF->ForeColor = System::Drawing::Color::Red ;
    this->cbFichierBIC->Checked = false;
    this->cbFichierBIC->ForeColor = System::Drawing::Color::Red ;
    this->lISF->ForeColor = System::Drawing::Color::Red ;
}

private: System::Void bClose_Click(System::Object^ sender, System::EventArgs^ e) {
    delete NomFichier;
    delete NomFichierPLT;
    delete point_dep;
    delete point_arr;
    delete NomFichierSEQ;
    Close();
}

private: System::Void bStep_Click(System::Object^ sender, System::EventArgs^ e) {
    try {
        if ( (Convert::ToDouble(this->eXArr->Text) <
Convert::ToDouble(this->eXDep->Text)) | (Convert::ToDouble(this->eYArr->Text) >
Convert::ToDouble(this->eYDep->Text)) ) {
            MessageBox::Show ("The starting point should be in
the left and high corner and the ending point should be in the right and low corner"
,"error",MessageBoxButtons::OK, MessageBoxIcon::Error );
            if (Convert::ToDouble(this->eXArr->Text) <
Convert::ToDouble(this->eXDep->Text)){
                // this->eXDep->Text = "";
                this->eXArr->Text = "";
                // this->eXArr->Font->Color = clRed;
            }
        }
    }
}

```

```

    }
    if (Convert::ToDouble(this->eYArr->Text) >
Convert::ToDouble(this->eYDep->Text)){
        // this->eYDep->Text = "";
        this->eYArr->Text = "";
        // this->eYArr->Font->Color = clRed;
    }
    if ( this->ePasH->Text == "" ) {
        MessageBox::Show ( "There is no value for horizontal
step" , "error",MessageBoxButtons::OK, MessageBoxIcon::Warning );
    }
    point_dep->SetX(Convert::ToDouble(this->eXDep->Text));
    point_dep->SetY(Convert::ToDouble(this->eYDep->Text));
    NomFichierPLT->SetDep(point_dep);
    point_arr->SetX(Convert::ToDouble(this->eXArr->Text));
    point_arr->SetY(Convert::ToDouble(this->eYArr->Text));
    NomFichierPLT->SetArr(point_arr);

    if (this->ePasV->Text == ""){
        this->ePasV->Text = this->ePasH->Text;
    }

    NomFichierPLT->SetPasH(Convert::ToDouble(this->ePasH-
>Text));
    NomFichierPLT->SetPasV(Convert::ToDouble(this->ePasV-
>Text));

    // quand une valeur est entrée pour le pas, la valeur
correspondante s'affiche pour le nombre de points
    if (this->ePasH->Modified == true){
        int temp=0;
        temp= NomFichierPLT->CalculNbPtsLigne(point_arr,
point_dep, NomFichierPLT->GetPasH());
        int temp2=0;
        temp2= NomFichierPLT->CalculNbPtsColonne(point_arr,
point_dep, NomFichierPLT->GetPasV());
        this->eNbPtsH->Text = temp+"";
        this->eNbPtsV->Text = temp2+"";
    }
    if (this->ePasV->Modified == true){
        int temp3 =0;
        temp3= NomFichierPLT->CalculNbPtsColonne(point_arr,
point_dep, NomFichierPLT->GetPasV());
        this->eNbPtsV->Text = temp3+"";
    }
    }
    catch(Exception ^ex) {
        MessageBox::Show ( "Informations are missing"
, "error",MessageBoxButtons::OK, MessageBoxIcon::Warning );
    }
}

private: System::Void bNumberPoints_Click(System::Object^ sender, System::EventArgs^ e)
{
    try {
        if ( (Convert::ToDouble(this->eXArr->Text) <
Convert::ToDouble(this->eXDep->Text)) | (Convert::ToDouble(this->eYArr->Text) >
Convert::ToDouble(this->eYDep->Text)) ) {
            MessageBox::Show ( "The starting point should be in
the left and high corner and the ending point should be in the right and low corner"
, "error",MessageBoxButtons::OK, MessageBoxIcon::Warning);
            if (Convert::ToDouble(this->eXArr->Text) <
Convert::ToDouble(this->eXDep->Text)){
                // this->eXDep->Text = "";
                this->eXArr->Text = "";
            }
            if (Convert::ToDouble(this->eYArr->Text) >
Convert::ToDouble(this->eYDep->Text)){
                // this->eYDep->Text = "";
                this->eYArr->Text = "";
            }
        }
    }
}

```



```

        this->eYArr->Text = "";
    }
    this->eNbPtsH->Text = "";
    this->eNbPtsV->Text = "";
    this->ePasH->Text = "";
    this->ePasV->Text = "";
}

// vérification de la concordance entre les points
de départ et d'arrivée,
// les pas et les nombres de points par ligne et par
colonne

    if ( ( (Convert::ToInt32(this->eNbPtsH->Text) -1) !=
Math::Round ( (Convert::ToDouble (this->eXArr->Text) - Convert::ToDouble(this->eXDep->Text))
/ Convert::ToDouble(this->ePasH->Text)) ) |
( (Convert::ToInt32(this->eNbPtsV->Text) -1)
Math::Round ( (Convert::ToDouble(this->eYDep->Text) - Convert::ToDouble(this->eYArr->Text))
/ Convert::ToDouble(this->ePasV->Text) ) ) )
{
    MessageBox::Show("The values of the
step and the number of points doesn't fit with the starting and the ending point."
,"error",MessageBoxButtons::OK, MessageBoxIcon::Warning );
    if ( (Convert::ToInt32(this->eNbPtsH->Text) -1) !=
Math::Round (
(Convert::ToDouble(this->eXArr->Text) - Convert::ToDouble(this->eXDep->Text))
/ Convert::ToDouble(this->ePasH->Text) ) ) {
        this->eNbPtsH->Text =
"";
    }
    if ( (Convert::ToInt32(this->eNbPtsV->Text) -1) !=
Math::Round (
(Convert::ToDouble(this->eYDep->Text) - Convert::ToDouble(this->eYArr->Text))
/ Convert::ToDouble(this->ePasV->Text) ) ) {
        this->eNbPtsV->Text =
"";
    }
}

// if ( ( (Convert::ToDouble(this->eXArr->Text)) !=
// (Convert::ToDouble(this->eXDep->Text) +
// (Convert::ToInt32(this->eNbPtsH->Text) *
Convert::ToDouble(this->ePasH->Text) )) ) &
// ( (Convert::ToDouble(this->eYDep->Text)) !=
// (Convert::ToDouble(this->eYArr->Text) +
// (Convert::ToInt32(this->eNbPtsV->Text) *
Convert::ToDouble(this->ePasV->Text) )) ) ) {
    MessageBox(this->Handle,"The
values of the step and the number of points doesn't fit with the starting and the ending point."
,"error",MB_ICONWARNING);
    // if ( (Convert::ToDouble(this->eXArr->Text)) !=
// (Convert::ToDouble(this->eXDep->Text) +
// (Convert::ToInt32(this->eNbPtsH->Text) * Convert::ToDouble(this->ePasH->Text) )) ) {
//         this->eNbPtsH->Text =
// "";
//     }
    // if ( (Convert::ToDouble(this->eYDep->Text)) !=
// (Convert::ToDouble(this->eYArr->Text) +

```

```

// (Convert::ToInt32(this-
>eNbPtsV->Text) * Convert::ToDouble(this->ePasV->Text) ) ) {
// this->eNbPtsV-
>Text = "";
// }
// }

// récupération des valeurs

NomFichierPLT->SetAbsDep(Convert::ToDouble(this-
>eXDep->Text));
NomFichierPLT->SetOrdDep(Convert::ToDouble(this-
>eYDep->Text));

NomFichierPLT->SetAbsArr(Convert::ToDouble(this-
>eXArr->Text));
NomFichierPLT->SetOrdArr(Convert::ToDouble(this-
>eYArr->Text));

NomFichierPLT->SetPasH(Convert::ToDouble(this-
>ePasH->Text));
NomFichierPLT->SetPasV(Convert::ToDouble(this-
>ePasV->Text));

NomFichierPLT->SetNbPtsH(Convert::ToInt32(this-
>eNbPtsH->Text));
NomFichierPLT->SetNbPtsV(Convert::ToInt32(this-
>eNbPtsV->Text));

sdPLT->ShowDialog();
if (this->sdPLT->FileName != ""){
    NomFichierPLT-
    this->cbFichierPLT->Checked = true;
    this->cbFichierPLT->ForeColor=
    this->lPCF->ForeColor=
    this->eSequenceType->Text = this->sdPLT-
    this->ePointsCoordinatesFile->Text = this-

    int lines;
    lines=Filenbrlignes(StringTOstring(this-
>eSequenceType->Text));
    lines = lines -5;
    eNbLignes->Text = lines+"";
    this->pcLogiciel->SelectedTab = this->tsSEQ
;
} // ferme le if (this->sdPLT->FileName != "")
}
else {
    MessageBox::Show ("One or more value are missing"
, "error", MessageBoxButtons::OK, MessageBoxIcon::Warning );
}
}
catch (Exception ^ex){
    MessageBox::Show("Informations are missing"
, "error", MessageBoxButtons::OK, MessageBoxIcon::Warning );
}
}

int Filenbrlignes(string nom_fichier) {
    string ligne;
    int count=0;
    ifstream fi(nom_fichier.c_str());
    if(fi==0) return(0);
    while(!fi.eof()) {
        getline(fi, ligne);
    }
}

```

```

        count++;
    }
    fi.close();
    return(count);}
private: System::Void bPhoto_Click(System::Object^ sender, System::EventArgs^ e) {
    try {
        this->opPhoto->ShowDialog();
        if (this->opPhoto->FileName!=""){
            this->ePhoto->Text = this->opPhoto->FileName;
            this->iPhoto->Load(this->opPhoto->FileName);
            // this->iPhoto->Image->FromFile(this->opPhoto-
>FileName);
        }
        else {
            this->ePhoto->Clear();
        }
    }
    catch (Exception ^ex){
        MessageBox::Show ("You should give a picture
file.", "error", MessageBoxButtons::OK, MessageBoxIcon::Warning );
    }
}
private: System::Void bClearPLT_Click(System::Object^ sender, System::EventArgs^ e) {
    this->eXDep->Text="";
    this->eXArr->Text="";
    this->eYDep->Text="";
    this->eYArr->Text="";
    this->ePasH->Text="";
    this->ePasV->Text="";
    this->eNbPtsH->Text="";
    this->eNbPtsV->Text="";
}
private: System::Void bParcourirST_Click(System::Object^ sender, System::EventArgs^ e) {
    opSequenceType->ShowDialog ();
    if (opSequenceType->FileName!="")
    {
        eSequenceType->Text = opSequenceType->FileName;
    }
    else
    {
        eSequenceType->Clear();
    }

    int lines = 5;
    if ( System::IO::File::Exists(opSequenceType->FileName) )
    {
        lines=File::ReadLines(StringToString(opSequenceType-
>FileName));

        lines = lines -5;
        eNbLignes->Text = lines.ToString() ;
    }
}
private: System::Void bParcourirDF_Click(System::Object^ sender, System::EventArgs^ e) {
    opDataFile->ShowDialog ();
    if (opDataFile->FileName!="")
    {
        eDataFile->Text = opDataFile->FileName;
    }
    else
    {
        eDataFile->Clear();
    }
}
private: System::Void bParcourirISF_Click(System::Object^ sender, System::EventArgs^ e)
{

```

```

        opInstrumentSettingFile->ShowDialog();
        if (opInstrumentSettingFile->FileName!="")
            eInstrumentSettingFile->Text = opInstrumentSettingFile-
>FileName;
            else
                eInstrumentSettingFile->Clear();
        }

        private: System::Void bCreerFichierSEQ_Click(System::Object^ sender, System::EventArgs^
e) {
            try {
                if ((eNbLignes->Text != "") & (eDataFile->Text != "") &
(eInstrumentSettingFile->Text != "")) {
                    NomFichierSEQ-
>SetSamplePosition(Convert::ToInt32(eNbLignes->Text));
                    NomFichierSEQ->SetDataFile(StringTOSTring(eDataFile-
>Text));
                    NomFichierSEQ-
>SetInstrumentSettingFile(StringTOSTring(eInstrumentSettingFile->Text));

                    NomFichierSEQ-
>SetPreMacro(StringTOSTring(cbPreMacro->Text));
                    NomFichierSEQ-
>SetPostMacro(StringTOSTring(cbPostMacro->Text));
                    NomFichierSEQ-
>SetSampleDescription(StringTOSTring(eSampleDescription->Text));
                    NomFichierSEQ->SetComments(StringTOSTring(eComments-
>Text));

                    NomFichierSEQ->SetRun(1);

                    sdSEQ->ShowDialog ();
                    if (this->sdSEQ->FileName != ""){
                        NomFichierSEQ-
>Sauvegarder(StringTOSTring(this->sdSEQ->FileName));
                        this->cbFichierSEQ->Checked = true;
                        this->cbFichierSEQ->ForeColor=
System::Drawing::Color::Green;
                        this->lAF->ForeColor=
System::Drawing::Color::Green;
                    } // ferme le if (this->sdSEQ->FileName != "")
                } // ferme le premier if
            } else {
                MessageBox::Show ("You should give a file name for
each file asked.", "error", MessageBoxButtons::OK, MessageBoxIcon::Warning );
            }
        }
        catch (Exception ^ex){
            MessageBox::Show("Informations are missing"
, "error", MessageBoxButtons::OK, MessageBoxIcon::Warning );
        }
    }

    private: System::Void bClearSEQ_Click(System::Object^ sender, System::EventArgs^ e) {
        this->eNbLignes->Clear();
        this->eDataFile->Clear();
        this->eInstrumentSettingFile->Clear();
        this->eSequenceType->Clear();
        this->eSampleDescription->Clear();
        this->eComments->Clear();
        this->cbPreMacro->Text = "";
        this->cbPostMacro->Text = "";
    }

    private: System::Void bPointsCoordinatesFile_Click(System::Object^ sender,
System::EventArgs^ e) {
        this->odPointsCoordinatesFile->ShowDialog();
        if (this->odPointsCoordinatesFile->FileName!="")
        {
            this->ePointsCoordinatesFile->Text = this-
>odPointsCoordinatesFile->FileName;
        }
    }

```

```

        else
        {
            this->ePointsCoordinatesFile->Clear();
        }
    }
private: System::Void bFolderOfTheFiles_Click(System::Object^ sender, System::EventArgs^
e) {
    this->odFolderOfTheFiles->ShowDialog ();
    if (this->odFolderOfTheFiles->FileName!="")
    {
        this->eFolderOfTheFiles->Text = this->odFolderOfTheFiles-
>FileName;
    }
    else
    {
        this->eFolderOfTheFiles->Clear();
    }
}
private: System::Void bGlobalFile_Click(System::Object^ sender, System::EventArgs^ e) {
    this->sdGlobalFile->ShowDialog();
    if (this->sdGlobalFile->FileName!="")
    {
        this->eGlobalFile->Text = this->sdGlobalFile->FileName;
    }
    else
    {
        this->eGlobalFile->Clear();
    }
}
private: System::Void bConvert_Click(System::Object^ sender, System::EventArgs^ e) {
    if (this->sdGlobalFile->FileName != ""){
        try {
            System::IO::File::Copy(this->ePointsCoordinatesFile-
>Text,this->eGlobalFile->Text,true);
            MessageBox::Show ("The file has been
created","information",MessageBoxButtons::OK, MessageBoxIcon::Warning );
        }
        catch (Exception ^ex)
        {
            MessageBox::Show ("There is a probleme with the
file","error",MessageBoxButtons::OK, MessageBoxIcon::Warning );
        }
    }
}
private: void MyDrawReversibleRectangle(Point p1, Point p2,bool fixed)
{
    Rectangle rc;

    // Convert the points to screen coordinates.
    p1 = PointToScreen(p1);
    p2 = PointToScreen(p2);

    // Normalize the rectangle.
    if (p1.X < p2.X)
    {
        rc.X = p1.X;
        rc.Width = p2.X - p1.X;
    }
    else
    {
        rc.X = p2.X;
        rc.Width = p1.X - p2.X;
    }

    if (p1.Y < p2.Y)
    {
        rc.Y = p1.Y;
        rc.Height = p2.Y - p1.Y;
    }
    else
    {

```

```

        rc.Y = p2.Y;
        rc.Height = p1.Y - p2.Y;
    }

    // Draw the reversible frame.
    if (fixed==false)
    {
        ControlPaint::DrawReversibleFrame(rc, Color::Red, FrameStyle::Dashed );
    }
}

private: System::Void iPhoto_MouseDown(System::Object^ sender,
System::Windows::Forms::MouseEventArgs^ e) {
    // Make a note that you "have the mouse".
    bHaveMouse = true;
    this->iPhoto->Invalidate();
    // Store the "starting point" for this rubber-band rectangle.
    ptOriginal.X = e->X+this->iPhoto->Location.X+this->pcLogiciel->Location.X+this-
>pcLogiciel->Padding.X;
    ptOriginal.Y = e->Y+this->iPhoto->Location.Y+this->pcLogiciel->Location.Y+this-
>pcLogiciel->ItemSize.Height+this->pcLogiciel->Padding.Y;

    // Special value lets you know that no previous
    // rectangle must be erased.
    ptLast.X = -1;
    ptLast.Y = -1;
}

private: System::Void iPhoto_MouseUp(System::Object^ sender,
System::Windows::Forms::MouseEventArgs^ e) {
    // Set internal flag to know that you no longer "have the mouse."
    bHaveMouse = false;

    // If you have drawn previously, draw again in that spot
    // to remove the lines.
    if (ptLast.X != -1)
    {
        Point ptCurrent;
        ptCurrent.X = e->X+this->iPhoto->Location.X+this->pcLogiciel-
>Location.X+this->pcLogiciel->Padding.X;
        ptCurrent.Y = e->Y+this->iPhoto->Location.Y+this->pcLogiciel-
>Location.Y+this->pcLogiciel->ItemSize.Height+this->pcLogiciel->Padding.Y;
        MyDrawReversibleRectangle(ptOriginal, ptLast,true);
    }

    // Set flags to know that there is no "previous" line to reverse.
    ptLast.X = -1;
    ptLast.Y = -1;
    ptOriginal.X = -1;
    ptOriginal.Y = -1;
}

private: System::Void iPhoto_MouseMove(System::Object^ sender,
System::Windows::Forms::MouseEventArgs^ e) {
    Point ptCurrent;
    ptCurrent.X = e->X+this->iPhoto->Location.X+this->pcLogiciel->Location.X+this-
>pcLogiciel->Padding.X;
    ptCurrent.Y = e->Y+this->iPhoto->Location.Y+this->pcLogiciel->Location.Y+this-
>pcLogiciel->ItemSize.Height+this->pcLogiciel->Padding.Y;

    // If you "have the mouse," draw your lines.
    if (bHaveMouse)
    {
        // If you have drawn previously, draw again in
        // that spot to remove the lines.
        if (ptLast.X != -1)
            MyDrawReversibleRectangle(ptOriginal, ptLast,false);

        // Update last point.
        ptLast = ptCurrent;

        // Draw new lines.
        MyDrawReversibleRectangle(ptOriginal, ptCurrent,false);
    }
}

```

```
    }
private: System::Void pictureBoxCircle1_MouseDown(System::Object^ sender,
System::Windows::Forms::MouseEventArgs^ e) {
    bHaveMouseCircle1=true;
}
private: System::Void pictureBoxCircle1_MouseMove(System::Object^ sender,
System::Windows::Forms::MouseEventArgs^ e) {
    if (bHaveMouseCircle1)
    {
        this->pictureBoxCircle1->Location=e->Location;
    }
}
private: System::Void pictureBoxCircle1_MouseUp(System::Object^ sender,
System::Windows::Forms::MouseEventArgs^ e) {
    bHaveMouseCircle1=false;
}
};
}
```

## Applied.pro

pro WID\_BASE\_APPLIED\_event, Event

```
wTarget = (widget_info(Event.id,/NAME) eq 'TREE' ? $
  widget_info(Event.id, /tree_root) : event.id)
```

```
wWidget = Event.top
```

```
case wTarget of
```

```
Widget_Info(wWidget, FIND_BY_UNAME='WID_BUTTON_Accept_APPLIED'): begin
  if( Tag_Names(Event, /STRUCTURE_NAME) eq 'WIDGET_BUTTON' )then $
    OnButtonPressAccept_APPLIED, Event
  end
```

```
Widget_Info(wWidget, FIND_BY_UNAME='WID_BUTTON_ExplorerPLT_APPLIED'): begin
  if( Tag_Names(Event, /STRUCTURE_NAME) eq 'WIDGET_BUTTON' )then $
    OnButtonPressExplorerPLT_APPLIED, Event
  end
```

```
Widget_Info(wWidget, FIND_BY_UNAME='WID_BUTTON_ExplorerDATA_APPLIED'): begin
  if( Tag_Names(Event, /STRUCTURE_NAME) eq 'WIDGET_BUTTON' )then $
    OnButtonPressExplorerData_APPLIED, Event
  end
```

```
end
else:
endcase
```

```
end
```

pro WID\_BASE\_APPLIED, GROUP\_LEADER=wGroup, \_EXTRA=\_VWBExtra\_

```
Resolve_Routine, 'Applied_eventcb',/COMPILE_FULL_FILE ; Load event callback routines
```

```
WID_BASE_APPLIED = Widget_Base( GROUP_LEADER=wGroup, $
  UNAME='WID_BASE_APPLIED',XOFFSET=5,YOFFSET=5,SCR_XSIZE=630 $
  ,SCR_YSIZE=455,TAB_MODE=2,TITLE='IDL',SPACE=3,XPAD=3 $
  ,YPAD=3)
```

```
WID_BUTTON_Accept_APPLIED = Widget_Button(WID_BASE_APPLIED, $
  UNAME='WID_BUTTON_Accept_APPLIED',XOFFSET=209,YOFFSET=275 $
  ,SCR_XSIZE=124,SCR_YSIZE=38,TAB_MODE=2,/ALIGN_CENTER $
  ,VALUE='Accept')
```

```
WID_LABEL_TITLE_APPLIED = Widget_Label(WID_BASE_APPLIED, $
  UNAME='WID_LABEL_TITLE_APPLIED',XOFFSET=40,YOFFSET=25 $
  ,SCR_XSIZE=372,SCR_YSIZE=18,/ALIGN_LEFT,VALUE='APPLIED NEW'+ $
  'PROJECT')
```

```
WID_LABEL_NAME_APPLIED = Widget_Label(WID_BASE_APPLIED, $
  UNAME='WID_LABEL_NAME_APPLIED',XOFFSET=39,YOFFSET=73 $
  ,/ALIGN_LEFT,VALUE='Name')
```

```
WID_TEXT_PROJECT_NAME_APPLIED = Widget_Text(WID_BASE_APPLIED, $
  UNAME='WID_TEXT_PROJECT_NAME_APPLIED',XOFFSET=84,YOFFSET=70 $
  ,SCR_XSIZE=367,SCR_YSIZE=21,TAB_MODE=2,/EDITABLE,XSIZE=20 $
  ,YSIZE=1)
```

```
WID_LABEL_COMMENT_APPLIED = Widget_Label(WID_BASE_APPLIED, $
  UNAME='WID_LABEL_COMMENT_APPLIED',XOFFSET=29,YOFFSET=121 $
  ,/ALIGN_LEFT,VALUE='Comment')
```

```
WID_TEXT_PROJECT_COMMENT_APPLIED = Widget_Text(WID_BASE_APPLIED, $
  UNAME='WID_TEXT_PROJECT_COMMENT_APPLIED',XOFFSET=83 $
  ,YOFFSET=121,SCR_XSIZE=367,SCR_YSIZE=21,TAB_MODE=2 $
  ,/EDITABLE,XSIZE=20,YSIZE=1)
```

```
WID_TEXT_DATA_APPLIED = Widget_Text(WID_BASE_APPLIED, $
  UNAME='WID_TEXT_DATA_APPLIED',XOFFSET=83,YOFFSET=220 $
```

```

,SCR_XSIZE=367 ,SCR_YSIZE=21 ,TAB_MODE=2 ,/EDITABLE ,XSIZE=20 $
,YSIZE=1)

WID_LABEL_DATA_APPLIED = Widget_Label(WID_BASE_APPLIED, $
  UNAME='WID_LABEL_DATA_APPLIED' ,XOFFSET=25 ,YOFFSET=222 $
  ,SCR_XSIZE=45 ,SCR_YSIZE=18 ,/ALIGN_LEFT ,VALUE='Data files')

WID_TEXT_PLT_APPLIED = Widget_Text(WID_BASE_APPLIED, $
  UNAME='WID_TEXT_PLT_APPLIED' ,XOFFSET=84 ,YOFFSET=170 $
  ,SCR_XSIZE=367 ,SCR_YSIZE=21 ,TAB_MODE=2 ,/EDITABLE ,XSIZE=20 $
  ,YSIZE=1)

WID_LABEL_PLT_APPLIED = Widget_Label(WID_BASE_APPLIED, $
  UNAME='WID_LABEL_PLT_APPLIED' ,XOFFSET=30 ,YOFFSET=172 $
  ,SCR_XSIZE=38 ,SCR_YSIZE=18 ,/ALIGN_LEFT ,VALUE='PLT file')

WID_BUTTON_ExplorerPLT_APPLIED = Widget_Button(WID_BASE_APPLIED, $
  UNAME='WID_BUTTON_ExplorerPLT_APPLIED' ,XOFFSET=470 $
  ,YOFFSET=173 ,SCR_XSIZE=47 ,SCR_YSIZE=17 ,TAB_MODE=2 $
  ,/ALIGN_CENTER ,VALUE='Browse')

WID_BUTTON_ExplorerDATA_APPLIED = Widget_Button(WID_BASE_APPLIED, $
  UNAME='WID_BUTTON_ExplorerDATA_APPLIED' ,XOFFSET=470 $
  ,YOFFSET=222 ,SCR_XSIZE=47 ,SCR_YSIZE=18 ,TAB_MODE=2 $
  ,/ALIGN_CENTER ,VALUE='Browse')

WID_LABEL_INFO_APP = Widget_Label(WID_BASE_APPLIED, $
  UNAME='WID_LABEL_INFO_APP' ,XOFFSET=39 ,YOFFSET=341 $
  ,SCR_XSIZE=580 ,SCR_YSIZE=15 ,/ALIGN_LEFT ,VALUE='Please fill'+ $
  ' the text areas')

WID_LABEL_INFO_FILE_APP = Widget_Label(WID_BASE_APPLIED, $
  UNAME='WID_LABEL_INFO_FILE_APP' ,XOFFSET=41 ,YOFFSET=374 $
  ,SCR_XSIZE=575 ,SCR_YSIZE=15 ,/ALIGN_LEFT ,VALUE='No file is'+ $
  ' currently read.')

Widget_Control, /REALIZE, WID_BASE_APPLIED

XManager, 'WID_BASE_APPLIED', WID_BASE_APPLIED, /NO_BLOCK

end

pro Applied, GROUP_LEADER=wGroup, _EXTRA=_VWBExtra_
  WID_BASE_APPLIED, GROUP_LEADER=wGroup, _EXTRA=_VWBExtra_
end

Applied eventcb.pro

pro OnButtonPressAccept_APPLIED, Event

wname=widget_info(Event.top, FIND_BY_UNAME='WID_TEXT_PROJECT_NAME_APPLIED')
  WIDGET_CONTROL, wname, GET_VALUE=name

  wcomment=widget_info(Event.top, FIND_BY_UNAME='WID_TEXT_PROJECT_COMMENT_APPLIED')
  WIDGET_CONTROL, wcomment, GET_VALUE=comment

  wfileplt=widget_info(Event.top, FIND_BY_UNAME='WID_TEXT_PLT_APPLIED')
  WIDGET_CONTROL, wfileplt, GET_VALUE=fileplt

  wdirectorydata=widget_info(Event.top, FIND_BY_UNAME='WID_TEXT_DATA_APPLIED')
  WIDGET_CONTROL, wdirectorydata, GET_VALUE=directorydata

if (name EQ "") then begin
  erreur=DIALOG_MESSAGE(" Please give a name to the project " , /ERROR)
  return
endif

```

```

if (fileplt EQ "") then begin
erreur=DIALOG_MESSAGE(" Please give a plt file name " , /ERROR)
return
endif

if (directorydata EQ "") then begin
erreur=DIALOG_MESSAGE(" Please give the name of the directory where the data are " , /ERROR)
return
endif

; récupération des ID de la fenêtre et des labels pour voir la progression

wwidget=widget_info(Event.top,FIND_BY_UNAME='WID_BASE_APPLIED')
WIDGET_CONTROL, wwidget, /UPDATE

wlabelinfo=widget_info(Event.top, FIND_BY_UNAME='WID_LABEL_INFO_APP')

wlabelinfofile=widget_info(Event.top, FIND_BY_UNAME='WID_LABEL_INFO_FILE_APP')

AppliedProject, name, comment, fileplt, directorydata, wwidget, wlabelinfo, wlabelinfofile

end

pro OnButtonPressExplorerPLT_APPLIED, Event

filePLT=DIALOG_PICKFILE(TITLE=" Fichier .PLT: positions",FILTER="*.plt", PATH='F:\THESE\logiciel
imagerie\testpltdl180.plt')
wtext=widget_info(Event.top, FIND_BY_UNAME='WID_TEXT_PLT_APPLIED')
WIDGET_CONTROL, wtext, SET_VALUE=filePLT

end

pro OnButtonPressExplorerData_APPLIED, Event

sFolder = DIALOG_PICKFILE( /DIRECTORY, $
TITLE="Choose directory containing Applied Data.")
wtext=widget_info(Event.top, FIND_BY_UNAME='WID_TEXT_DATA_APPLIED')
WIDGET_CONTROL, wtext, SET_VALUE=sFolder
Result = FILE_SEARCH(sFolder+"\\*_**.txt")

end

pro Applied_eventcb
end

```

## **BDparam.pro**

```

pro WID_BASE_BD_PARAM_event, Event

wTarget = (widget_info(Event.id,/NAME) eq 'TREE' ? $
widget_info(Event.id, /tree_root) : event.id)

wWidget = Event.top

case wTarget of

Widget_Info(wWidget, FIND_BY_UNAME='WID_BUTTON_CHANGE'): begin
if( Tag_Names(Event, /STRUCTURE_NAME) eq 'WIDGET_BUTTON' )then $
use_parameters, Event
end
else:
endcase

end

pro WID_BASE_BD_PARAM, GROUP_LEADER=wGroup, _EXTRA=_VWBExtra_

Resolve_Routine, 'BDparam_eventcb',/COMPILE_FULL_FILE ; Load event callback routines

WID_BASE_BD_PARAM = Widget_Base( GROUP_LEADER=wGroup, $

```

```
UNAME='WID_BASE_BD_PARAM',XOFFSET=5,YOFFSET=5,SCR_XSIZE=517 $
,SCR_YSIZE=352,TAB_MODE=2,TITLE='Databank parameters' $
,SPACE=3,XPAD=3,YPAD=3)
```

```
WID_DROPLIST_DATASOURCE = Widget_Droplist(WID_BASE_BD_PARAM, $
UNAME='WID_DROPLIST_DATASOURCE',XOFFSET=68,YOFFSET=78 $
,TITLE='choose the data source :',VALUE=[ 'serveur', $
'imaging', 'pc jerome' ])
```

```
display_datasources, WID_DROPLIST_DATASOURCE, _EXTRA=_VWBExtra_
```

```
WID_LABEL_DATASOURCE = Widget_Label(WID_BASE_BD_PARAM, $
UNAME='WID_LABEL_DATASOURCE',XOFFSET=33,YOFFSET=39 $
,/ALIGN_LEFT,VALUE='DATASOURCE')
```

```
WID_LABEL_USER = Widget_Label(WID_BASE_BD_PARAM, $
UNAME='WID_LABEL_USER',XOFFSET=35,YOFFSET=121,/ALIGN_LEFT $
,VALUE='USER')
```

```
WID_LABEL_PASSWORD = Widget_Label(WID_BASE_BD_PARAM, $
UNAME='WID_LABEL_PASSWORD',XOFFSET=24,YOFFSET=197 $
,/ALIGN_LEFT,VALUE='PASSWORD')
```

```
WID_BUTTON_CHANGE = Widget_Button(WID_BASE_BD_PARAM, $
UNAME='WID_BUTTON_CHANGE',XOFFSET=312,YOFFSET=219 $
,/ALIGN_CENTER,VALUE='Use these parameters')
```

```
WID_TEXT_USER = Widget_Text(WID_BASE_BD_PARAM, $
UNAME='WID_TEXT_USER',XOFFSET=67,YOFFSET=151,/EDITABLE $
,VALUE=[ 'root', " ],XSIZE=20,YSIZE=1)
```

```
WID_TEXT_PASSWORD = Widget_Text(WID_BASE_BD_PARAM, $
UNAME='WID_TEXT_PASSWORD',XOFFSET=68,YOFFSET=235,/EDITABLE $
,VALUE=[ 'toor', " ],XSIZE=20,YSIZE=1)
```

```
Widget_Control, /REALIZE, WID_BASE_BD_PARAM
```

```
XManager, 'WID_BASE_BD_PARAM', WID_BASE_BD_PARAM, /NO_BLOCK
```

```
end
```

```
pro BDparam, GROUP_LEADER=wGroup, _EXTRA=_VWBExtra_
WID_BASE_BD_PARAM, GROUP_LEADER=wGroup, _EXTRA=_VWBExtra_
end
```

## **BDparam\_eventcb.pro**

```
pro BDparam_eventcb
end
```

```
pro use_parameters, Event
```

```
COMMON source, BDdatasource
COMMON parambd, BDuser, BDpassword
```

```
wdatasources=widget_info(Event.top, FIND_BY_UNAME='WID_DROPLIST_DATASOURCE')
WIDGET_CONTROL, wdatasources, GET_VALUE=datasources
iBDdatasource = widget_info(wdatasources, /DROPLIST_SELECT)
BDdatasource = datasources[iBDdatasource]
```

```
wusers=widget_info(Event.top, FIND_BY_UNAME='WID_TEXT_USER')
WIDGET_CONTROL, wusers, GET_VALUE=BDuser
```

```
wpasswords=widget_info(Event.top, FIND_BY_UNAME='WID_TEXT_PASSWORD')
WIDGET_CONTROL, wpasswords, GET_VALUE=BDpassword
```

```

widget_control, Event.top, /destroy

end

pro display_datasources, wWidget, _EXTRA=_VWBExtra_

COMMON source, BDdatasource
COMMON parambd, BDuser, BDpassword
COMMON BDSources, sources

WIDGET_CONTROL, wWidget, SET_VALUE=sources.DATASOURCE

end

```

## **Bruker.pro**

```

pro WID_BASE_BRUKER_event, Event

wTarget = (widget_info(Event.id,/NAME) eq 'TREE' ? $
  widget_info(Event.id, /tree_root) : event.id)

wWidget = Event.top

case wTarget of

Widget_Info(wWidget, FIND_BY_UNAME='WID_BUTTON_ExplorerDATA_BRUKER'): begin
  if (Tag_Names(Event, /STRUCTURE_NAME) eq 'WIDGET_BUTTON' )then $
    OnButtonPressExplorerData_BRUKER, Event
  end
Widget_Info(wWidget, FIND_BY_UNAME='WID_BUTTON_Accept'): begin
  if (Tag_Names(Event, /STRUCTURE_NAME) eq 'WIDGET_BUTTON' )then $
    OnButtonPressAccept_BRUKER, Event
  end
else:
endcase

end

pro WID_BASE_BRUKER, GROUP_LEADER=wGroup, _EXTRA=_VWBExtra_

Resolve_Routine, 'Bruker_eventcb',/COMPILE_FULL_FILE ; Load event callback routines

WID_BASE_BRUKER = Widget_Base( GROUP_LEADER=wGroup, $
  UNAME='WID_BASE_BRUKER',XOFFSET=5 ,YOFFSET=5 ,SCR_XSIZE=675 $
  ,SCR_YSIZE=424 ,TAB_MODE=2 ,TITLE='IDL' ,SPACE=3 ,XPAD=3 $
  ,YPAD=3)

WID_BUTTON_ExplorerDATA_BRUKER = Widget_Button(WID_BASE_BRUKER, $
  UNAME='WID_BUTTON_ExplorerDATA_BRUKER',XOFFSET=465 $
  ,YOFFSET=193 ,SCR_XSIZE=47 ,SCR_YSIZE=18 ,/ALIGN_CENTER $
  ,VALUE='Browse')

WID_LABEL_DATA_BRUKER = Widget_Label(WID_BASE_BRUKER, $
  UNAME='WID_LABEL_DATA_BRUKER',XOFFSET=24 ,YOFFSET=193 $
  ,SCR_XSIZE=45 ,SCR_YSIZE=18 ,/ALIGN_LEFT ,VALUE='Data files')

WID_TEXT_DATA_BRUKER = Widget_Text(WID_BASE_BRUKER, $
  UNAME='WID_TEXT_DATA_BRUKER',XOFFSET=83 ,YOFFSET=191 $
  ,SCR_XSIZE=367 ,SCR_YSIZE=21 ,/EDITABLE ,XSIZE=20 ,YSIZE=1)

WID_TEXT_PROJECT_COMMENT_BRUKER = Widget_Text(WID_BASE_BRUKER, $
  UNAME='WID_TEXT_PROJECT_COMMENT_BRUKER',XOFFSET=83 $
  ,YOFFSET=136 ,SCR_XSIZE=367 ,SCR_YSIZE=21 ,/EDITABLE ,XSIZE=20 $
  ,YSIZE=1)

WID_LABEL_COMMENT_BRUKER = Widget_Label(WID_BASE_BRUKER, $
  UNAME='WID_LABEL_COMMENT_BRUKER',XOFFSET=28 ,YOFFSET=137 $

```

```

./ALIGN_LEFT ,VALUE='Comment')

WID_TEXT_PROJECT_NAME_BRUKER = Widget_Text(WID_BASE_BRUKER, $
  UNAME='WID_TEXT_PROJECT_NAME_BRUKER' ,XOFFSET=82 ,YOFFSET=86 $
  ,SCR_XSIZE=367 ,SCR_YSIZE=21 ,TAB_MODE=2 ,/EDITABLE ,XSIZE=20 $
  ,YSIZE=1)

WID_LABEL_NAME_BRUKER = Widget_Label(WID_BASE_BRUKER, $
  UNAME='WID_LABEL_NAME_BRUKER' ,XOFFSET=37 ,YOFFSET=89 $
  ,/ALIGN_LEFT ,VALUE='Name')

WID_LABEL_TITLE_BRUKER = Widget_Label(WID_BASE_BRUKER, $
  UNAME='WID_LABEL_TITLE_BRUKER' ,XOFFSET=38 ,YOFFSET=38 $
  ,SCR_XSIZE=372 ,SCR_YSIZE=18 ,/ALIGN_LEFT ,VALUE='BRUKER NEW'+ $
  ' PROJECT')

WID_BUTTON_Accept = Widget_Button(WID_BASE_BRUKER, $
  UNAME='WID_BUTTON_Accept' ,XOFFSET=281 ,YOFFSET=249 $
  ,SCR_XSIZE=124 ,SCR_YSIZE=38 ,/ALIGN_CENTER ,VALUE='Accept')

WID_LABEL_INFO_BRUK = Widget_Label(WID_BASE_BRUKER, $
  UNAME='WID_LABEL_INFO_BRUK' ,XOFFSET=41 ,YOFFSET=307 $
  ,SCR_XSIZE=618 ,SCR_YSIZE=15 ,/ALIGN_LEFT ,VALUE='Please fill'+ $
  ' the text areas')

WID_LABEL_INFO_FILE_BRUK = Widget_Label(WID_BASE_BRUKER, $
  UNAME='WID_LABEL_INFO_FILE_BRUK' ,XOFFSET=38 ,YOFFSET=345 $
  ,SCR_XSIZE=625 ,SCR_YSIZE=15 ,/ALIGN_LEFT ,VALUE='No file is'+ $
  ' currently read.')

Widget_Control, /REALIZE, WID_BASE_BRUKER

XManager, 'WID_BASE_BRUKER', WID_BASE_BRUKER, /NO_BLOCK

end

pro Bruker, GROUP_LEADER=wGroup, _EXTRA=_VWBExtra_
  WID_BASE_BRUKER, GROUP_LEADER=wGroup, _EXTRA=_VWBExtra_
end

Bruker eventcb.pro

pro OnButtonPressExplorerData_BRUKER, Event
  sFolder = DIALOG_PICKFILE( /DIRECTORY, $
    TITLE="Choose directory containing Bruker Data.")
    wtext=widget_info(Event.top, FIND_BY_UNAME='WID_TEXT_DATA_BRUKER')
    WIDGET_CONTROL, wtext, SET_VALUE=sFolder
    Result = FILE_SEARCH(sFolder+"\*_**.txt')

end

pro OnButtonPressAccept_BRUKER, Event

  wname=widget_info(Event.top, FIND_BY_UNAME='WID_TEXT_PROJECT_NAME_BRUKER')
  WIDGET_CONTROL, wname, GET_VALUE=name

  wcomment=widget_info(Event.top, FIND_BY_UNAME='WID_TEXT_PROJECT_COMMENT_BRUKER')
  WIDGET_CONTROL, wcomment, GET_VALUE=comment

  wdirectorydata=widget_info(Event.top, FIND_BY_UNAME='WID_TEXT_DATA_BRUKER')
  WIDGET_CONTROL, wdirectorydata, GET_VALUE=directorydata

if (name EQ "") then begin
  erreur=DIALOG_MESSAGE(" Please give a name to the project " , /ERROR)
  return
endif

```

```

if (directorydata EQ "") then begin
erreur=DIALOG_MESSAGE(" Please give the name of the directory where the data are " , /ERROR)
return
endif

```

; récupération des ID de la fenêtre et des labels pour voir la progression

```

wwidget=widget_info(Event.top,FIND_BY_UNAME='WID_BASE_BRUKER')
WIDGET_CONTROL, wwidget, /UPDATE

```

```

wlabelinfo=widget_info(Event.top, FIND_BY_UNAME='WID_LABEL_INFO_BRUK')

```

```

wlabelinfofile=widget_info(Event.top, FIND_BY_UNAME='WID_LABEL_INFO_FILE_BRUK')

```

```

BrukerProject, name, comment, directorydata, wwidget, wlabelinfo, wlabelinfofile

```

```

end

```

```

pro Bruker_eventcb
end

```

## **ImagingSoftware.pro**

```

WID_BASE_IMAGING_event, Event

```

```

wTarget = (widget_info(Event.id,/NAME) eq 'TREE' ? $
widget_info(Event.id, /tree_root) : event.id)

```

```

wWidget = Event.top

```

```

case wTarget of

```

```

Widget_Info(wWidget, FIND_BY_UNAME='WID_BUTTON_CHOICE_APPLIED'): begin
if( Tag_Names(Event, /STRUCTURE_NAME) eq 'WIDGET_BUTTON' )then $
OnButtonAppliedPress, Event
end

```

```

Widget_Info(wWidget, FIND_BY_UNAME='WID_BUTTON_CHOICE_BRUKER'): begin
if( Tag_Names(Event, /STRUCTURE_NAME) eq 'WIDGET_BUTTON' )then $
OnButtonAppliedPress, Event
end

```

```

Widget_Info(wWidget, FIND_BY_UNAME='WID_BUTTON_CHOICE_ACCEPT'): begin
if( Tag_Names(Event, /STRUCTURE_NAME) eq 'WIDGET_BUTTON' )then $
OnButtonChoicePress, Event
end

```

```

Widget_Info(wWidget, FIND_BY_UNAME='WID_BUTTON_RECONSTRUCTION_ONLY'): begin
if( Tag_Names(Event, /STRUCTURE_NAME) eq 'WIDGET_BUTTON' )then $
see_projects, Event
end

```

```

Widget_Info(wWidget, FIND_BY_UNAME='WID_BUTTON_PARAM_BD'): begin
if( Tag_Names(Event, /STRUCTURE_NAME) eq 'WIDGET_BUTTON' )then $
change_bd_param, Event
end

```

```

else:
endcase

```

```

end

```

```

pro WID_BASE_IMAGING, GROUP_LEADER=wGroup, _EXTRA=_VWBExtra_

```

```

Resolve_Routine, 'ImagingSoftware_eventcb',/COMPILE_FULL_FILE ; Load event callback routines

```

```

WID_BASE_IMAGING = Widget_Base( GROUP_LEADER=wGroup, $
UNAME='WID_BASE_IMAGING' ,XOFFSET=5 ,YOFFSET=5 ,SCR_XSIZE=535 $
,SCR_YSIZE=348 ,TAB_MODE=1 ,TITLE='MALDI IMAGING TEAM SOFTWARE' $
,SPACE=3 ,XPAD=3 ,YPAD=3)

```

```

WID_LABEL_WELCOME = Widget_Label(WID_BASE_IMAGING, $
UNAME='WID_LABEL_WELCOME' ,XOFFSET=122 ,YOFFSET=24 ,ALIGN_LEFT $
,VALUE='WELCOME ON THE MALDI IMAGING TEAM SOFTWARE')

```

```

WID_LABEL_CHOICE = Widget_Label(WID_BASE_IMAGING, $
  UNAME='WID_LABEL_CHOICE',XOFFSET=27,YOFFSET=80,/ALIGN_LEFT $
  ,VALUE='To load data in the databank before the reconstruction,+ $
  ' choose one of the spectrometers :')

WID_BASE_CHOICE = Widget_Base(WID_BASE_IMAGING, $
  UNAME='WID_BASE_CHOICE',XOFFSET=152,YOFFSET=108 $
  ,SCR_XSIZE=208,SCR_YSIZE=101,TITLE='IDL',COLUMN=1 $
  ,/EXCLUSIVE)

WID_BUTTON_CHOICE_APPLIED = Widget_Button(WID_BASE_CHOICE, $
  UNAME='WID_BUTTON_CHOICE_APPLIED',SCR_XSIZE=208,SCR_YSIZE=22 $
  ,/ALIGN_LEFT,VALUE='Applied Voyager DE-STR MALDI TOF')

WID_BUTTON_CHOICE_BRUKER = Widget_Button(WID_BASE_CHOICE, $
  UNAME='WID_BUTTON_CHOICE_BRUKER',SCR_XSIZE=187,SCR_YSIZE=22 $
  ,/ALIGN_LEFT,VALUE='Bruker UltraFlex MALDI TOF TOF')

WID_BUTTON_CHOICE_ACCEPT = Widget_Button(WID_BASE_IMAGING, $
  UNAME='WID_BUTTON_CHOICE_ACCEPT',XOFFSET=403,YOFFSET=117 $
  ,SCR_XSIZE=82,SCR_YSIZE=44,TAB_MODE=2,/ALIGN_CENTER $
  ,VALUE='Accept')

WID_LABEL_RECONSTRUCTION_ONLY = Widget_Label(WID_BASE_IMAGING, $
  UNAME='WID_LABEL_RECONSTRUCTION_ONLY',XOFFSET=33,YOFFSET=231 $
  ,SCR_XSIZE=271,SCR_YSIZE=15,/ALIGN_LEFT,VALUE='To use data'+ $
  ' already loaded in the databank, click here :')

WID_BUTTON_RECONSTRUCTION_ONLY = Widget_Button(WID_BASE_IMAGING, $
  UNAME='WID_BUTTON_RECONSTRUCTION_ONLY',XOFFSET=323 $
  ,YOFFSET=224,TAB_MODE=2,/ALIGN_CENTER,VALUE='See already'+ $
  ' loaded data)

WID_BUTTON_PARAM_BD = Widget_Button(WID_BASE_IMAGING, $
  UNAME='WID_BUTTON_PARAM_BD',XOFFSET=196,YOFFSET=277 $
  ,TAB_MODE=2,/ALIGN_CENTER,VALUE='Change BD parameters')

Widget_Control, /REALIZE, WID_BASE_IMAGING

XManager, 'WID_BASE_IMAGING', WID_BASE_IMAGING, /NO_BLOCK

end

pro ImagingSoftware, GROUP_LEADER=wGroup, _EXTRA=_VWBExtra_
  WID_BASE_IMAGING, GROUP_LEADER=wGroup, _EXTRA=_VWBExtra_
end

```

## **ImagingSoftware eventcb.pro**

```

pro OnButtonChoicePress, Event

COMMON source, BDdatasource
COMMON parambd, BDuser, BDpassword

wchoiceapplied = WIDGET_INFO(Event.top, FIND_BY_UNAME = 'WID_BUTTON_CHOICE_APPLIED')
wchoicebruker = WIDGET_INFO(Event.top, FIND_BY_UNAME = 'WID_BUTTON_CHOICE_BRUKER')

IF WIDGET_INFO(wchoiceapplied, /BUTTON_SET) THEN BEGIN
  Applied
ENDIF ELSE BEGIN
  IF WIDGET_INFO(wchoicebruker, /BUTTON_SET) THEN BEGIN
    Bruker
  ENDIF ELSE BEGIN
    erreur=DIALOG_MESSAGE(" Please choose Applied or Bruker " , /ERROR)
  ENDIF
ENDIF

```

```

        return
    ENDELSE
ENDELSE

end

pro ImagingSoftware_eventcb
end

pro see_projects, Event

COMMON list_project, liste
COMMON source, BDdatasource
COMMON parambd, BDuser, BDpassword

oDBsee_projects = obj_new('IDLdbDatabase')
oDBsee_projects->Connect, DataSource = BDdatasource, user=BDuser[0], password=BDpassword[0]

querynb = "SELECT COUNT(*) AS nombre FROM project;"
oDBsee_projects->ExecuteSQL, querynb
oRSnb = obj_new('IDLdbRecordSet', oDBsee_projects, SQL=querynb)
resultbis = oRSnb->MoveCursor(/FIRST)
nombre=oRSnb->GetField(0)

query = "SELECT name FROM project;"
oDBsee_projects->ExecuteSQL, query
oRS = obj_new('IDLdbRecordSet', oDBsee_projects, SQL=query)
result = oRS->MoveCursor(/FIRST)
liste = STRARR(30, nombre)
liste[0]=oRS->GetField(0)
FOR i = 1, (nombre - 1) DO BEGIN
    result = oRS->MoveCursor(/NEXT)
    liste[i]=oRS->GetField(0)
ENDFOR

OBJ_DESTROY, oRS

reconstruction_only

end

pro change_bd_param, Event

COMMON BDsources, sources

oDBchange_bd_param = obj_new('IDLdbDatabase')
sources = oDBchange_bd_param->GetDatasources()

OBJ_DESTROY, oDBchange_bd_param

BDparam

end

```

## **Appliedproject.pro**

PRO AppliedProject, name, comment, fileplt, directorydata, wwidge, wlabelinfo, wlabelinfofile

```

COMMON bloc, projectID
COMMON source, BDdatasource
COMMON parambd, BDuser, BDpassword
COMMON RANGEOFMASS, mass_range_min, mass_range_max
COMMON REFAVERAGE, ref_avg
COMMON REFSUM, ref_sum
COMMON REFMAX, ref_max
COMMON MASS, masses

WIDGET_CONTROL, wlabelinfo, SET_VALUE='Please wait, work in progress ...'
WIDGET_CONTROL, wwidge, /UPDATE

oDBApplied_project = obj_new('IDLdbDatabase')
oDBApplied_project->Connect, DataSource = BDdatasource, user=BDuser[0], password=BDpassword[0]

```

```
query = "INSERT INTO project (name,comment,trademark,pltfile,datadirectory) VALUES
('"+name+"', '"+comment+"', '"+Applied'", '"+fileplt+"', '"+directorydata+"');"
```

```
query=STRJOIN(STRSPLIT(query,"\", /EXTRACT), '\')
oDBApplied_project->ExecuteSQL, query
```

```
query = "SELECT project.projectID FROM project WHERE name = '"+name+"' AND comment = '"+comment+"' AND trademark
= 'Applied' AND pltfile = '"+fileplt+"' AND datadirectory = '"+directorydata+'";"
```

```
query=STRJOIN(STRSPLIT(query,"\", /EXTRACT), '\')
oDBApplied_project->ExecuteSQL,query
oRS = obj_new('IDLdbRecordSet', oDBApplied_project, SQL=query)
```

```
result = oRS->MoveCursor(/LAST)
```

```
projectID=oRS->GetField(0)
```

```
OBJ_DESTROY, oRS
```

```
; ouverture du fichier plt et récupération des données contenues dedans
```

```
nombrepoints=FILE_LINES(filePLT)-5
OPENR, lunPLT, /GET_LUN, filePLT
positions=STRARR(5) ;les premières lignes de texte inutiles
donnees= LONARR(3,nombrepoints) ;les données en trois colonnes
READF, lunPLT, positions, donnees
```

```
position = LONARR(nombrepoints)
```

```
FOR i=0, (nombrepoints[0]-1) DO BEGIN
    query = "INSERT INTO position (position, X, Y, project) VALUES
('"+STRING(i)+"', '"+STRING(donnees[0,i])+"', '"+STRING(donnees[1,i])+"', '"+STRING(projectID)+"');"
    oDBApplied_project->ExecuteSQL,query
    queryselect = "SELECT position.positionID FROM position WHERE position = '"+STRING(i)+"' AND X
='"+STRING(donnees[0,i])+' AND Y = '"+STRING(donnees[1,i])+' AND project = '"+STRING(projectID)+"';"
    oRS = obj_new('IDLdbRecordSet', oDBApplied_project, SQL=queryselect)
    position[i]=oRS->GetField(0)
    OBJ_DESTROY, oRS
    WIDGET_CONTROL, wlabelinfo, SET_VALUE='Please wait, data are loaded.'
    WIDGET_CONTROL, wlabelinfofile, SET_VALUE='File currently read : plt.file'
    WIDGET_CONTROL, wwidget, /UPDATE
ENDFOR
```

```
FREE_LUN, lunPLT
```

```
spectrafile=FILE_SEARCH(STRJOIN(STRSPLIT(directorydata[0]+'*_1_*.xml',"\", /EXTRACT), '\'), /QUOTE,
COUNT=fichiersounon)
```

```
IF fichiersounon EQ 0 THEN BEGIN
    erreur=DIALOG_MESSAGE(" There is no XML file in the directory " , /ERROR)
    return
ENDIF
```

```
; récupération des données des spectres
```

```
xmlObj = OBJ_NEW('xml_to_peakarray')
xmlObj->ParseFile, spectrafile[0]
myArray = xmlObj->GetArray()
OBJ_DESTROY, xmlObj
```

```
lignes = N_ELEMENTS(myArray)
myArray = REFORM(myArray, 3, (lignes/3), /OVERWRITE)
```

```
; remplissage de la table position fait ici pour les fichiers bruker mais deja effectué plus haut pour les fichiers applied
```

```
query="INSERT INTO imaging.spectrum (project, mass, intensity, area, position) VALUES "
FOR j=0, (lignes/3-2) DO BEGIN
    query =
query+"('"+STRING(projectID)+"', '"+STRING(myArray[2,j])+"', '"+STRING(myArray[0,j])+"', '"+STRING(myArray[1,j])+"', '"+STRING(po
sition[0])+"');"
ENDFOR
query =
query+"('"+STRING(projectID)+"', '"+STRING(myArray[2,j])+"', '"+STRING(myArray[0,j])+"', '"+STRING(myArray[1,j])+"', '"+STRING(po
sition[0])+"');"
oDBApplied_project->ExecuteSQL,query
WIDGET_CONTROL, wlabelinfofile, SET_VALUE='File currently read : '+STRING(spectrafile[0])
```



## Bruckerproject.pro

PRO BruckerProject, name, comment, directorydata, wwidget, wlabelinfo, wlabelinfofile

```
COMMON bloc, projectID
COMMON source, BDdatasource
COMMON parambd, BDuser, BDpassword
COMMON RANGEOFMASS, mass_range_min, mass_range_max
COMMON REFAVERAGE, ref_avg
COMMON REFSUM, ref_sum
COMMON REFMAX, ref_max
COMMON MASS, masses
```

```
WIDGET_CONTROL, wlabelinfo, SET_VALUE='Please wait, work in progress ...'
WIDGET_CONTROL, wwidget, /UPDATE
```

```
oDBbrucker_project = obj_new('IDLdbDatabase')
oDBbrucker_project->Connect, DataSource = BDdatasource, user=BDuser[0], password=BDpassword[0]
sql="INSERT INTO imaging.project (name,comment,trademark,pltfile,datadirectory)
VALUES('"+name+"','"+comment+"','Bruker','no need','"+directorydata+"');"
sql=STRJOIN(STRSPLIT(sql,"", /EXTRACT), '\')
oDBbrucker_project->ExecuteSQL, sql
```

```
queryproject = "SELECT project.projectID FROM project WHERE name = '"+name+"' AND comment = '"+comment+"' AND
trademark = 'Bruker' AND pltfile = 'no need' AND datadirectory = '"+directorydata+'";"
queryproject=STRJOIN(STRSPLIT(queryproject,"", /EXTRACT), '\')
oDBbrucker_project->ExecuteSQL,queryproject
oRS = obj_new('IDLdbRecordSet', oDBbrucker_project, SQL=queryproject)
```

```
result = oRS->MoveCursor(/LAST)
```

```
projectID=oRS->GetField(0)
```

```
OBJ_DESTROY, oRS
```

```
spectrafile=FILE_SEARCH(STRJOIN(STRSPLIT(directorydata[0]+'_R00X[0-9][0-9]Y[0-9][0-9][0-9]1\*\pdata1\peaklist.xml',"", /EXTRACT), '\'), /QUOTE, COUNT=fichiersounon)
```

```
IF fichiersounon EQ 0 THEN BEGIN
    erreur=DIALOG_MESSAGE(" There is no XML file in the directory " , /ERROR)
    return
ENDIF
```

```
nb_files = N_ELEMENTS(spectrafile)
```

```
longueur = STRLEN(spectrafile[0])
nom_fichier1=STRMID(spectrafile[0],0,(longueur-36))
nom_fichier2=STRMID(spectrafile[0],(longueur-36),3) ; valeur de X
nom_fichier3=STRMID(spectrafile[0],(longueur-33),1)
nom_fichier4=STRMID(spectrafile[0],(longueur-32),3) ; valeur de Y
nom_fichier5=STRMID(spectrafile[0],(longueur-29))
```

; le remplissage de la table position va être fait en même temps que la récupération des données des spectres  
; ainsi une seule boucle pour tout

; récupération des données des spectres

```
xmlObj = OBJ_NEW('xml_to_peakarray')
xmlObj->ParseFile, spectrafile[0]
myArray = xmlObj->GetArray()
OBJ_DESTROY, xmlObj
```

```
lignes = N_ELEMENTS(myArray)
myArray = REFORM(myArray, 3, (lignes/3), /OVERWRITE)
```

; remplissage de la table position

```
positions = LONARR(nb_files)
queryposition = "INSERT INTO position (position, X, Y, project) VALUES
('"+STRING(1)+"','"+nom_fichier2+"','"+nom_fichier4+"','"+STRING(projectID)+"');"
oDBbrucker_project->ExecuteSQL,queryposition
queryselect = "SELECT position.positionID FROM position WHERE position = '"+STRING(1)+"' AND X = '"+nom_fichier2+"' AND
Y = '"+nom_fichier4+"' AND project = '"+STRING(projectID)+"';"
```

```

oDBbrouker_project->ExecuteSQL,queryselect
oRS = obj_new('IDLdbRecordSet', oDBbrouker_project, SQL=queryselect)
positions[0]=oRS->GetField(0)

WIDGET_CONTROL, wlabelinfo, SET_VALUE='Please wait, data are loaded.'
WIDGET_CONTROL, wwidget, /UPDATE

query="INSERT INTO imaging.spectrum (project, mass, intensity, area, position, reference) VALUES "
FOR j=0, (lignes/3-2) DO BEGIN
    query =
    query+"("+STRING(projectID)+","+STRING(myArray[2,j])+","+STRING(myArray[0,j])+","+STRING(myArray[1,j])+","+STRING(po
sitions[0])+","+STRING(0)+");"
ENDFOR
query =
query+"("+STRING(projectID)+","+STRING(myArray[2,j])+","+STRING(myArray[0,j])+","+STRING(myArray[1,j])+","+STRING(po
sitions[0])+","+STRING(0)+");"
oDBbrouker_project->ExecuteSQL,query
WIDGET_CONTROL, wlabelinfofile, SET_VALUE='File currently read : '+STRING(spectrafile[0])
WIDGET_CONTROL, wwidget, /UPDATE

FOR i = 1, nb_files-1 DO BEGIN
    xmlObj = OBJ_NEW('xml_to_peakarray')
    xmlObj->ParseFile, spectrafile[i]
    myArray = xmlObj->GetArray()
    OBJ_DESTROY, xmlObj
        nom_fichier2=STRMID(spectrafile[i],(longueur-36),3) ; valeur de X
        nom_fichier4=STRMID(spectrafile[i],(longueur-32),3) ; valeur de Y

        queryposition = "INSERT INTO position (position, X, Y, project) VALUES
("+STRING(i+1)+","+nom_fichier2+","+nom_fichier4+","+STRING(projectID)+");"
        oDBbrouker_project->ExecuteSQL,queryposition
        queryselect = "SELECT position.positionID FROM position WHERE position = "+STRING(i+1)+" AND X
="+nom_fichier2+" AND Y =" +nom_fichier4+" AND project = "+STRING(projectID)+";"
        oDBbrouker_project->ExecuteSQL,queryselect
        oRS = obj_new('IDLdbRecordSet', oDBbrouker_project, SQL=queryselect)
        positions[i]=oRS->GetField(0)

        IF myArray[0] NE -1 THEN BEGIN
            lignes = N_ELEMENTS(myArray)
            nom_fichier2=STRMID(spectrafile[i],(longueur-36),3) ; valeur de X
            nom_fichier4=STRMID(spectrafile[i],(longueur-32),3) ; valeur de Y
            myArray = REFORM(myArray, 3, (lignes/3), /OVERWRITE)
            query="INSERT INTO imaging.spectrum (project, mass, intensity, area, position, reference) VALUES "
            FOR j=0, (lignes/3-2) DO BEGIN
                query =
                query+"("+STRING(projectID)+","+STRING(myArray[2,j])+","+STRING(myArray[0,j])+","+STRING(myArray[1,j])+","+STRING(po
sitions[i])+","+STRING(0)+");"
            ENDFOR
            query =
            query+"("+STRING(projectID)+","+STRING(myArray[2,j])+","+STRING(myArray[0,j])+","+STRING(myArray[1,j])+","+STRING(po
sitions[i])+","+STRING(0)+");"
            oDBbrouker_project->ExecuteSQL,query
            ENDIF ELSE print, 'fichier évité', nom_fichier2+1, nom_fichier4+1
            WIDGET_CONTROL, wlabelinfofile, SET_VALUE='File currently read : '+STRING(spectrafile[i])
            WIDGET_CONTROL, wwidget, /UPDATE
        ENDFOR

        querynb = "SELECT count(distinct(masse)) as nombrelignes from (select mass as masse FROM spectrum s WHERE
s.project = "+ STRING(projectID) + " GROUP BY ROUND(mass, 1)) as nombre;"
        querybornes = "SELECT MIN(mass) as borneinf, MAX(mass) as bornesup FROM spectrum WHERE spectrum.project = "+
STRING(projectID) + " ; "
        oDBbrouker_project->ExecuteSQL, querynb
        oDBbrouker_project->ExecuteSQL, querybornes
        oRSnb = obj_new('IDLdbRecordSet', oDBbrouker_project, SQL=querynb)
        oRSbornes = obj_new('IDLdbRecordSet', oDBbrouker_project, SQL=querybornes)

        result = oRSnb->MoveCursor(/LAST)
        resultbis = oRSbornes->MoveCursor(/LAST)
        lignes = oRSnb->GetField(0)
        mass_range_min=oRSbornes->GetField(0)
        mass_range_max=oRSbornes->GetField(1)

; récupération des spectres de référence

WIDGET_CONTROL, wlabelinfofile, SET_VALUE=' Saving reference spectra '
WIDGET_CONTROL, wwidget, /UPDATE

```

```
queryref = "SELECT MAX(intensity) as maxima, SUM(intensity) as somme, AVG(intensity) as moyenne, mass FROM spectrum
s WHERE s.project ="+STRING(projectID)+" GROUP BY ROUND(mass, 1); "
```

```
oDBbruker_project->ExecuteSQL, queryref
oRSqueryref = obj_new('IDLdbRecordSet', oDBbruker_project, SQL=queryref)
result = oRSqueryref->MoveCursor(/FIRST)
masses = DBLARR(lignes)
k = 0L
WHILE k LT lignes DO BEGIN
    query1 = "INSERT INTO imaging.spectrum (project, mass, intensity, area, position, reference) VALUES "
    query2 = "INSERT INTO imaging.spectrum (project, mass, intensity, area, position, reference) VALUES "
    query3 = "INSERT INTO imaging.spectrum (project, mass, intensity, area, position, reference) VALUES "
    l = 1
    WHILE l LT 1000 AND k LT lignes-1 DO BEGIN
        masses[k] = oRSqueryref->GetField(3)
        query1 = query1 + " (" +STRING(projectID)+" ,"+STRING(masses[k])+" ,"+STRING(oRSqueryref-
>GetField(0))+", "+STRING(0)+" ,"+STRING(0)+" , 'max' );"
        query2 = query2 + " (" +STRING(projectID)+" ,"+STRING(masses[k])+" ,"+STRING(oRSqueryref-
>GetField(1))+", "+STRING(0)+" ,"+STRING(0)+" , 'sum' );"
        query3 = query3 + " (" +STRING(projectID)+" ,"+STRING(masses[k])+" ,"+STRING(oRSqueryref-
>GetField(2))+", "+STRING(0)+" ,"+STRING(0)+" , 'avg' );"
        result = oRSqueryref->MoveCursor(/NEXT)
        k = k+1
        l = l+1
    ENDWHILE
    masses[k] = oRSqueryref->GetField(3)
    query1 = query1+" (" +STRING(projectID)+" ,"+STRING(masses[k])+" ,"+STRING(oRSqueryref-
>GetField(0))+", "+STRING(0)+" ,"+STRING(0)+" , 'max' );"
    query2 = query2+" (" +STRING(projectID)+" ,"+STRING(masses[k])+" ,"+STRING(oRSqueryref-
>GetField(1))+", "+STRING(0)+" ,"+STRING(0)+" , 'sum' );"
    query3 = query3+" (" +STRING(projectID)+" ,"+STRING(masses[k])+" ,"+STRING(oRSqueryref-
>GetField(2))+", "+STRING(0)+" ,"+STRING(0)+" , 'avg' );"
    oDBbruker_project->ExecuteSQL, query1
    oDBbruker_project->ExecuteSQL, query2
    oDBbruker_project->ExecuteSQL, query3
    result = oRSqueryref->MoveCursor(/NEXT)
    k = k+1
ENDWHILE

WIDGET_CONTROL, wlabelinfofile, SET_VALUE=' Data loading is ended '
WIDGET_CONTROL, wwidget, /UPDATE
```

```
param_image
```

```
END
```

## Imageplus manipspectre\_define.pro

```
FUNCTION iimageplus_manipspectre::Init, $
    _REF_EXTRA = _extra

; Initialize the manipulator.
IF (~self->IDLitManipulator::Init( $
    NAME='Image and spectrum', $
    TYPES=['IDLIMAGE'], $
    _EXTRA = _extra)) THEN $
    RETURN, 0

; Register the cursor of the manipulator.
self->iimageplus_manipspectre::DoRegisterCursor

; Indicate success if all has succeeded.
RETURN, 1
END

; *****
; Configure the mouse down method. This is activated when
; the mouse button is clicked over the image.
PRO iimageplus_manipspectre::OnMouseDown, oWin, x, y, iButton, $
    KeyMods, nClicks
```

```

COMMON source, BDdatasource
COMMON parambd, BDuser, BDpassword
COMMON bloc, projectID
COMMON minimas, minX, minY

; Call our superclass.
self->IDLitManipulator::OnMouseDown, $
  oWin, x, y, iButton, KeyMods, nClicks

; Return if there is no selection. Otherwise validate selection.
IF (self.nSelectionList EQ 0) THEN $
  RETURN

; Access the first selected item and make sure it is an image.
olmage = (*self.pSelectionList)[0]
IF (OBJ_ISA(olmage, 'IDLitVisImage')) THEN BEGIN
  ; Set the olmage field of the class structure to be
  ; the retrieved IDLitVisImage object.
  self.olmage = olmage

  ; Record the current values for the target objects.
  iStatus = self->RecordUndoValues()
ENDIF

.....
;connect to the database
oDBspectre = obj_new('IDLdbDatabase')
oDBspectre->Connect, DataSource = BDdatasource, user=BDuser[0], password=BDpassword[0]

positionquery="select positionID from position where project= " + STRING(projectID) + " and x = " + string(x-203+minX) + " and
y = " + string(y-176+minY) + "; "
oDBspectre->ExecuteSQL, positionquery
oRSSpectre = obj_new('IDLdbRecordSet', oDBspectre, SQL=positionquery)
result = oRSSpectre->MoveCursor(/FIRST)
position = oRSSpectre->GetField(0)

lignesquery="select count(intensity) as nbignes from spectrum where project= " + STRING(projectID) + " and position = " +
string(position) + "; "
oDBspectre->ExecuteSQL, lignesquery
oRSlignes = obj_new('IDLdbRecordSet', oDBspectre, SQL=lignesquery)
result = oRSlignes->MoveCursor(/FIRST)
lignes = oRSlignes->GetField(0)

sql="select intensity, mass from spectrum where project= " + STRING(projectID) + " and position = " + string(position) + " order
by mass ; "
oDBspectre->ExecuteSQL, sql
oRSSql = obj_new('IDLdbRecordSet', oDBspectre, SQL=sql)
result = oRSSql->MoveCursor(/FIRST)
spectre = DBLARR(lignes)
masses = DBLARR(lignes)
i = 0L
WHILE (i LT lignes-1) DO BEGIN
  spectre[i]=oRSSql->GetField(0)
  masses[i]=oRSSql->GetField(1)
  result = oRSSql->MoveCursor(/NEXT)
  i = i +1
ENDWHILE

iplot, masses, spectre

END

.. *****
;
; Configure the DoRegisterCursor method
; This method will create the cursor for the manipulator.

PRO iimageplus_manipspectre::DoRegisterCursor

strArray = [ $
  ' .#.' '$
  ' .#.' '$
  ' .#.' '$

```



```

; Register the new tool.
ITREGISTER, 'Image and spectrum Tool', 'iimageplustool'

; Create an instance of the new tool.
identifier = IDLitSys_CreateTool('Image and spectrum Tool', $
  NAME = 'Image and spectrum Tool', $
  VISUALIZATION_TYPE = ['IMAGE'], $
  INITIAL_DATA = oparmset, _EXTRA = _extra, $
  TITLE = 'Image and spectrum Tool')
END

```

## **iimageplustool define.pro**

```

FUNCTION iimageplustool::Init, _REF_EXTRA = _extra

; Initialize the inherited ilmage tool. If this fails, return.
IF ~(self->IDLitToolImage::Init(_EXTRA = _extra)) THEN $
  RETURN, 0

; Register the new Image and Spectrum manipulator. The Description
; appears in the status bar when the manipulator is activated.

self->RegisterManipulator, 'Image and Spectrum', 'iimageplus_manipspectre', $
  DESCRIPTION='Click over image ' $
  + ' to display the spectrum at this coordinates', $
  ICON = FILEPATH('iimageplus_lut.bmp', $
    SUBDIRECTORY=['examples', 'doc', 'itools'])

; Indicate success.
RETURN, 1

END

; *****
;
; Tool Class Definition
PRO iimageplustool__define

; Define the structue of the tool.
structure = {iimageplustool, $
  INHERITS IDLitToolImage $ ; provides itool interface
}
END

```

## **Param image.pro**

```

pro WID_BASE_PARAM_IMAGE_event, Event

wTarget = (widget_info(Event.id,/NAME) eq 'TREE' ? $
  widget_info(Event.id, /tree_root) : event.id)

wWidget = Event.top

case wTarget of

Widget_Info(wWidget, FIND_BY_UNAME='WID_BUTTON_CREATE_IMAGE'): begin
  if( Tag_Names(Event, /STRUCTURE_NAME) eq 'WIDGET_BUTTON' )then $
    create_image, Event
  end
Widget_Info(wWidget, FIND_BY_UNAME='WID_DRAW_MAX'): begin
  if( Tag_Names(Event, /STRUCTURE_NAME) eq 'WIDGET_DRAW' )then $
    if( Event.type eq 0 )then $
      maxzoom, Event
    if( Tag_Names(Event, /STRUCTURE_NAME) eq 'WIDGET_DRAW' )then $
      if( Event.type eq 1 )then $
        maxzoom, Event
    end
Widget_Info(wWidget, FIND_BY_UNAME='WID_DRAW_SUM'): begin
  if( Tag_Names(Event, /STRUCTURE_NAME) eq 'WIDGET_DRAW' )then $
    if( Event.type eq 0 )then $
      zoomsum, Event

```

```

if( Tag_Names(Event, /STRUCTURE_NAME) eq 'WIDGET_DRAW' )then $
  if( Event.type eq 1 )then $
    zoomsum, Event
  end
end
Widget_Info(wWidget, FIND_BY_UNAME='WID_DRAW_AVERAGE'): begin
  if( Tag_Names(Event, /STRUCTURE_NAME) eq 'WIDGET_DRAW' )then $
    if( Event.type eq 0 )then $
      zoomavg, Event
    end
    if( Tag_Names(Event, /STRUCTURE_NAME) eq 'WIDGET_DRAW' )then $
      if( Event.type eq 1 )then $
        zoomavg, Event
      end
    end
  end
Widget_Info(wWidget, FIND_BY_UNAME='WID_BUTTON_ADD'): begin
  if( Tag_Names(Event, /STRUCTURE_NAME) eq 'WIDGET_BUTTON' )then $
    add, Event
  end
end
Widget_Info(wWidget, FIND_BY_UNAME='WID_BUTTON_SUP'): begin
  if( Tag_Names(Event, /STRUCTURE_NAME) eq 'WIDGET_BUTTON' )then $
    sup, Event
  end
end
else:
endcase

end

pro WID_BASE_PARAM_IMAGE, GROUP_LEADER=wGroup, _EXTRA=_VWBExtra_

Resolve_Routine, 'param_image_eventcb',/COMPILE_FULL_FILE ; Load event callback routines

WID_BASE_PARAM_IMAGE = Widget_Base( GROUP_LEADER=wGroup, $
  UNAME='WID_BASE_PARAM_IMAGE',XOFFSET=5,YOFFSET=5 $
  ,SCR_XSIZE=1037,SCR_YSIZE=523,TAB_MODE=2,TITLE='Picture'+
  ' parameters',SPACE=3,XPAD=3,YPAD=3)

WID_LABEL_MASS = Widget_Label(WID_BASE_PARAM_IMAGE, $
  UNAME='WID_LABEL_MASS',XOFFSET=733,YOFFSET=200,SCR_XSIZE=18 $
  ,SCR_YSIZE=18,/ALIGN_LEFT,VALUE='m/z')

WID_LABEL_ME = Widget_Label(WID_BASE_PARAM_IMAGE, $
  UNAME='WID_LABEL_ME',XOFFSET=691,YOFFSET=250,/ALIGN_LEFT $
  ,VALUE='Margin of error')

WID_TEXT_ME = Widget_Text(WID_BASE_PARAM_IMAGE, UNAME='WID_TEXT_ME' $
  ,XOFFSET=780,YOFFSET=250,/EDITABLE,XSIZE=20,YSIZE=1)

WID_LABEL_THRESHOLD = Widget_Label(WID_BASE_PARAM_IMAGE, $
  UNAME='WID_LABEL_THRESHOLD',XOFFSET=690,YOFFSET=300 $
  ,SCR_XSIZE=73,SCR_YSIZE=18,/ALIGN_LEFT,VALUE='Peak'+ $
  ' threshold')

WID_TEXT_THRESHOLD = Widget_Text(WID_BASE_PARAM_IMAGE, $
  UNAME='WID_TEXT_THRESHOLD',XOFFSET=780,YOFFSET=300,/EDITABLE $
  ,XSIZE=20,YSIZE=1)

WID_BUTTON_CREATE_IMAGE = Widget_Button(WID_BASE_PARAM_IMAGE, $
  UNAME='WID_BUTTON_CREATE_IMAGE',XOFFSET=691,YOFFSET=391 $
  ,SCR_XSIZE=77,SCR_YSIZE=21,/ALIGN_CENTER,VALUE='Create'+ $
  ' image')

WID_LABEL_MASS_RANGE_MIN_VALUE = Widget_Label(WID_BASE_PARAM_IMAGE, $
  UNAME='WID_LABEL_MASS_RANGE_MIN_VALUE',XOFFSET=139,YOFFSET=17 $
  ,SCR_XSIZE=151,SCR_YSIZE=18,/ALIGN_LEFT,VALUE='')

display_mass_range_min, WID_LABEL_MASS_RANGE_MIN_VALUE, _EXTRA=_VWBExtra_

WID_LABEL_MASS_RANGE_MAX_VALUE = Widget_Label(WID_BASE_PARAM_IMAGE, $
  UNAME='WID_LABEL_MASS_RANGE_MAX_VALUE',XOFFSET=500,YOFFSET=21 $

```

```

,SCR_XSIZE=152 ,SCR_YSIZE=18 ,/ALIGN_LEFT ,VALUE=")

display_mass_range_max, WID_LABEL_MASS_RANGE_MAX_VALUE, _EXTRA=_VWBExtra_

WID_LABEL_MASS_RANGE_MIN = Widget_Label(WID_BASE_PARAM_IMAGE, $
  UNAME='WID_LABEL_MASS_RANGE_MIN' ,XOFFSET=32 ,YOFFSET=20 $
  ,/ALIGN_LEFT ,VALUE='Lower mass value')

WID_LABEL_MASS_RANGE_MAX = Widget_Label(WID_BASE_PARAM_IMAGE, $
  UNAME='WID_LABEL_MASS_RANGE_MAX' ,XOFFSET=404 ,YOFFSET=23 $
  ,/ALIGN_LEFT ,VALUE='Higher mass value')

WID_DRAW_MAX = Widget_Draw(WID_BASE_PARAM_IMAGE, $
  UNAME='WID_DRAW_MAX' ,XOFFSET=94 ,YOFFSET=77 ,SCR_XSIZE=509 $
  ,SCR_YSIZE=126 ,/BUTTON_EVENTS)

display_spectrum_max, WID_DRAW_MAX, _EXTRA=_VWBExtra_

WID_DRAW_SUM = Widget_Draw(WID_BASE_PARAM_IMAGE, $
  UNAME='WID_DRAW_SUM' ,XOFFSET=95 ,YOFFSET=212 ,SCR_XSIZE=509 $
  ,SCR_YSIZE=126 ,/BUTTON_EVENTS)

display_spectrum_sum, WID_DRAW_SUM, _EXTRA=_VWBExtra_

WID_DRAW_AVERAGE = Widget_Draw(WID_BASE_PARAM_IMAGE, $
  UNAME='WID_DRAW_AVERAGE' ,XOFFSET=95 ,YOFFSET=349 $
  ,SCR_XSIZE=509 ,SCR_YSIZE=126 ,/BUTTON_EVENTS)

display_spectrum_average, WID_DRAW_AVERAGE, _EXTRA=_VWBExtra_

WID_LABEL_REFERENCES = Widget_Label(WID_BASE_PARAM_IMAGE, $
  UNAME='WID_LABEL_REFERENCES' ,XOFFSET=22 ,YOFFSET=53 $
  ,SCR_XSIZE=294 ,SCR_YSIZE=15 ,/ALIGN_LEFT ,VALUE='REFERENCE'+ $
  ' SPECTRA (click in the plot to zoom)')

WID_LABEL_REF_MAX = Widget_Label(WID_BASE_PARAM_IMAGE, $
  UNAME='WID_LABEL_REF_MAX' ,XOFFSET=30 ,YOFFSET=127 $
  ,SCR_XSIZE=49 ,SCR_YSIZE=34 ,/ALIGN_RIGHT ,VALUE='Maxima'+ $
  ' values')

WID_LABEL_REF_SUM = Widget_Label(WID_BASE_PARAM_IMAGE, $
  UNAME='WID_LABEL_REF_SUM' ,XOFFSET=35 ,YOFFSET=258 $
  ,SCR_XSIZE=50 ,SCR_YSIZE=41 ,/ALIGN_RIGHT ,VALUE='Sum of'+ $
  ' spectra')

WID_LABEL_REF_AVERAGE = Widget_Label(WID_BASE_PARAM_IMAGE, $
  UNAME='WID_LABEL_REF_AVERAGE' ,XOFFSET=34 ,YOFFSET=395 $
  ,SCR_XSIZE=52 ,SCR_YSIZE=41 ,/ALIGN_RIGHT ,VALUE='Average of'+ $
  ' spectra')

WID_BASE_CHOIX_CALCUL = Widget_Base(WID_BASE_PARAM_IMAGE, $
  UNAME='WID_BASE_CHOIX_CALCUL' ,XOFFSET=713 ,YOFFSET=62 $
  ,SCR_XSIZE=254 ,SCR_YSIZE=78 ,TITLE='IDL' ,COLUMN=1 $
  ,/EXCLUSIVE)

WID_BUTTON_CHOIX_MAX = Widget_Button(WID_BASE_CHOIX_CALCUL, $
  UNAME='WID_BUTTON_CHOIX_MAX' ,SCR_XSIZE=252 ,SCR_YSIZE=39 $
  ,/ALIGN_LEFT ,VALUE='maximum intensity of the peaks')

WID_BUTTON_CHOIX_AREA = Widget_Button(WID_BASE_CHOIX_CALCUL, $

```

```

UNAME='WID_BUTTON_CHOIX_AREA' ,SCR_XSIZE=252 ,SCR_YSIZE=39 $
,/ALIGN_LEFT ,VALUE='area of the peaks')

WID_LABEL_CHOIX = Widget_Label(WID_BASE_PARAM_IMAGE, $
UNAME='WID_LABEL_CHOIX' ,XOFFSET=692 ,YOFFSET=29 ,/ALIGN_LEFT $
,VALUE='Choose the caclulation method :')

WID_LABEL_PARAM_INFO = Widget_Label(WID_BASE_PARAM_IMAGE, $
UNAME='WID_LABEL_PARAM_INFO' ,XOFFSET=675 ,YOFFSET=451 $
,SCR_XSIZE=330 ,SCR_YSIZE=15 ,/ALIGN_LEFT ,VALUE='Please fill'+ $
'the blanks')

WID_BASE_CHOIX_ME = Widget_Base(WID_BASE_PARAM_IMAGE, $
UNAME='WID_BASE_CHOIX_ME' ,XOFFSET=923 ,YOFFSET=232 $
,SCR_XSIZE=100 ,SCR_YSIZE=55 ,TITLE='IDL' ,COLUMN=1 $
,/EXCLUSIVE)

WID_BUTTON_PERCENT = Widget_Button(WID_BASE_CHOIX_ME, $
UNAME='WID_BUTTON_PERCENT' ,SCR_XSIZE=57 ,SCR_YSIZE=20 $
,/ALIGN_LEFT ,VALUE='%')

WID_BUTTON_ABS = Widget_Button(WID_BASE_CHOIX_ME, $
UNAME='WID_BUTTON_ABS' ,/ALIGN_LEFT ,VALUE='Absolute')

WID_TEXT_MASS = Widget_Text(WID_BASE_PARAM_IMAGE, $
UNAME='WID_TEXT_MASS' ,XOFFSET=780 ,YOFFSET=200 ,/EDITABLE $
,XSIZE=20 ,YSIZE=1)

WID_BUTTON_ADD = Widget_Button(WID_BASE_PARAM_IMAGE, $
UNAME='WID_BUTTON_ADD' ,XOFFSET=923 ,YOFFSET=321 ,/ALIGN_CENTER $
,VALUE='ADD')

WID_BUTTON_SUP = Widget_Button(WID_BASE_PARAM_IMAGE, $
UNAME='WID_BUTTON_SUP' ,XOFFSET=977 ,YOFFSET=321 ,/ALIGN_CENTER $
,VALUE='SUP')

WID_DROPLIST_MASS = Widget_Droplist(WID_BASE_PARAM_IMAGE, $
UNAME='WID_DROPLIST_MASS' ,XOFFSET=803 ,YOFFSET=360 $
,SCR_XSIZE=214 ,SCR_YSIZE=23)

Widget_Control, /REALIZE, WID_BASE_PARAM_IMAGE

XManager, 'WID_BASE_PARAM_IMAGE', WID_BASE_PARAM_IMAGE, /NO_BLOCK

end

pro param_image, GROUP_LEADER=wGroup, _EXTRA=_VWBExtra_
WID_BASE_PARAM_IMAGE, GROUP_LEADER=wGroup, _EXTRA=_VWBExtra_
end

```

## **Param image eventcb.pro**

```

pro param_image_eventcb
end

pro create_image, Event

COMMON bloc, projectID
COMMON source, BDdatasource
COMMON parambd, BDuser, BDpassword
COMMON minimas, minX, minY

wwidgetparam=widget_info(Event.top,FIND_BY_UNAME='WID_BASE_PARAM_IMAGE')
wlabelinfoparam=widget_info(Event.top, FIND_BY_UNAME='WID_LABEL_PARAM_INFO')

```

```
WIDGET_CONTROL, wlabelinfoparam, SET_VALUE='Please wait, work in progress ... '  
WIDGET_CONTROL, wwidgetparam, /UPDATE
```

; récupération des valeurs entrées par l'utilisateur dans la fenêtre

```
wmasses = WIDGET_INFO(Event.top, FIND_BY_UNAME='WID_DROPLIST_MASS')  
WIDGET_CONTROL, wmasses, GET_VALUE=listemasses
```

; récupération du nombre d'éléments dans la liste pour connaître le nombre de tours de la boucle

```
nb_masses = N_ELEMENTS(listemasses)
```

```
masses=DBLARR(nb_masses)  
me=DBLARR(nb_masses)  
typeme=STRARR(nb_masses)  
threshold=DBLARR(nb_masses)
```

; récupération des masses dans un tableau, des marges d'erreur dans un autre et des seuils dans un troisième

```
FOR j=0, nb_masses-1 DO BEGIN  
    massesetc = STRSPLIT(listemasses[j], '|', /EXTRACT)  
    masses[j] = massesetc[0]  
    me[j] = massesetc[1]  
    typeme[j] = massesetc[2]  
    threshold[j] = massesetc[3]  
ENDFOR
```

; récupération de la méthode de calcul

```
wchoicemax = WIDGET_INFO(Event.top, FIND_BY_UNAME = 'WID_BUTTON_CHOIX_MAX')  
wchoicearea = WIDGET_INFO(Event.top, FIND_BY_UNAME = 'WID_BUTTON_CHOIX_AREA')
```

; mise en place de la base de données

```
oDBcreate_image = obj_new('IDLdbDatabase')  
oDBcreate_image->Connect, DataSource = BDdatasource, user=BDuser[0], password=BDpassword[0]
```

; récupération des dimensions de l'image

```
querydim="select count(distinct(X)) as largeur, count(distinct(Y)) as hauteur from position where project=" + STRING(projectID)  
+ ";"  
oDBcreate_image->ExecuteSQL, querydim  
oRSdim = obj_new('IDLdbRecordSet', oDBcreate_image, SQL=querydim)
```

```
result = oRSdim->MoveCursor(/LAST)
```

```
largeur=oRSdim->GetField(0)  
hauteur=oRSdim->GetField(1)
```

; création d'un tableau de mêmes dimensions que l'image, rempli de 0

```
image=DBLARR(largeur, hauteur, nb_masses)  
superposition=DBLARR(largeur, hauteur)
```

; récupération des valeurs de x et y pour calculer le pas  
; on a besoin du pas pour savoir si les valeurs récupérées sont adjacentes

```
querypas="select max(X) as maxX, min(x) as minX, max(y) as maxY, min(y) as minY from position where project=" +  
STRING(projectID) + ";"  
oDBcreate_image->ExecuteSQL, querypas  
oRSpas = obj_new('IDLdbRecordSet', oDBcreate_image, SQL=querypas)
```

```
result = oRSpas->MoveCursor(/FIRST)
```

```
maxX=uint(oRSpas->GetField(0))  
minX=uint(oRSpas->GetField(1))  
maxY=uint(oRSpas->GetField(2))  
minY=uint(oRSpas->GetField(3))
```

```
pas_horiz = uint((maxX-minX)/(largeur-1))  
pas_verti = uint((maxY-minY)/(hauteur-1))
```

; calcul de la fenêtre de valeurs

```
limit_min=DBLARR(nb_masses)
```

```
limit_max=DBLARR(nb_masses)
```

```
FOR j=0, nb_masses-1 DO BEGIN  
IF typeme[j] EQ 'percent' THEN BEGIN  
    limit_min[j] = (1-me[j]/100)*masses[j]  
    limit_max[j] = (1+me[j]/100)*masses[j]  
ENDIF ELSE BEGIN  
    limit_min[j] = masses[j] - me[j]  
    limit_max[j] = masses[j] + me[j]  
ENDELSE
```

;récupérer le nombre de lignes du résultat et faire une boucle for à la place de la boucle while

```
IF WIDGET_INFO(wchoicemax, /BUTTON_SET) THEN BEGIN  
querycountshort = "SELECT count(distinct(spectrum.position)) as nbrows from spectrum where mass > "  
querycount = querycountshort + STRING(limit_min[j]) + " and mass < " + STRING(limit_max[j]) + " and spectrum.project = " +  
STRING(projectID) + " and intensity > " + STRING(threshold[j]) + " ;"  
ENDIF ELSE BEGIN  
    IF WIDGET_INFO(wchoiceara, /BUTTON_SET) THEN BEGIN  
        querycountshort = "SELECT count(distinct(spectrum.position)) as nbrows from spectrum where mass > "  
        querycount = querycountshort + STRING(limit_min[j]) + " and mass < " + STRING(limit_max[j]) + " and  
spectrum.project = " + STRING(projectID) + " and area > " + STRING(threshold[j]) + " ;"  
        ENDIF ELSE BEGIN  
            erreur=DIALOG_MESSAGE(" Please choose a computing method : with the maxima or the area of the peaks " ,  
/ERROR)  
            return  
        ENDELSE  
    ENDELSE  
oDBcreate_image->ExecuteSQL, querycount  
oRScount = obj_new('IDLdbRecordSet', oDBcreate_image, SQL=querycount)
```

```
nb_rows=oRScount->GetField(0)
```

; si le radio button est sur max, le calcul se fait avec les maxs, sinon avec les aires des pics

```
IF WIDGET_INFO(wchoicemax, /BUTTON_SET) THEN BEGIN  
    ; récupération des pics  
    queryshort = "SELECT mass, X, Y, MAX(intensity) as maximum from spectrum left join position on  
spectrum.position=position.positionID where mass > " + STRING(limit_min[j]) + " and mass < " + STRING(limit_max[j]) ;+, $  
    query = queryshort + " and spectrum.project = " + STRING(projectID) + " and intensity > " + STRING(threshold[j]) + "  
and spectrum.position != 0 group by X,Y order by spectrum.position;"  
    oDBcreate_image->ExecuteSQL,query  
    oRS = obj_new('IDLdbRecordSet', oDBcreate_image, SQL=query)  
    suivant = oRS->MoveCursor(/FIRST)  
    IF(suivant EQ 1)THEN BEGIN  
        FOR i = 0, nb_rows-1 DO BEGIN  
            x=oRS->GetField(1)  
            y=oRS->GetField(2)  
            coorX=uint(x-minX)/pas_horiz  
            coorY=uint(y-minY)/pas_verti  
            value=oRS->GetField(3)  
            image[coorX, coorY, j]=value  
            suivant = oRS->MoveCursor(/NEXT)  
  
            IF value GT superposition[coorX, coorY] THEN superposition[coorX, coorY] = value  
  
        ENDFOR  
        print, string(masses[j]) + 'ok'  
    ENDIF ELSE BEGIN  
        erreur=DIALOG_MESSAGE(" There is no result to your query, please try another one " , /ERROR)  
        return  
    ENDELSE  
ENDIF ELSE BEGIN  
    IF WIDGET_INFO(wchoiceara, /BUTTON_SET) THEN BEGIN  
        ; récupération des pics  
        queryshort = "SELECT mass, X, Y, MAX(area) as maximum from spectrum left join position on  
spectrum.position=position.positionID where mass > " + STRING(limit_min[j]) + " and mass < " + STRING(limit_max[j]) ;+, $  
        query = queryshort + " and spectrum.project = " + STRING(projectID) + " and area > " +  
STRING(threshold[j]) + " and spectrum.position != 0 group by X,Y order by spectrum.position;"  
        oDBcreate_image->ExecuteSQL,query  
        oRS = obj_new('IDLdbRecordSet', oDBcreate_image, SQL=query)  
        suivant = oRS->MoveCursor(/FIRST)  
        IF(suivant EQ 1)THEN BEGIN  
            FOR i = 0, nb_rows-1 DO BEGIN  
                x=oRS->GetField(1)  
                y=oRS->GetField(2)
```

```

        coorX=(x-minX)/pas_horiz
        coorY=(y-minY)/pas_verti
        value=oRS->GetField(3)
        image[coorX, coorY, j]=value
        suivant = oRS->MoveCursor(/NEXT)
        IF value GT superposition[coorX, coorY] THEN superposition[coorX, coorY] = value

    ENDFOR
    print, string(masses[j]) + 'ok'
    ENDIF ELSE BEGIN
        erreur=DIALOG_MESSAGE(" There is no result to your query, please try another one " , /ERROR)
        return
    ENDELSE
    ENDIF ELSE BEGIN
        erreur=DIALOG_MESSAGE(" Please choose a computing method : with the maxima or the area of the peaks " ,
/ERROR)
        return
    ENDELSE
ENDELSE

ENDFOR

WIDGET_CONTROL, wlabelinfoparam, SET_VALUE='Please wait, the software will display the picture soon.'
WIDGET_CONTROL, wwidgetparam, /UPDATE

FOR i = 0, nb_masses-1 DO BEGIN
    IIMAGEPLUSTOOL, image[*,*], TITLE=STRING(masses[i])
ENDFOR

IIMAGEPLUSTOOL, superposition, TITLE='all'

WIDGET_CONTROL, wlabelinfoparam, SET_VALUE='The processing is ended, you can enter some other values.'
WIDGET_CONTROL, wwidgetparam, /UPDATE

wlist=widget_info(Event.top, FIND_BY_UNAME='WID_DROPLIST_MASS')
WIDGET_CONTROL, wlist, SET_VALUE="

end

pro display_mass_range_min, wWidget, _EXTRA=_VWBExtra_

COMMON RANGEOFMASS, mass_range_min, mass_range_max

WIDGET_CONTROL, wWidget, SET_VALUE=STRING(mass_range_min)

end

pro display_mass_range_max, wWidget, _EXTRA=_VWBExtra_

COMMON RANGEOFMASS, mass_range_min, mass_range_max

WIDGET_CONTROL, wWidget, SET_VALUE=STRING(mass_range_max)

end

pro display_spectrum_max, wWidget, _EXTRA=_VWBExtra_

COMMON MASS, masses
COMMON RANGEOFMASS, mass_range_min, mass_range_max
COMMON source, BDdatasource
COMMON parambd, BDuser, BDpassword
COMMON bloc, projectID
COMMON NB, lignes

oDBspectrum_max = obj_new('IDLdbDatabase')
oDBspectrum_max->Connect, DataSource = BDdatasource, user=BDuser[0], password=BDpassword[0]

querynblignes = "SELECT count(distinct(masse)) as nombrelignes from (select mass as masse FROM spectrum s WHERE
s.project = "+ STRING(projectID) + " GROUP BY ROUND(mass, 1)) as nombre;"
oDBspectrum_max->ExecuteSQL, querynblignes
oRSnblignes = obj_new('IDLdbRecordSet', oDBspectrum_max, SQL=querynblignes)
result = oRSnblignes->MoveCursor(/LAST)
lignes = oRSnblignes->GetField(0)
querymax="select intensity, mass from spectrum where project="+ STRING(projectID) + " AND reference = 'max' order by
mass;"
oDBspectrum_max->ExecuteSQL,querymax

```

```

oRSmax = obj_new('IDLdbRecordSet', oDBspectrum_max, SQL=querymax)

result = oRSmax->MoveCursor(/FIRST)
ref_max = DBLARR(lignes)
i = 0L
WHILE i LT lignes DO BEGIN
    ref_max[i]=oRSmax->GetField(0)
    result = oRSmax->MoveCursor(/NEXT)
    i = i +1
ENDWHILE

WIDGET_CONTROL, wWidget, /REALIZE

plot, masses, ref_max, X RANGE=[mass_range_min, mass_range_max], XSTYLE=9, XTICKFORMAT='(I8)',
YTICKFORMAT='(I8)'

end

pro display_spectrum_sum, wWidget, _EXTRA=_VWBExtra_

COMMON MASS, masses
COMMON RANGEOFMASS, mass_range_min, mass_range_max
COMMON source, BDdatasource
COMMON parambd, BDuser, BDpassword
COMMON bloc, projectID
COMMON NB, lignes

oDBspectrum_sum = obj_new('IDLdbDatabase')
oDBspectrum_sum->Connect, DataSource = BDdatasource, user=BDuser[0], password=BDpassword[0]

querysum="select intensity, mass from spectrum where project=" + STRING(projectID) + " AND reference = 'sum' order by
mass;"
oDBspectrum_sum->ExecuteSQL,querysum
oRSSum = obj_new('IDLdbRecordSet', oDBspectrum_sum, SQL=querysum)

result = oRSSum->MoveCursor(/FIRST)
ref_sum = DBLARR(lignes)
i = 0L
WHILE i LT lignes-1 DO BEGIN
    ref_sum[i]=oRSSum->GetField(0)
    result = oRSSum->MoveCursor(/NEXT)
    i = i +1
ENDWHILE

plot, masses, ref_sum, X RANGE=[mass_range_min, mass_range_max], XSTYLE=9

end

pro display_spectrum_average, wWidget, _EXTRA=_VWBExtra_

COMMON MASS, masses
COMMON RANGEOFMASS, mass_range_min, mass_range_max
COMMON source, BDdatasource
COMMON parambd, BDuser, BDpassword
COMMON bloc, projectID
COMMON NB, lignes

oDBspectrum_avg = obj_new('IDLdbDatabase')
oDBspectrum_avg->Connect, DataSource = BDdatasource, user=BDuser[0], password=BDpassword[0]

queryavg="select intensity, mass from spectrum where project=" + STRING(projectID) + " AND reference = 'avg' order by
mass;"
oDBspectrum_avg->ExecuteSQL,queryavg
oRSavg = obj_new('IDLdbRecordSet', oDBspectrum_avg, SQL=queryavg)

result = oRSavg->MoveCursor(/FIRST)
ref_avg = DBLARR(lignes)
i = 0L
WHILE i LT lignes-1 DO BEGIN
    ref_avg[i]=oRSavg->GetField(0)
    result = oRSavg->MoveCursor(/NEXT)
    i = i +1
ENDWHILE

plot, masses, ref_avg, X RANGE=[mass_range_min, mass_range_max], XSTYLE=9, XTICKFORMAT='(I8)',
YTICKFORMAT='(I8)'

```

```

end

pro add, Event

; récupération des valeurs entrées par l'utilisateur dans la fenêtre

wmass=widget_info(Event.top, FIND_BY_UNAME='WID_TEXT_MASS')
WIDGET_CONTROL, wmass, GET_VALUE=mass

wme=widget_info(Event.top, FIND_BY_UNAME='WID_TEXT_ME')
WIDGET_CONTROL, wme, GET_VALUE=me

wthreshold=widget_info(Event.top, FIND_BY_UNAME='WID_TEXT_THRESHOLD')
WIDGET_CONTROL, wthreshold, GET_VALUE=threshold

; marge d'erreur en pourcentage ou en absolue

wchoicepercent = WIDGET_INFO(Event.top, FIND_BY_UNAME = 'WID_BUTTON_PERCENT')
wchoiceabs = WIDGET_INFO(Event.top, FIND_BY_UNAME = 'WID_BUTTON_ABS')

; récupération de la droplist pour voir ce qu'elle contient

wlist=widget_info(Event.top, FIND_BY_UNAME='WID_DROPLIST_MASS')
WIDGET_CONTROL, wlist, GET_VALUE=ligne

IF ligne[0] EQ " THEN BEGIN
    IF WIDGET_INFO(wchoicepercent, /BUTTON_SET) THEN BEGIN
        ligne = mass + '|' + me + '|' + 'percent' + '|' + threshold
    ENDIF ELSE BEGIN
        IF WIDGET_INFO(wchoiceabs, /BUTTON_SET) THEN BEGIN
            ligne = mass + '|' + me + '|' + 'abs' + '|' + threshold
        ENDIF ELSE BEGIN
            erreur=DIALOG_MESSAGE(" Please choose for the margin of error : percent of the mass range or absolute
range " , /ERROR)
            return
        ENDELSE
    ENDELSE
    ENDIF ELSE BEGIN
        IF WIDGET_INFO(wchoicepercent, /BUTTON_SET) THEN BEGIN
            ligne = [ligne, mass + '|' + me + '|' + 'percent' + '|' + threshold]
        ENDIF ELSE BEGIN
            IF WIDGET_INFO(wchoiceabs, /BUTTON_SET) THEN BEGIN
                ligne = [ligne, mass + '|' + me + '|' + 'abs' + '|' + threshold]
            ENDIF ELSE BEGIN
                erreur=DIALOG_MESSAGE(" Please choose for the margin of error : percent of the mass range or absolute
range " , /ERROR)
                return
            ENDELSE
        ENDELSE
    ENDELSE
ENDIF

wlist=widget_info(Event.top, FIND_BY_UNAME='WID_DROPLIST_MASS')
WIDGET_CONTROL, wlist, SET_VALUE=ligne

end

pro maxzoom, Event

COMMON MASS, masses
COMMON RANGEOFMASS, mass_range_min, mass_range_max
COMMON source, BDdatasource
COMMON parambd, BDuser, BDpassword
COMMON bloc, projectID
COMMON NB, lignes

oDBspectrum_max = obj_new('IDLdbDatabase')
oDBspectrum_max->Connect, DataSource = BDdatasource, user=BDuser[0], password=BDpassword[0]

querynblignes = "SELECT count(distinct(masse)) as nombrelignes from (select mass as masse FROM spectrum s WHERE
s.project = "+ STRING(projectID) + " GROUP BY ROUND(mass, 1)) as nombre;"
oDBspectrum_max->ExecuteSQL, querynblignes
oRSnblignes = obj_new('IDLdbRecordSet', oDBspectrum_max, SQL=querynblignes)
result = oRSnblignes->MoveCursor(/LAST)
lignes = oRSnblignes->GetField(0)

```

```

querymax="select intensity, mass from spectrum where project=" + STRING(projectID) + " AND reference = 'max' order by
mass;"
oDBspectrum_max->ExecuteSQL,querymax
oRSmax = obj_new('IDLdbRecordSet', oDBspectrum_max, SQL=querymax)

result = oRSmax->MoveCursor(/FIRST)
ref_max = DBLARR(lignes)
i = 0L
WHILE i LT lignes DO BEGIN
    ref_max[i]=oRSmax->GetField(0)
    result = oRSmax->MoveCursor(/NEXT)
    i = i + 1
ENDWHILE

iplot, masses, ref_max, TITLE='Reference spectrum of the maxima values' ; , XRANGE=[mass_range_min, mass_range_max],
XSTYLE=9, XTICKFORMAT='(I8)', YTICKFORMAT='(I8)'

end

pro zoomsum, Event

COMMON MASS, masses
COMMON RANGEOFMASS, mass_range_min, mass_range_max
COMMON source, BDdatasource
COMMON parambd, BDuser, BDpassword
COMMON bloc, projectID
COMMON NB, lignes

oDBspectrum_sum = obj_new('IDLdbDatabase')
oDBspectrum_sum->Connect, DataSource = BDdatasource, user=BDuser[0], password=BDpassword[0]

querysum="select intensity, mass from spectrum where project=" + STRING(projectID) + " AND reference = 'sum' order by
mass;"
oDBspectrum_sum->ExecuteSQL,querysum
oRSsum = obj_new('IDLdbRecordSet', oDBspectrum_sum, SQL=querysum)

result = oRSsum->MoveCursor(/FIRST)
ref_sum = DBLARR(lignes)
i = 0L
WHILE i LT lignes-1 DO BEGIN
    ref_sum[i]=oRSsum->GetField(0)
    result = oRSsum->MoveCursor(/NEXT)
    i = i + 1
ENDWHILE

iplot, masses, ref_sum, TITLE='Sum of spectra'

end

pro zoomavg, Event

COMMON MASS, masses
COMMON RANGEOFMASS, mass_range_min, mass_range_max
COMMON source, BDdatasource
COMMON parambd, BDuser, BDpassword
COMMON bloc, projectID
COMMON NB, lignes

oDBspectrum_avg = obj_new('IDLdbDatabase')
oDBspectrum_avg->Connect, DataSource = BDdatasource, user=BDuser[0], password=BDpassword[0]

queryavg="select intensity, mass from spectrum where project=" + STRING(projectID) + " AND reference = 'avg' order by
mass;"

oDBspectrum_avg->ExecuteSQL,queryavg
oRSavg = obj_new('IDLdbRecordSet', oDBspectrum_avg, SQL=queryavg)

result = oRSavg->MoveCursor(/FIRST)
ref_avg = DBLARR(lignes)
i = 0L
WHILE i LT lignes-1 DO BEGIN
    ref_avg[i]=oRSavg->GetField(0)
    result = oRSavg->MoveCursor(/NEXT)
    i = i + 1

```

ENDWHILE

iplot, masses, ref\_avg, TITLE='Average of spectra'

end

## **Reconstruction\_only.pro**

pro WID\_BASE\_RECONSTRUCTION\_ONLY\_event, Event

wTarget = (widget\_info(Event.id,/NAME) eq 'TREE' ? \$  
widget\_info(Event.id, /tree\_root) : event.id)

wWidget = Event.top

case wTarget of

Widget\_Info(wWidget, FIND\_BY\_UNAME='WID\_BUTTON\_OPEN'): begin  
if( Tag\_Names(Event, /STRUCTURE\_NAME) eq 'WIDGET\_BUTTON' )then \$  
open\_project, Event  
end  
else:  
endcase

end

pro WID\_BASE\_RECONSTRUCTION\_ONLY, GROUP\_LEADER=wGroup, \_EXTRA=\_VWBExtra\_

Resolve\_Routine, 'reconstruction\_only\_eventcb',/COMPILE\_FULL\_FILE ; Load event callback routines

WID\_BASE\_RECONSTRUCTION\_ONLY = Widget\_Base( GROUP\_LEADER=wGroup, \$  
UNAME='WID\_BASE\_RECONSTRUCTION\_ONLY',XOFFSET=5,YOFFSET=5 \$  
,SCR\_XSIZE=483,SCR\_YSIZE=343,TAB\_MODE=2,TITLE='Projects with'+ \$  
' data already loaded in the databank',SPACE=3,XPAD=3 \$  
,YPAD=3)

WID\_LABEL\_PROJECTS = Widget\_Label(WID\_BASE\_RECONSTRUCTION\_ONLY, \$  
UNAME='WID\_LABEL\_PROJECTS',XOFFSET=94,YOFFSET=71,/ALIGN\_LEFT \$  
,VALUE='Choose one of this projects :')

WID\_BUTTON\_OPEN = Widget\_Button(WID\_BASE\_RECONSTRUCTION\_ONLY, \$  
UNAME='WID\_BUTTON\_OPEN',XOFFSET=310,YOFFSET=166 \$  
,/ALIGN\_CENTER,VALUE='OPEN')

WID\_DROPLIST\_PROJECTS = \$  
Widget\_Droplist(WID\_BASE\_RECONSTRUCTION\_ONLY, \$  
UNAME='WID\_DROPLIST\_PROJECTS',XOFFSET=105,YOFFSET=125)

display\_list\_projects, WID\_DROPLIST\_PROJECTS, \_EXTRA=\_VWBExtra\_

WID\_LABEL\_INFO\_REC = Widget\_Label(WID\_BASE\_RECONSTRUCTION\_ONLY, \$  
UNAME='WID\_LABEL\_INFO\_REC',XOFFSET=55,YOFFSET=244 \$  
,SCR\_XSIZE=416,SCR\_YSIZE=15,/ALIGN\_LEFT,VALUE='Please fill'+ \$  
' the blanks')

Widget\_Control, /REALIZE, WID\_BASE\_RECONSTRUCTION\_ONLY

XManager, 'WID\_BASE\_RECONSTRUCTION\_ONLY', WID\_BASE\_RECONSTRUCTION\_ONLY, /NO\_BLOCK

end

pro reconstruction\_only, GROUP\_LEADER=wGroup, \_EXTRA=\_VWBExtra\_

WID\_BASE\_RECONSTRUCTION\_ONLY, GROUP\_LEADER=wGroup, \_EXTRA=\_VWBExtra\_  
end

## Reconstruction only eventcb.pro

```
pro reconstruction_only_eventcb

end

pro open_project, Event

COMMON bloc, projectID
COMMON source, BDdatasource
COMMON parambd, BDuser, BDpassword
COMMON RANGEOFMASS, mass_range_min, mass_range_max
COMMON MASS, masses

wwidgetrec=widget_info(Event.top,FIND_BY_UNAME='WID_BASE_RECONSTRUCTION_ONLY')
wlabelinfo=widget_info(Event.top, FIND_BY_UNAME='WID_LABEL_INFO_REC')
WIDGET_CONTROL, wlabelinfo, SET_VALUE='Please wait, work in progress ... '
WIDGET_CONTROL, wwidgetrec, /UPDATE

wprojects=widget_info(Event.top, FIND_BY_UNAME='WID_DROPLIST_PROJECTS')

WIDGET_CONTROL, wprojects, GET_VALUE=projects
proj_choisi = widget_info(wprojects, /DROPLIST_SELECT)

oDBopen_project = obj_new('IDLdbDatabase')
sources = oDBopen_project->GetDatasources()
oDBopen_project->Connect, DataSource = BDdatasource, user=BDuser[0], password=BDpassword[0]

query = "select projectID from project where name = '"+ projts[proj_choisi] + "' ; "
oDBopen_project->ExecuteSQL,query
oRS = obj_new('IDLdbRecordSet', oDBopen_project, SQL=query)

result = oRS->MoveCursor(/LAST)
projectID = oRS->GetField(0)

querynb = "SELECT count(distinct(masse)) as nombrelignes from (select mass as masse FROM spectrum s WHERE
s.project = '"+ STRING(projectID) + "' GROUP BY ROUND(mass, 1)) as nombre;"
querybornes = "SELECT MIN(mass) as borneinf, MAX(mass) as bornesup FROM spectrum WHERE spectrum.project = '"+
STRING(projectID) + "' ; "
oDBopen_project->ExecuteSQL, querynb
oDBopen_project->ExecuteSQL, querybornes
oRSnb = obj_new('IDLdbRecordSet', oDBopen_project, SQL=querynb)
oRSbornes = obj_new('IDLdbRecordSet', oDBopen_project, SQL=querybornes)

result = oRSnb->MoveCursor(/LAST)
resultbis = oRSbornes->MoveCursor(/LAST)
lignes = oRSnb->GetField(0)
mass_range_min=oRSbornes->GetField(0)
mass_range_max=oRSbornes->GetField(1)

querymass = "SELECT mass FROM spectrum s WHERE s.project = '"+STRING(projectID)+" AND reference='avg' order by
mass; "

oDBopen_project->ExecuteSQL, querymass

oRSquerymass = obj_new('IDLdbRecordSet', oDBopen_project, SQL=querymass)

result = oRSquerymass->MoveCursor(/FIRST)

masses = DBLARR(lignes)
i = 0L
WHILE i LT lignes-1 DO BEGIN
    masses[i]=oRSquerymass->GetField(0)
    result = oRSquerymass->MoveCursor(/NEXT)
    i = i +1
ENDWHILE
OBJ_DESTROY, oRSquerymass

param_image

WIDGET_CONTROL, wlabelinfo, SET_VALUE='Process ended. '
WIDGET_CONTROL, wwidgetrec, /UPDATE

end
```

```

pro display_list_projects, wWidget, _EXTRA=_VWBExtra_

COMMON list_project, liste

WIDGET_CONTROL, wWidget, SET_VALUE=liste

end

```

## Xml to peakarray define.pro

```

FUNCTION xml_to_peakarray::Init
  self.pkArray = PTR_NEW(float,/ALLOCATE_HEAP)
  RETURN, self->IDLffxmlsax::Init()
END

PRO xml_to_peakarray::Cleanup
  IF (PTR_VALID(self.pkArray)) THEN PTR_FREE, self.pkArray
  self->IDLffXMLSAX::Cleanup
END

PRO xml_to_peakarray::characters, data
  self.charBuffer = self.charBuffer + data
END

PRO xml_to_peakarray::StartDocument
  IF (N_ELEMENTS(*self.pkArray) GT 0) THEN $
    void = TEMPORARY(*self.pkArray)
  END

PRO xml_to_peakarray::startElement, URI, local, strName, attrName, attrValue

  CASE strName OF
    "pklist": BEGIN
      IF (N_ELEMENTS(*self.pkArray) GT 0) THEN $
        void = TEMPORARY(*self.pkArray); clear out memory
      END
      "pk" : ; Do nothing
      "absi" : self.charBuffer = ""
      "area" : self.charBuffer = ""
      "mass" : self.charBuffer = ""
      ELSE : ; Do nothing
    ENDCASE

  END

PRO xml_to_peakarray::EndElement, URI, Local, strName

  CASE strName of
    "pklist": ; Do nothing
    "pk":
      "absi" : BEGIN
        IF (N_ELEMENTS(*self.pkArray) EQ 0) THEN $
          *self.pkArray = DOUBLE(self.charBuffer) $
        ELSE $
          *self.pkArray = [*self.pkArray, DOUBLE(self.charBuffer)]
        END
      "area" : BEGIN
        IF (N_ELEMENTS(*self.pkArray) EQ 0) THEN $
          *self.pkArray = DOUBLE(self.charBuffer) $
        ELSE $
          *self.pkArray = [*self.pkArray, DOUBLE(self.charBuffer)]
        END
      "mass" : BEGIN
        IF (N_ELEMENTS(*self.pkArray) EQ 0) THEN $
          *self.pkArray = DOUBLE(self.charBuffer) $
        ELSE $
          *self.pkArray = [*self.pkArray, DOUBLE(self.charBuffer)]
        END
      ELSE : ; Do nothing

```

```

ENDCASE

END

FUNCTION xml_to_peakarray::GetArray
  IF (N_ELEMENTS(*self.pkArray) GT 0) THEN $
    RETURN, *self.pkArray $
  ELSE RETURN, -1
END

END

PRO xml_to_peakarray__define
  void = {xml_to_peakarray, $
    INHERITS IDLffXMLSAX, $
    charBuffer : "", $
    pkArray : PTR_NEW() }

END

```

## **Applied.prc**

```

HEADER
; IDL Visual Widget Builder Resource file. Version 1
; Generated on: 12/18/2007 00:11.05
VERSION 1
END

WID_BASE_APPLIED BASE 5 5 630 455
TAB_MODE = 3
TLB
CAPTION "IDL"
XPAD = 3
YPAD = 3
SPACE = 3
BEGIN
  WID_BUTTON_Accept_APPLIED PUSHBUTTON 209 275 124 38
  VALUE "Accept"
  ALIGNCENTER
  TAB_MODE = 3
  ONACTIVATE "OnButtonPressAccept_APPLIED"
  END
  WID_LABEL_TITLE_APPLIED LABEL 40 25 372 18
  VALUE "APPLIED NEW PROJECT"
  ALIGNLEFT
  END
  WID_LABEL_NAME_APPLIED LABEL 39 73 0 0
  VALUE "Name"
  ALIGNLEFT
  END
  WID_TEXT_PROJECT_NAME_APPLIED TEXT 84 70 367 21
  TAB_MODE = 3
  EDITABLE
  WIDTH = 20
  HEIGHT = 1
  END
  WID_LABEL_COMMENT_APPLIED LABEL 29 121 0 0
  VALUE "Comment"
  ALIGNLEFT
  END
  WID_TEXT_PROJECT_COMMENT_APPLIED TEXT 83 121 367 21
  TAB_MODE = 3
  EDITABLE
  WIDTH = 20
  HEIGHT = 1

```

```

END
WID_TEXT_DATA_APPLIED TEXT 83 220 367 21
TAB_MODE = 3
EDITABLE
WIDTH = 20
HEIGHT = 1
END
WID_LABEL_DATA_APPLIED LABEL 25 222 45 18
VALUE "Data files"
ALIGNLEFT
END
WID_TEXT_PLT_APPLIED TEXT 84 170 367 21
TAB_MODE = 3
EDITABLE
WIDTH = 20
HEIGHT = 1
END
WID_LABEL_PLT_APPLIED LABEL 30 172 38 18
VALUE "PLT file"
ALIGNLEFT
END
WID_BUTTON_ExplorerPLT_APPLIED PUSHBUTTON 470 173 47 17
VALUE "Browse"
ALIGNCENTER
TAB_MODE = 3
ONACTIVATE "OnButtonPressExplorerPLT_APPLIED"
END
WID_BUTTON_ExplorerDATA_APPLIED PUSHBUTTON 470 222 47 18
VALUE "Browse"
ALIGNCENTER
TAB_MODE = 3
ONACTIVATE "OnButtonPressExplorerData_APPLIED"
END
WID_LABEL_INFO_APP LABEL 39 341 580 15
VALUE "Please fill the text areas"
ALIGNLEFT
END
WID_LABEL_INFO_FILE_APP LABEL 41 374 575 15
VALUE "No file is currently read."
ALIGNLEFT
END
END

```

## **BDparam.prc**

```

HEADER
; IDL Visual Widget Builder Resource file. Version 1
; Generated on: 05/02/2008 10:20.52
VERSION 1
END

```

```

WID_BASE_BD_PARAM BASE 5 5 517 352
TAB_MODE = 3
TLB
CAPTION "Databank parameters"
XPAD = 3
YPAD = 3
SPACE = 3
BEGIN

```

```

WID_DROPLIST_DATASOURCE DROPLIST 68 78 0 0
ONPOSTCREATE "display_datasources"
CAPTION "choose the data source :"
NUMITEMS = 3
ITEM "serveur"
ITEM "imaging"
ITEM "pc jerome"
END
WID_LABEL_DATASOURCE LABEL 33 39 0 0
VALUE "DATASOURCE"
ALIGNLEFT
END
WID_LABEL_USER LABEL 35 121 0 0
VALUE "USER"
ALIGNLEFT
END
WID_LABEL_PASSWORD LABEL 24 197 0 0
VALUE "PASSWORD"
ALIGNLEFT
END
WID_BUTTON_CHANGE PUSHBUTTON 312 219 0 0
VALUE "Use these parameters"
ALIGNCENTER
ONACTIVATE "use_parameters"
END
WID_TEXT_USER TEXT 67 151 0 0
NUMITEMS = 2
ITEM "root"
ITEM ""
EDITABLE
WIDTH = 20
HEIGHT = 1
END
WID_TEXT_PASSWORD TEXT 68 235 0 0
NUMITEMS = 2
ITEM "toor"
ITEM ""
EDITABLE
WIDTH = 20
HEIGHT = 1
END
END

```

## **Bruker.prc**

```

HEADER
; IDL Visual Widget Builder Resource file. Version 1
; Generated on: 12/18/2007 00:10.46
VERSION 1
END

WID_BASE_BRUKER BASE 5 5 675 424
TAB_MODE = 3
TLB
CAPTION "IDL"
XPAD = 3
YPAD = 3
SPACE = 3

```

```

BEGIN
  WID_BUTTON_ExplorerDATA_BRUKER PUSHBUTTON 465 193 47 18
  VALUE "Browse"
  ALIGNCENTER
  ONACTIVATE "OnButtonPressExplorerData_BRUKER"
  END
  WID_LABEL_DATA_BRUKER LABEL 24 193 45 18
  VALUE "Data files"
  ALIGNLEFT
  END
  WID_TEXT_DATA_BRUKER TEXT 83 191 367 21
  EDITABLE
  WIDTH = 20
  HEIGHT = 1
  END
  WID_TEXT_PROJECT_COMMENT_BRUKER TEXT 83 136 367 21
  EDITABLE
  WIDTH = 20
  HEIGHT = 1
  END
  WID_LABEL_COMMENT_BRUKER LABEL 28 137 0 0
  VALUE "Comment"
  ALIGNLEFT
  END
  WID_TEXT_PROJECT_NAME_BRUKER TEXT 82 86 367 21
  TAB_MODE = 3
  EDITABLE
  WIDTH = 20
  HEIGHT = 1
  END
  WID_LABEL_NAME_BRUKER LABEL 37 89 0 0
  VALUE "Name"
  ALIGNLEFT
  END
  WID_LABEL_TITLE_BRUKER LABEL 38 38 372 18
  VALUE "BRUKER NEW PROJECT"
  ALIGNLEFT
  END
  WID_BUTTON_Accept PUSHBUTTON 281 249 124 38
  VALUE "Accept"
  ALIGNCENTER
  ONACTIVATE "OnButtonPressAccept_BRUKER"
  END
  WID_LABEL_INFO_BRUK LABEL 41 307 618 15
  VALUE "Please fill the text areas"
  ALIGNLEFT
  END
  WID_LABEL_INFO_FILE_BRUK LABEL 38 345 625 15
  VALUE "No file is currently read."
  ALIGNLEFT
  END
END

```

## **ImagingSoftware.prc**

```

HEADER
; IDL Visual Widget Builder Resource file. Version 1
; Generated on: 12/21/2007 15:52.56

```

```

VERSION 1
END

WID_BASE_IMAGING BASE 5 5 535 348
TAB_MODE = 2
TLB
CAPTION "MALDI IMAGING TEAM SOFTWARE"
XPAD = 3
YPAD = 3
SPACE = 3
BEGIN
    WID_LABEL_WELCOME LABEL 122 24 0 0
    VALUE "WELCOME ON THE MALDI IMAGING TEAM SOFTWARE"
    ALIGNLEFT
    END
    WID_LABEL_CHOICE LABEL 27 80 0 0
    VALUE "To load data in the databank before the reconstruction, choose one
of the spectrometers :"
    ALIGNLEFT
    END
    WID_BASE_CHOICE BASE 152 108 208 101
    COLUMNS = 1
    EXCLUSIVE
    CAPTION "IDL"
    BEGIN
        WID_BUTTON_CHOICE_APPLIED PUSHBUTTON -1 -1 208 22
        VALUE "Applied Voyager DE-STR MALDI TOF"
        ALIGNLEFT
        ONACTIVATE "OnButtonAppliedPress"
        END
        WID_BUTTON_CHOICE_BRUKER PUSHBUTTON -1 -1 187 22
        VALUE "Bruker UltraFlex MALDI TOF TOF"
        ALIGNLEFT
        ONACTIVATE "OnButtonAppliedPress"
        END
    END
    WID_BUTTON_CHOICE_ACCEPT PUSHBUTTON 403 117 82 44
    VALUE "Accept"
    ALIGNCENTER
    TAB_MODE = 3
    ONACTIVATE "OnButtonChoicePress"
    END
    WID_LABEL_RECONSTRUCTION_ONLY LABEL 33 231 271 15
    VALUE "To use data already loaded in the databank, click here :"
    ALIGNLEFT
    END
    WID_BUTTON_RECONSTRUCTION_ONLY PUSHBUTTON 323 224 0 0
    VALUE "See already loaded data"
    ALIGNCENTER
    TAB_MODE = 3
    ONACTIVATE "see_projects"
    END
    WID_BUTTON_PARAM_BD PUSHBUTTON 196 277 0 0
    VALUE "Change BD parameters"
    ALIGNCENTER
    TAB_MODE = 3
    ONACTIVATE "change_bd_param"
    END
END
END

```

## Param image.prc

```
HEADER
; IDL Visual Widget Builder Resource file. Version 1
; Generated on: 05/06/2008 14:54.50
VERSION 1
END

WID_BASE_PARAM_IMAGE BASE 5 5 1037 523
TAB_MODE = 3
TLB
CAPTION "Picture parameters"
XPAD = 3
YPAD = 3
SPACE = 3
BEGIN
  WID_LABEL_MASS LABEL 733 200 18 18
  VALUE "m/z"
  ALIGNLEFT
  END
  WID_LABEL_ME LABEL 691 250 0 0
  VALUE "Margin of error"
  ALIGNLEFT
  END
  WID_TEXT_ME TEXT 780 250 0 0
  EDITABLE
  WIDTH = 20
  HEIGHT = 1
  END
  WID_LABEL_THRESHOLD LABEL 690 300 73 18
  VALUE "Peak threshold"
  ALIGNLEFT
  END
  WID_TEXT_THRESHOLD TEXT 780 300 0 0
  EDITABLE
  WIDTH = 20
  HEIGHT = 1
  END
  WID_BUTTON_CREATE_IMAGE PUSHBUTTON 691 391 77 21
  VALUE "Create image"
  ALIGNCENTER
  ONACTIVATE "create_image"
  END
  WID_LABEL_MASS_RANGE_MIN_VALUE LABEL 139 17 151 18
  VALUE ""
  ALIGNLEFT
  ONPOSTCREATE "display_mass_range_min"
  END
  WID_LABEL_MASS_RANGE_MAX_VALUE LABEL 500 21 152 18
  VALUE ""
  ALIGNLEFT
  ONPOSTCREATE "display_mass_range_max"
  END
  WID_LABEL_MASS_RANGE_MIN LABEL 32 20 0 0
  VALUE "Lower mass value"
  ALIGNLEFT
  END
  WID_LABEL_MASS_RANGE_MAX LABEL 404 23 0 0
  VALUE "Higher mass value"
  ALIGNLEFT
```

```

END
WID_DRAW_MAX DRAW 94 77 509 126
ONPOSTCREATE "display_spectrum_max"
ONBUTTON "maxzoom"
END
WID_DRAW_SUM DRAW 95 212 509 126
ONPOSTCREATE "display_spectrum_sum"
ONBUTTON "zoomsum"
END
WID_DRAW_AVERAGE DRAW 95 349 509 126
ONPOSTCREATE "display_spectrum_average"
ONBUTTON "zoomavg"
END
WID_LABEL_REFERENCES LABEL 22 53 294 15
VALUE "REFERENCE SPECTRA (click in the plot to zoom)"
ALIGNLEFT
END
WID_LABEL_REF_MAX LABEL 30 127 49 34
VALUE "Maxima values"
ALIGNRIGHT
END
WID_LABEL_REF_SUM LABEL 35 258 50 41
VALUE "Sum of spectra"
ALIGNRIGHT
END
WID_LABEL_REF_AVERAGE LABEL 34 395 52 41
VALUE "Average of spectra"
ALIGNRIGHT
END
WID_BASE_CHOIX_CALCUL BASE 713 62 254 78
COLUMNS = 1
EXCLUSIVE
CAPTION "IDL"
BEGIN
  WID_BUTTON_CHOIX_MAX PUSHBUTTON -1 -1 252 39
  VALUE "maximum intensity of the peaks"
  ALIGNLEFT
  END
  WID_BUTTON_CHOIX_AREA PUSHBUTTON -1 -1 252 39
  VALUE "area of the peaks"
  ALIGNLEFT
  END
END
WID_LABEL_CHOIX LABEL 692 29 0 0
VALUE "Choose the caclulation method :"
ALIGNLEFT
END
WID_LABEL_PARAM_INFO LABEL 675 451 330 15
VALUE "Please fill the blanks"
ALIGNLEFT
END
WID_BASE_CHOIX_ME BASE 923 232 100 55
COLUMNS = 1
EXCLUSIVE
CAPTION "IDL"
BEGIN
  WID_BUTTON_PERCENT PUSHBUTTON -1 -1 57 20
  VALUE "%"
  ALIGNLEFT
  END
  WID_BUTTON_ABS PUSHBUTTON -1 -1 0 0

```

```

        VALUE "Absolute"
        ALIGNLEFT
        END
    END
    WID_TEXT_MASS TEXT 780 200 0 0
    EDITABLE
    WIDTH = 20
    HEIGHT = 1
    END
    WID_BUTTON_ADD PUSHBUTTON 923 321 0 0
    VALUE "ADD"
    ALIGNCENTER
    ONACTIVATE "add"
    END
    WID_BUTTON_SUP PUSHBUTTON 977 321 0 0
    VALUE "SUP"
    ALIGNCENTER
    ONACTIVATE "sup"
    END
    WID_DROPLIST_MASS DROPLIST 803 360 214 23
    END
END

```

### **Reconstruction\_only.prc**

```

HEADER
; IDL Visual Widget Builder Resource file. Version 1
; Generated on: 12/18/2007 00:11.52
VERSION 1
END

WID_BASE_RECONSTRUCTION_ONLY BASE 5 5 483 343
TAB_MODE = 3
TLB
CAPTION "Projects with data already loaded in the databank"
XPAD = 3
YPAD = 3
SPACE = 3
BEGIN
    WID_LABEL_PROJECTS LABEL 94 71 0 0
    VALUE "Choose one of this projects :"
    ALIGNLEFT
    END
    WID_BUTTON_OPEN PUSHBUTTON 310 166 0 0
    VALUE "OPEN"
    ALIGNCENTER
    ONACTIVATE "open_project"
    END
    WID_DROPLIST_PROJECTS DROPLIST 105 125 0 0
    ONPOSTCREATE "display_list_projects"
    END
    WID_LABEL_INFO_REC LABEL 55 244 416 15
    VALUE "Please fill the blanks"
    ALIGNLEFT
    END
END

```