Abstract

LC-MSMS combined with library search approaches is getting popular as an alternative to existing methods in clinical toxicology, for the screening of toxics. Such screenings are traditionally operated in emergency units, where they act as an early diagnostic test of the patient condition. As a result, routine laboratories need robust and easy to use software solutions that can handle the complexity of spectra generated by LCMSMS as well as deliver high quality results in the shortest time possible. In this thesis, we present a small molecule identification platform called SmileMS. This platform was developed to be used in applications based on LC-MSMS identification but also to act as an exploration platform for further investigation in the field of small molecule identification. Throughout the conception of SmileMS, these investigations concerned four areas: the creation scoring models, the handling of spectral libraries, the theoretical fragmentation of molecules, and the setting of engineering strategies enabling software exploration of data.

Reference


URN : urn:nbn:ch:unige-85792
DOI : 10.13097/archive-ouverte/unige:8579

Available at:
http://archive-ouverte.unige.ch/unige:8579

Disclaimer: layout of this document may differ from the published version.
LC-MSMS identification of small molecules; 

*SmileMS*, a small molecule identification platform applied to clinical toxicology

**THÈSE**

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

pour obtenir le grade de Docteur ès sciences, mention 

bioinformatique

par

Yann Mauron 

de 

Genève (Suisse)

These N° 4203

GENÈVE 

2010
Abstract

Liquid Chromatography Tandem Mass Spectrometry (LC-MSMS) combined with library search approaches is getting popular as an alternative to existing methods in clinical toxicology, for the screening of toxics. Such screenings are traditionally operated in emergency units, where they act as an early diagnostic test of the patient condition. As a result, routine laboratories need robust and easy to use software solutions that can handle the complexity of spectra generated by LC-MSMS as well as deliver high quality results in the shortest time possible.

In this thesis, we present a small molecule identification platform called SmileMS. This platform was developed to be used in applications based on LC-MSMS identification but also to act as an exploration platform for further investigation in the field of small molecule identification. Throughout the conception of SmileMS, these investigations concerned four main areas: the creation and enhancement of scoring models, the handling of spectral libraries, the theoretical fragmentation of chemical structures, and the setting of mechanisms and engineering strategies enabling software exploration of data.

In SmileMS, identification is based on X-Rank, a scoring model relying on statistical observations. This scoring model is the result of our exploration of data acquired at the University Hospital of Geneva. We recently published the mathematic principles of this scoring model in Analytical Chemistry, a important journal in the field of small molecule identification. The discriminating power of X-Rank was benchmarked against other identification algorithms, using datasets from different instruments. Overall, X-Rank showed better results in terms of sensitivity and specificity. In the case of cross-platform identification particularly, X-Rank could better discriminate correct from wrong matches. Additional enhancements of X-Rank further improved performance and led to clearly classify results into correct and wrong identifications.

Besides algorithms, a major limitation for library search in LC-MSMS is the poor availability of spectral libraries. In this context, we present a standardization of libraries, which enables their use in multiple applications and allows sharing them among collaborating laboratories. Hence, this approach represents a solution to the scarcity of libraries, as well as to the expensive cost of building them.

We also explored aspects related to the theoretical fragmentation of chemical compounds. To gain deeper insight into spectra fragmen-
tation patterns in mass spectrometry, two dedicated tools were developed: the first to visualize molecules and their fragments and the second to automatically construct *fragmentation graphs*. The notion of *fragmentation graph*, which considers fragments of a molecule a system rather than disconnected parts, is a first step towards the modeling of fragmentation patterns. The understanding of the fragmentation patterns is important for spectra annotation, for verifying uncertain identifications and, more ambitiously, for the identification of unknown molecules.

Finally, from a software engineering point of view, we conceived a platform able to help us to explore, but also to integrate new development. Particularly, thanks to the modular design of *SmileMS* and the use of open source libraries, exploration areas exposed in this thesis were facilitated and easily integrated in the platform.
French Summary

Combinée à une approche d’identification par comparaison avec une bibliothèque de spectres de référence, la chromatographie en phase liquide, couplée à une spectrométrie de masse en tandem (LC-MSMS), tend à s’imposer en tant que méthode analytique de choix, lors de la recherche de toxiques en milieu médical. Ce type de recherche est très fréquemment employé en milieu hospitalier afin d’établir un premier diagnostic, par exemple lors d’admissions de patients au service des urgences. Il en résulte une forte demande de la part des laboratoires de toxicologie pour des applications logicielles permettant une prise en charge efficace de la complexité générée par la LC-MSMS. De plus, ces applications se doivent d’être faciles d’emploi et de produire des résultats pertinents, dans un laps de temps réduit.

Le présent document décrit SmileMS, une plate-forme logicielle dédiée à l’identification de petites molécules. Cette plate-forme a été développée pour être utilisée dans le cadre d’activités en relation avec l’identification LC-MSMS, mais également dans le but de soutenir l’exploration scientifique relative au domaine de l’identification de petites molécules. Au travers de la conception de SmileMS, quatre domaines ont été particulièrement explorés: la création et l’amélioration de métriques d’identification, la gestion de bibliothèques de référence, les mécanismes sous-jacents à la fragmentation moléculaire et la conception de mécanismes ainsi que l’utilisation de stratégies d’ingénierie software permettant l’exploration de données.

Outres les aspects algorithmiques, la disponibilité restreinte en matière de bibliothèques de référence est une limitation supplémentaire à l’applicabilité de la recherche en bibliothèque, dans le domaine de la LC-MSMS. Afin de répondre à cette problématique, nous présentons une approche permettant de normaliser les bibliothèques de spectres existantes, les rendant utilisables dans plusieurs contextes et leur permettant d’être partagées au sein de laboratoires collaborants. La problématique du coût de conception des bibliothèques de référence pourrait également être atténuée par ce type de mécanismes de partage d’information.

Une autre voie d’exploration traitait de la fragmentation théorique de structures moléculaires. Afin d’approfondir notre connaissance dans ce domaine, nous avons conçu deux applications permettant respectivement de visualiser des structures moléculaires ainsi que leurs fragments et de construire de manière automatique des graphiques de fragmentation. Cette notion de graphiques de fragmentation propose une vision systémique des fragments moléculaires, opposée à la vision plus traditionnelle de fragments indépendants. Cette représentation est également un premier pas vers la modélisation des mécanismes de fragmentation moléculaire. Une meilleure compréhension de ces mécanismes serait d’intérêt majeur, permettant l’annotation de spectres de fragmentation, la vérification d’identifications douteuses, ainsi que plus ambitieusement, l’identification de molécules ne possédant pas de spectre de référence dans la bibliothèque utilisée pour l’identification.

Finalement, d’un point de vue d’ingénierie logicielle, nous présentons une plate-forme capable de soutenir l’exploration de données et d’intégrer facilement de nouveaux développements. Les explorations décrites dans ce résumé, entre autres, ont pu être facilitées et aisément intégrées grâce à la conception modulaire de SmileMS.
Acknowledgments

I would like to thank all people who have helped and inspired me during my doctoral study.

I especially want to thank my three advisors to help me realize this thesis. I thank Dr. Frédérique Lisacek from the Swiss Institute of Bioinformatics (SIB), for her support in many ways. A special thanks goes to Dr. Alexandre Masselot from Genebio, whose encouragement, guidance and support from the initial to the final level enabled me to develop an understanding of the subject. Finally, I want to thank Professor Ron Appel from the Swiss Institute of Bioinformatics (SIB) for making this thesis possible.

Besides my advisors, I would like to acknowledge the rest of my thesis committee, Prof. Bastian Chopard from the Centre universitaire d’informatique (CUI) of the University of Geneva and Dr. Marc Augsburger from the Centre Universitaire Romand de Médecine Légale (CURML), for accepting to be part of the jury.

From Genebio I would like to thank first Nasri Nahas, who offered to me the possibility to undertake this joint Ph.D. project. Moreover I would like to show my gratitude to Dr. Catherine Zwahlen, Dr. Nicolas Budin, Dr. Pierre-Alain Binz, Patrick Brechbiehl, Olivier Evalet, and to all present and former members of Genebio, for their helpful collaborations and for creating such a pleasant working atmosphere.

At the Swiss Institute of Bioinformatics (SIB), it is a pleasure to thank Frederic Nikitin, Erik Ahrme, Dr. Markus Mueller, and all the other members that were involved in our project.

In the frame of our collaboration with the University Hospital of Geneva (HUG), I owe my deepest gratitude to Prof. Denis Hochstrasser, Dr. Marc Fathi, Véronique Viette, and all the others from the clinical toxicology laboratory for this prolific and interesting cooperation, which allowed me to gain closer insight into a direct field of application for Mass Spectrometry.

For the beneficial collaboration, I would like to give props to all the
members of the Centre Universitaire Romand de Médecine Légale (CURML), and especially to Dr. Marc Augsburger, Hicham Kharbouche and Anette Jordan.

I say also thank Dr. Patrick Edder, Dr. Didier Ortelli and all the other co-workers from the Service de la consommation et des affaires vétérinaires (SCAV), for their interest and for working with us on this common project.

I express my gratitude towards the KTI / CTI fund which provided an essential financial support that made the undertaking of this Ph.D. project possible.

Furthermore, I would like to acknowledge the Doctoral School of Bioinformatics and the University of Geneva for offering this Ph.D. program, which decisively enabled me to increase my knowledge in bioinformatics and to achieve this important step in my career.

This thesis would not have been possible without the flawless commitment of my dear fellow-worker Roman Mylonas.

I also want to thank my family Dominique, Serge and Caroline, as well as my friends and co-workers Michael, David, and Justin.

Last but not least, I would like to give my special thanks to Véra whose patience and love enabled me to complete this work.
Guide to readers

This thesis is divided into four parts. First, the Introduction exposes the environment in which my project is inscribed. Second, the Challenges chapter highlights the main challenges addressed in the next chapters. Third, the six following chapters detail the work done around six key areas; the identification of small molecules from an algorithmic point of view, the issue of spectral libraries, the complexity of chemical compound fragmentation, the engineering of an exploratory and production platform, the identification platform in itself, and a first application of this platform at the University Hospital of Geneva. Finally, a discussion concludes this thesis.

Being inscribed in a bioinformatic context, this thesis presents two orientations; some more biological aspects and other related to computer science. Particularly, chapter 6 tends to be more computer-related, whereas others are application-oriented.

Furthermore, SmileMS is a project supported by many collaborations and individuals. Among them, Roman Mylonas dedicated his doctorate work to this project. Our two thesis should be considered complementary, since they offer a more global view on a project with multiple applications and implications. We both contributed to all topics covered in the two documents. In this respect, this document contains some parts that are also discussed in the thesis of Roman Mylonas. These parts are noted with a (*) after the chapter or section titles. We also point to each other thesis, when a point is discussed there in more details. We tried to help consistent notation between the two documents the most often possible.

Finally, this project includes a production dimension, since SmileMS is one of the products proposed by Geneva Bioinformatics and is routinely used at the University Hospital of Geneva.
# Table of Contents

1 **Introduction** ........................................... 15  
1.1 From biology to bioinformatics ......................... 16  
1.2 Clinical toxicology: a story of toxics .................. 20  
1.3 Current screening instrumentations .................... 22  
1.4 Current instrument software solutions (*) .............. 34  
1.5 *SmileMS*: a small molecule identification platform .... 44  

2 **Challenges** ............................................. 45  
2.1 Identification of small molecules ....................... 46  
2.2 The scarcity of spectral libraries ....................... 49  
2.3 Further investigation: LC-MSMS Fragmentation patterns 50  
2.4 *SmileMS*: a small molecule identification platform .... 51  
2.5 Study case: the emergency department of the University  
Hospital of Geneva ....................................... 53  

3 **X-Rank scoring model** ................................. 55  
3.1 Algorithm description ................................... 56  
3.2 Scoring model performance (*) ......................... 61  
3.3 Scoring model enhancement ............................. 69  

4 **Spectral Libraries** ..................................... 79  
4.1 Library architecture and interactions .................. 80  
4.2 Library management and sharing ........................ 84  

5 **Chemical Compound Fragmentation** .................... 91  
5.1 Context .................................................. 92  
5.2 Fragmentation-energy driven fragmentation ............. 93  
5.3 Software applications ................................ 97
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 Exploratory software engineering</td>
<td></td>
</tr>
<tr>
<td>6.1 Context</td>
<td>110</td>
</tr>
<tr>
<td>6.2 SmileMS software development</td>
<td>116</td>
</tr>
<tr>
<td>6.3 Achievement and prospect</td>
<td>140</td>
</tr>
<tr>
<td>7 SmileMs Identification Platform: concepts and usage (*)</td>
<td>143</td>
</tr>
<tr>
<td>7.1 Platform overview</td>
<td>144</td>
</tr>
<tr>
<td>7.2 Data processing</td>
<td>144</td>
</tr>
<tr>
<td>7.3 Interface usage</td>
<td>148</td>
</tr>
<tr>
<td>7.4 Bruker Daltonics: an integration case study</td>
<td>154</td>
</tr>
<tr>
<td>8 Case Study: the University Hospital of Geneva (HUG)</td>
<td>157</td>
</tr>
<tr>
<td>8.1 LC-MSMS as an alternative to the REMEDI</td>
<td>158</td>
</tr>
<tr>
<td>8.2 The HUG environment</td>
<td>160</td>
</tr>
<tr>
<td>8.3 Comparison with existing software</td>
<td>166</td>
</tr>
<tr>
<td>8.4 Libraries</td>
<td>169</td>
</tr>
<tr>
<td>9 Discussion and conclusion</td>
<td>173</td>
</tr>
<tr>
<td>9.1 Data exploration platform</td>
<td>174</td>
</tr>
<tr>
<td>9.2 Collaboration with the HUG</td>
<td>177</td>
</tr>
<tr>
<td>9.3 Beyond clinical toxicology</td>
<td>179</td>
</tr>
<tr>
<td>10 Publications, presentations and posters</td>
<td>183</td>
</tr>
<tr>
<td>Bibliography</td>
<td>185</td>
</tr>
<tr>
<td>Index</td>
<td>201</td>
</tr>
<tr>
<td>Glossary</td>
<td>205</td>
</tr>
</tbody>
</table>
List of Figures

1.1 General principle of mass spectrometry. .......................... 18
1.2 Selected applications of mass spectrometry in homeland se-
curity, space missions, water quality monitoring, and pro-
teins study. .................................................. 19
1.3 Position of the toxicology within a classification of human
diseases. ....................................................... 21
1.4 Variability of LC-MSMS spectra compared to GC-MS ones. 25
1.5 Standard steps of a LC-MSMS experiment. ....................... 27
1.6 Liquid chromatography; the reversed phase column. .......... 28
1.7 Range of action for various ionization techniques. .............. 30
1.8 Analysis of 237 real sample between a REMEDI system and
a LC-MSMS. (Courtesy of Veronique Viette) ........................ 35

2.1 Variability of spectra across instruments and acquisition pa-
rameters. .................................................................. 47
2.2 Example of extreme spectral similarity. .............................. 48

3.1 Computation of scoring model parameters; matches. .......... 58
3.2 Computation of scoring model parameters; matching ranks. 59
3.3 Examples of Gamma distributions. ................................. 60
3.4 Example of score computation .................................... 62
3.5 ROC curves of four comparison cases ............................. 66
3.6 Clustering analysis of a test set using X-Rank. .................. 70
3.7 Two cases of peculiar scoring model behavior. .................. 71
3.8 Molecular structures and spectra of two close benzodiazepines. 72
3.9 Distribution of the percentage of shared peaks within spectra
of a given compound. ............................................. 73
3.10 Construction and use of a consensus signature. ................. 74
3.11 Decision tree configuration. ....................................... 76
List of Figures

4.1 Library layer system enhancing multiple applications libraries. 83
5.1 Fragmentation spectra of nicotine acquired under increasing fragmentation energy. 84
5.2 Fragmentation graph of nicotine. 85
5.3 Fragmentation graph of nicotine with annotated intensities of fragments. 86
5.4 Evolution of selected peak intensities in relation to fragmentation energy. 87
5.5 Basic molecular visualization tool. 88
5.6 Fragmentation graph tool. 89
5.7 Connectivity matrix of nicotine. 90
6.1 Dry-lab and Wet-lab cohabitation. 91
6.2 The waterfall model. 92
6.3 The iterative development model. 93
6.4 The spiral model. 94
6.5 High-level representation of SmileMS platform. 95
6.6 The model-view-controller pattern. 96
6.7 Interactions between the interface and the web service layer during the submission process. 97
6.8 Sequence diagram of the identification process business logic layer. 98
6.9 Data structure of spectra and libraries. 99
6.10 Data structure of chemical compound entities. 100
6.11 XML structure SmileMS results. 101
6.12 Data structure of users and their privileges. 102
6.13 Representation of a Data Access Object (DAO). 103
6.14 Examples of Flex development. 104
6.15 Spring dependency injection. 105
6.16 Aspect Oriented Programming (AOP). 106
7.1 The five steps of data processing. 107
7.2 Noise reduction by peaks filtering. 108
7.3 The three steps of the aggregation process. 109
7.4 SmileMS visualization workflow. 110
7.5 SmileMS library enrichment workflow. 111
7.6 The library management dedicated module. 112
7.7 Bruker Daltonics integration workflow. 113
8.1 Schematic representation of the HUG analysis workflow. . . 160
8.2 MRM versus GUS acquisitions. . . . . . . . . . . . . . . . . . 165
8.3 Spectral match proposed by Cliquid. . . . . . . . . . . . . 170
8.4 Spectrum of the missing temazepam molecular compound. . 172
List of Tables

1.1 Main methodologies operated in clinical environments. . . . . 23
1.2 Main chromatographic methods used for HPLC. . . . . . . . 27
1.3 Fragmentation types that apply to LC-MSMS. . . . . . . . 32
1.4 Mass analysers and their performance . . . . . . . . . . . . . 33
1.5 Instrument acquisition software from major MS manufacturer. 36
1.6 Software platforms used in the field of small molecule iden-
     tification. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 38
1.7 Algorithms used for library search in GC-MS and LC-MSMS. 39
1.8 LC-MSMS libraries and information regarding their access,
     thematic, and size. . . . . . . . . . . . . . . . . . . . . . . . . 41

3.1 Four comparative tests assessing X-Rank performance. . . . . 63
3.2 Description of a publicly available test mix for drug analysis. 65
3.3 Comparison of SmileMS and MS Search for a test mix identi-
     fication. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 68
3.4 Score densities for a clear and an ambiguous identification. . 73
3.5 Decision tree classifier performance. . . . . . . . . . . . . . 75
3.6 Neural network classifier performance. . . . . . . . . . . . . 76
3.7 SVM classifier performance. . . . . . . . . . . . . . . . . . . 77
3.8 Improvement in false negative and false positive rates com-
     pared to X-Rank only. . . . . . . . . . . . . . . . . . . . . . . . 78

7.1 Major file formats currently supported by SmileMS. . . . . . 146

8.1 SmileMS identification of a blood sample using two different
     instruments. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 163
8.2 Compounds differentially identified between the AmaZon and
     the QTRAP. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 164
8.3 Results of SmileMS for a MRM acquired test sample. . . . . 168
8.4 Results of Cliquid for a MRM acquired test sample. . . . . . 168
Chapter 1

Introduction

Contents

1.1 From biology to bioinformatics .......................... 16
  1.1.1 Biological Sciences and the art of observation 16
  1.1.2 Bioinformatics to cope with complexity .... 17
  1.1.3 Mass Spectrometry; a bioinformatics early adopter ............ 17

1.2 Clinical toxicology: a story of toxics .................... 20
  1.2.1 Towards a classification of diseases ........... 20
  1.2.2 Clinical toxicology; a division of toxicology ... 21

1.3 Current screening instrumentations ................... 22
  1.3.1 Screening techniques in clinical toxicology . 22
  1.3.2 Liquid Chromatography Tandem Mass Spectrometry (LC-MSMS) (+) ............ 26
  1.3.3 LC-MSMS in clinical environment ............ 34

1.4 Current instrument software solutions (+) ............ 34
  1.4.1 Instrument acquisition (+) ................. 35
  1.4.2 Library search identification (+) ............ 37
  1.4.3 Chemical fragmentation applications ....... 41
  1.4.4 Software engineering and design .......... 44

1.5 SmileMS: a small molecule identification platform .... 44
CHAPTER 1. INTRODUCTION

1.1 From biology to bioinformatics

1.1.1 Biological Sciences and the art of observation

From the study of Natural History in the 18th and 19th centuries to the modern era of Omics [Kan09], biology has encountered amazing changes [Sap03]. Interestingly, past centuries can each be assigned a new analytical technique that pushes previous boundaries a little further. The 18th century is thus characterized by the use of magnifiers and focuses on organs. With the 19th century, the use of microscopes enables scientists to observe cells. The next century introduces electronic microscopes and opens the door to the observation of molecules. Finally, the 21st century sees the integration of the tree levels: organs, cells and molecules.

Whereas biologists were supposed to master the art of observation in the past, they are now assisted by complex analytical devices. This ubiquitousness of instruments in modern laboratories is a dominant characteristic of current research. The new assistance provided by instruments covers a large range of applications and has permitted countless major breakthroughs. Specifically, recent equipment has opened new perspectives by enabling scientist to access ever more complex and previously unknown systems. Traditional laboratory experiments coupled with computational resources have seen the emergence of system biology ([Car07] is an example of such combination of approaches). This marks the new possibility to consider biological systems as a whole, rather than as a multitude of independent micro-mechanisms.

On the other hand, the technicisation of laboratories is often associated with manipulation complexity and a constantly growing amount of produced data. These data are also more elaborated. Molecular biology is a dramatic example of data complexity and growth [Mar01]. More than 1500 genomes have been sequenced since the 1990’, with sequencing centers able to generate tens of terabytes of data, daily. The cost, in terms of money and time, also drastically diminished. While it took years and hundreds of millions dollars to sequence the first genomes, these figures are reduced to weeks and thousands of dollars nowadays [Drm10, Lan01].

Today, molecular biology is no more an isolated example. Environmental or evolution studies see the same trend in terms of data generation. Fast-paced production of data, as well as complexification of these data have to be addressed. In that regards, bioinformatics has become an omnipresent tool throughout life science applications.
1.1.2 Bioinformatics to cope with complexity

The term bioinformatics was first mentioned in 1978 by Paulien Hogeweg and Ben Hesper, who “identified the study of informatic processes in biotic systems as an open and promising research area” [Hog10]. Bioinformatics can be defined as the application of computer science and allied technologies to help biologist investigations [Nai07, Cla03]. This rather new field of the Biology is an interdisciplinary field\(^1\) that deals with areas such as algorithmic, statistic modelization, and data management [Bay03].

The issue published on March 29th 2004 of “Just the Facts: A Basic Introduction to the Science Underlying NCBI Resources” highlights two important benefits of bioinformatics; a more global perspective in experimental design, and the ability to capitalize on the emerging technology of database-mining [NCB04]. Particularly, bioinformatics focuses on handling modern instrument data complexity and aims at dealing (i.e. storing, structuring, processing, and so on) with large amount of data.

Today, bioinformatics plays a major role in numerous biology applications [Kap04]. A recent literature survey [Koh09] stressed two fields as particularly concerned: proteomics [Bog03] and genomics [Iqb06]. With the human genome project for example [Gre05], bioinformatics began to find more and more clinical applications [Bay03]. However, bioinformatics in clinical environment does not only concentrate on gene- or protein-centric applications. Drug design and metabolomics are two other instances of medical applications that are approached by bioinformatics.

Along with its application-specific uses, bioinformatics also plays a more transversal role by supporting technologies that can be employed in multiple fields. Microarray and mass spectrometry are two technologies that traditionally generate important quantities of information and that have been adopted by bioinformatics very early.

1.1.3 Mass Spectrometry; a bioinformatics early adopter

Mass Spectrometry (MS) is an analytical methodology that aims at accurately measuring mass over charge (m/z) ratios of ionized molecules. Samples are injected into a mass spectrometer, which measures spectra intensities and m/z values (Figure 1.1).

---

\(^1\)Bioinformatics can be considered at the crossings of computer science, mathematics, physics, and biology [Bay03, NCB04].
Mass spectrometry is used for both routine and research purposes. Examples of spectrometry application can be found in:\footnote{Other applications can be found in [Ash09].}

- **Biotechnology**, for the analysis of proteins, peptides, or oligonucleotides (Figure 1.2 D),
- **Pharmaceutical industry**, for drug discovery, combinatorial chemistry, pharmacokinetics, or drug metabolism analysis,
- **Clinic**, for neonatal screening, haemoglobin analysis, or drug testing,
- **Environment**, for PAHs, PCBs, water quality, or food contamination (Figure 1.2 C),
- **Geology**, for the assessment oil composition,
- **Military**, for the detection of toxics during military operations,
- **Forensics**, for the study of drivers under drug influence,
- **Homeland security**, for explosive detection (Figure 1.2 A),
• **Anti-doping**, for the detection of doping products,

• **Space industry**, for the analysis collected material (Figure 1.2 B).

Figure 1.2: selected applications of mass spectrometry. A: Guardian Explosives Trace Portal produced by Syagen. This new generation security portal includes a TOF mass spectrometer. B: Close-up of the GC mass spectrometer embarked on Phoenix Mars mission. C: sensor provided by Harbour Mastery and used to monitor water quality. This sensor is equipped with multiple communication and analysis systems, as well as a mass spectrometer. D: Mass spectrometer employed by the Proteomics Core Facility of the University of Geneva, for the study of proteins.

Progress in accuracy, rapidity, sensitivity and specificity of MS instruments makes this technology suitable for an always growing pool of applications.

As previously mentioned, mass spectrometry properly illustrates the current need for bioinformatic resources [Kum09, Deu08]. Bioinformatics
has become critical to mass spectrometry, due to the amount of data produced [Cri04]. Modern spectrometers can generate thousands to hundreds of thousands of fragment ion spectra per hour of data acquisition [Nes07]. Each of these spectra has then to be analyzed and assigned a molecule. This task was formerly done by human experts through spectra interpretation. Nowadays, this process is achieved by computers and asks for storage capabilities and computing resources. Furthermore, the complexity generated by mass spectrometry is no more treatable by manual spectral annotation [Kea03]. Finally, multiple post-acquisition challenges arise from MS. Noise filtering, deisotopisation, deconvolution, peak detection, identification, and finally quantification are among the most obvious of these challenges. As a result, the past ten years have seen the development of multiple algorithms that try to decipher information hidden behind the complexity of fragmentation spectra [Deu08, McLan].

In that sense, interactions between bioinformatics and MS have contributed to dramatically accelerate bioinformatics headways in algorithmic and data management. The pair bioinformatics/mass spectrometry is now considered reliable, and is increasingly present in routine applications. Clinical toxicology is a good example for which bioinformatics serves MS on a daily basis.

1.2 Clinical toxicology: a story of toxics

1.2.1 Towards a classification of diseases

The Journal of Proteome Research recently exposed that human diseases should be classified as being due to two main factors: our genes and our environment [Hoc08]. These two causes are however not always exclusive. They can also be cumulated. Type II diabetes is an example of a cumulation of these two factors. While diet and physical activity are known to play an important role, genetic causes are also designated as influencing factors [Pra98].

In the category of diseases provoked by our environment, two subcategories can again be distinguished: first, diseases caused by microbes and second, diseases provoked by chemical substances. The former field is covered by microbiology, whereas the latter is addressed by toxicology.

This classification of human diseases is represented in Figure 1.3.
1.2.2 Clinical toxicology: a division of toxicology

In the field of clinical toxicology, a sub-division of toxicology dealing with medical aspects, intoxications and adverse reactions are two important factors of hospitalization. In Switzerland for instance, the number of intoxication incidents was about 24’000 in 2003. In addition, two-third of reported poisonings, whether voluntary or not, were due to drug consumption [Stu05]. Because of this high prevalence, and because it is now common for every emergency service to routinely include drug screening tests before admission [Brv09, Sat09, Kon08], intoxications, adverse reactions, and medical toxicology in general, benefit from very active research.

Figure 1.3: a classification of human diseases. Human diseases are due to our genes or to our environment. Environment diseases can again be separated into two sub-categories; first the diseases caused by microbes and second the ones provoked by chemical substances. (Figure adapted from [Hoc09])

Many classes of molecules are of interest for intoxications and adverse reaction investigations. These classes usually account for several hundreds of compounds. Among them, antidepressant [Ree09], antipsychotic [LA09], and benzodiazepine [O’b05, Buc95] are frequently cited. At the Geneva University Hospitals (HUG), the class of benzodiazepine alone accounts
for a huge percentage of daily analyses. In 2008, it has been reported that almost 80% of toxicological analyses confirmed the presence of this class of molecules. Benzodiazepine is a family of chemical compounds that includes drugs such as diazepam (Valium), chlordiazepoxide (Librium), oxazepam (Serax), lorazepam (Ativan), alprazolam (Xanax), and clonazepam (Clonopin). Some, such as flurazepam (Dalmane), alprazolam (Xanax) and triazolam (Halcion), are commonly used as sleeping aids. Besides their important prevalence, benzodiazepines are difficult to detect at low concentration, depending on the used analytical instrumentation [Val96]. This example illustrates the need to carefully select screening instruments.

For a routine use, clinical environment imposes strict criteria for this selection. These criteria are based on cost, robustness, speed, simplicity of use, genericity and the variety of detected drugs [Vie10]. The relative importance of these criteria varies depending on the context of application. For example, the speed of analysis is crucial in emergency medicine, whereas a thorough detection range will be plebiscited in forensic applications. From another point of view, an extremely low false positive rate is imperative for clinical application, even if it is at the cost of a higher false negative rate.

1.3 Current screening instrumentations

1.3.1 Screening techniques in clinical toxicology

Multiple instrumental settings meet at least a subset of the criteria imposed by clinical laboratories. Table 1.1 reads the main methodologies routinely employed in medical applications to detect and identify chemical compounds. This table also highlights their main advantages and drawbacks.

While some medical institutes adopt a combination of such methods to cover their needs, it can be time-consuming to combine multiple analyses, increasing the number of technical steps to achieve a full drug screening. In addition, this approach requires expertise in an array of technologies. Consequently, screening analyses are often restricted to a subset of compounds, with additional investigation only in case of peculiar suspicion.

As described in the following paragraphs, the identification of benzodiazepines is once more an instructing example that enhances strengths and

---

3Names given between parentheses correspond to drug names.
4Usually expressed as turnaround time (TAT).
### 1.3. CURRENT SCREENING INSTRUMENTATIONS

<table>
<thead>
<tr>
<th>Instrumental setting</th>
<th>Advantages</th>
<th>Drawbacks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Immunoassays</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[Fer94] [Moe08]</td>
<td>a) Rapid b) Easy to use</td>
<td>a) Limited to specific classes of molecules b) Antigen-antibody reaction not always specific</td>
</tr>
<tr>
<td><strong>LC-UV</strong> (Remedi [Bin89])</td>
<td>a) Easy to use b) Wide panel of compounds included in shipped library c) plug and play system</td>
<td>a) Support from Bio-Rad discontinued b) Not very sensitive for specific classes of molecules</td>
</tr>
<tr>
<td><strong>GC-MS</strong> [Mau04] [Vog08] [Moe08]</td>
<td>a) Very large and machine independent libraries b) Reference methods exist c) High sensitivity and specificity</td>
<td>a) Tedious sample preparation b) No possible automation c) Limited to specific classes of molecules</td>
</tr>
<tr>
<td><strong>LC-MSMS</strong> [Vog08]</td>
<td>a) High sensitivity and specificity b) Possible automation c) Rapid</td>
<td>a) Requires substantial expertise and know-how b) Instrument dependent libraries c) Matrix effect</td>
</tr>
</tbody>
</table>

1 Liquid Chromatography Ultra-violet 2 Gaz Chromatography Mass Spectrometry 3 Liquid Chromatography Tandem Mass Spectrometry

Table 1.1: main methodologies operated in clinical environments.

weaknesses of the presented technologies.

**a) REMEDI**

The REMEDI instrument was traditionally used to achieve routine drug screening for patient admitted in hospital emergency units. This instrument is a LC-UV drug screening system recognized for its ease of use and the fact that it is sold as a complete solution (i.e. including reagents, a spectral library, and some analysis guidelines).

However, REMEDI machines have been announced to be discontinued
by the end of 2009. Moreover, these machines suffer from a limited range of detection and a lack of sensitivity for certain substances. Benzodiazepines, among others, are poorly detected by the REMEDi, due to a detection limit unadapted to medical screening.\(^5\)

b) **Immunoassay (\(\ast\))**

Immunoassay is a biochemical technology based on the interactions between antibodies and antigens. Immunoassay tests are usually complementary to other approaches (such as previously described REMEDi) or used to achieve targeted analyses. They present the advantage of being largely automated and fast (with a turnaround time lower than one hour) [Ten09]. Furthermore, they do not require high skills to be operated.

On the other hand, two main drawbacks of this technology are its limitation in terms of detection range and its lack of specificity for certain reactions. This last point appeared to be the source of important errors, with for example, the reported admission of a child for maltreatment, due to an improper identification of methadone instead of diphenhydramine [Rog07]. Taking the example of benzodiazepines, immunoassays allows to identify the class of molecules but not specific members of this class. Finally, in addition to these two limitations, immunoassays tend to be very expensive [Vog08].

c) **Gas Chromatography Mass Spectrometry (GC-MS) (\(\ast\))**

GC-MS is another alternative to poorly detected chemical compounds. GC-MS methodologies present the tremendous advantage of being well established and considered robust, mainly thanks to canonical experimental settings. This robustness is often attributed to the high sensitivity and specificity of this technique.

However, just like previously described methods, GC-MS caries its own limitations. Among other disadvantages, it involves the extraction and derivatization of analytes and has a limited detection range of molecules with respect to their size and type. Current software systems also imply an extensive manual work during the interpretation of spectra.

---

\(^5\)Examples: Triazolam (detection limit: 46.5 \([\mu g/l]\), therapeutic concentration: 2-20 \([\mu g/l]\), toxic concentration: 40 \([\mu g/l]\)), Olanzapine (detection limit: 306 \([\mu g/l]\), therapeutic concentration: 20-30 \([\mu g/l]\), toxic concentration: 200 \([\mu g/l]\)), Clonazepam (detection limit: 135 \([\mu g/l]\), therapeutic concentration: 10-80 \([\mu g/l]\), toxic concentration: 100 \([\mu g/l]\))
1.3. CURRENT SCREENING INSTRUMENTATIONS

(4) Liquid chromatography tandem mass spectrometry (LC-MSMS)

Finally, liquid chromatography tandem mass spectrometry (LC-MSMS) arose as a valuable technique in terms of rapidity and range of detected substances. This technique equals and sometimes even outperforms GC-MS systems in terms of sensitivity and specificity. Besides, LC-MSMS benefit from potential automation, which dramatically diminishes its turnaround time.

Although the idea of applying LC-MSMS to drug screening is not new [Dru99], this technique still suffers from inconveniences. One important disadvantage of LC-MSMS, which is also one of its advantage paradoxically, is the diversity of existing instruments and workflows. This diversity makes LC-MSMS instrumentations difficult to properly master. For instance, ionization and fragmentation sources can vary from one instrument to another, leading to a variability that is difficult to handle while identifying generated spectra against a reference library. Figure 1.4 shows the variability of caffein spectra generated by LC-MSMS compared to an absence of variability using GC-MS.

Figure 1.4: (*) LC-MSMS generated spectra (left) show important variability compared to spectra generated by GC-MS (right). Presented spectra were retrieved from MassBank (http://www.massbank.jp/).
Furthermore, with a lighter sample preparation compared to GC-MS, LC-MSMS identification can be affected by matrix effects\(^6\). This artifact can be highly harmful for systematic toxicological analyses [Mül02], although it can be reduced by adapted sample preparation.

LC-MSMS technique was chosen to be the successor of the REMEDI at the University Hospital of Geneva (HUG). This technique is further detailed in the following sections.

1.3.2 Liquid Chromatography Tandem Mass Spectrometry (LC-MSMS) (*)

A LC-MSMS experiment involves four main steps (Figure 1.5):

1. a sample extraction,
2. a liquid chromatography (LC) separation,
3. a MSMS analysis,
4. and a computer analysis.

The three last steps of this process are described in the following paragraphs. Sample extraction is not covered in this document.

a) Liquid chromatography (*)

A chromatography is a physical separation method. In the scope of LC-MSMS experiments it is a first step separation ensuring that molecules do not arrive in the mass spectrometer at the same moment. In a liquid or gaseous chromatographic separation, substances are selectively separated thanks to two immiscible phases; a mobile phase and a stationary phase [Nie06]. Whereas the mobile phase is gaseous in gas chromatography, it is liquid in the case of liquid chromatography. Due to analytes with identical or barely distinguishable molecular mass (i.e. molecules or fragment that will not be separated by the next analytical step; the MSMS analysis), it is crucial to achieve good chromatographic performance.

There exist a number of different techniques for liquid chromatography (Table 1.2). Small molecules are usually well separable by their differing

\(^6\)Matrix effects are the alteration of ionization efficiency by the presence of co-eluting substances.
1.3. CURRENT SCREENING INSTRUMENTATIONS

Figure 1.5: LC-MSMS experiments involve four steps: a sample extraction, a LC separation, a MS analysis, and a computer analysis.

polarity [Man08]. In this case the chromatographic method of choice is the Revers Phase Chromatography, but there exist studies using alternative chromatographic methods such as HILIC [Tol02].

<table>
<thead>
<tr>
<th>Chromatographic method</th>
<th>Separation technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal phase (NP-HPLC)</td>
<td>polar differences</td>
</tr>
<tr>
<td>Reverse phase (RPC)</td>
<td>polar differences</td>
</tr>
<tr>
<td>Hydrophilic Interaction (HILIC)</td>
<td>polar differences</td>
</tr>
<tr>
<td>Size exclusion (SEC)</td>
<td>molecule size</td>
</tr>
<tr>
<td>Ion exchange</td>
<td>molecular charge</td>
</tr>
<tr>
<td>Bioaffinity</td>
<td>complex building</td>
</tr>
</tbody>
</table>

Table 1.2: main chromatographic methods used for HPLC.

Reversed phase columns consist of a non-polar stationary phase. C18 bonded silica is the most popular type of reversed-phase HPLC packing [Man08]. The mobile phase usually consists in an aqueous blend of water with a miscible and polar organic solvent such as acetonitrile or methanol.
In this system, polar molecules such as acids will elute first, while the non-polar compounds will elute later (Figure 1.6). When eluting, substances are injected into the mass spectrometer through the ion source.

Figure 1.6: the reversed phase column uses a non-polar stationary phase and a polar mobile phase for separation. The polar compounds elute first.

b) Mass spectrometer architecture (∗)

A mass spectrometer (Figure 1.5, labeled "Mass Spectrometer") consists of three pieces:

1. an ion source,
2. one or more mass analyser(s) coupled with a fragmentation unit,
3. and a detector.

When an analysis is conducted, samples are first ionized in the ion source. Ions are then separated according to their m/z ratios in the analyzer. Ions selected by the last analyzer (in the case where multiple analyzers are chained) are then detected by the detector. Finally, the generated signal is analyzed by a computer [Lan05].

Within the LC-MS family\(^7\), many types of instrument exist. Yet, two categories enable MSMS experiments. The first one is the combination of

\(^7\)LC-MS is here considered a family to which LC-MSMS belongs.
two or more mass spectrometer subparts, while the second one is the use of analyzers capable of storing ions [Hof96].

To further classify LC-MSMS instruments, three features are to be considered: the ionization device, the fragmentation device, and the analyzer. The detector could be added to this list, though it tends to have less influence on produced spectra than the other elements.

**Ionization devices (⋆)**

Ionization devices play a central role in the analysis of chemical compounds. The ionization process adds or removes charge(s) to or from the molecule. Only ionized compounds are screened by the analyzer.

Figure 1.7 highlights the most used ionization types. As described in this figure, each types of ionization is efficient for a specific range of molecules. This range is determined by the ionic character of molecules, as well as by their mass. For instance, electrospray ionization (ESI) is more efficient for polar substances [Cec01, Agi01], while atmospheric pressure chemical ionization (APCI) targets less polar molecules, with a mass lower than $10^3 [Da] - 10^4 [Da]$ [Agi01]. Moreover, volatility of substances also plays a role in the ionization process. Chemical ionization, for example, is limited to volatile compounds, whereas fast atom bombardment (FAB), APCI, and ESI can be applied to non-volatile compounds as well [Hof96].

In practice, APCI and electrospray ionizations seem to be the most frequently adopted techniques.

**Fragmentation devices (⋆)**

The role of the fragmentation device is to break molecules into sub-structures. While non-fragmented molecules ($MS_1$ level) inform about the mass of a chemical compound, each consecutive fragmentations (i.e. producing $MS_n$ level spectra) increases the amount of structural information [Hou00].

Besides the technique and level of fragmentation, the total amount of transferred energy is of major interest. Higher energy increases the number of fragmentation events, and thus the number of fragment ions. This is due the the increase of internal energy of the precursor ion, which leads to electronic and vibrational excitation of the ion [Gab05]. On the other hand, lower energies seem to increase the number of rearrangements [Hof96, McLan]. In reality, these two claims are not so contrasted, since rearrangements and cleavages follow a not-well understood equilibrium [Gab05] that is controlled by the internal energy, and probably less known factors.
CHAPTER 1. INTRODUCTION

Figure 1.7: (*) each type of ionization is efficient for a specific range of molecules. This range is determined by the ionic character of molecules, volatility, and mass.

Table 1.3 summarizes the most common fragmentation techniques. In addition to these methods, fragmentation can occur during the ionization process itself. This type of ionization is called in-source fragmentation and is used for single-stage mass spectrometry, in a controlled manner.

Analyzers (*)

The analyzer is the third piece of hardware that plays an important role in the MSMS mechanism. The role of analyzers is to separate ions according to their m/z ratio. Table 1.4 compiles available mass analyzers and their individual performance\(^8\) in terms of:

- resolution,
- accuracy,
- mass range,
- dynamic range,

\(^8\)Definitions of performance terms are given in Table 1.4.
• speed.

As aforementioned, an important characteristic is the possibility to achieve $MS_n$ experiments. Some analyzers, such as ion traps, directly enable $MS_n$, whereas other needs to be chained. This is the case for the quadrupole mass filters for instance.

c) Computer analysis (∗)

Data acquired and produced by mass spectrometers are treated by computer systems from the acquisition, to the signal extraction and the identification. This domain will be more thoroughly represented in the following section 1.4.
<table>
<thead>
<tr>
<th>Fragmentation Method</th>
<th>Energy Range</th>
<th>Instruments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collision-induced dissociation (CID)</td>
<td>High, Low</td>
<td>IT, QqQ, FTICR</td>
<td>Collision-induced dissociation by collision of precursor ions by multiple inert target gas molecules in the collision cell. Energy range 1 - 100 eV.</td>
</tr>
<tr>
<td>Surface induced dissociation (SID)</td>
<td>High</td>
<td>Tandem TOF, IT</td>
<td>Same as above with precursors of higher translational energies (instrument dependent). Surface induced dissociation (SID).</td>
</tr>
<tr>
<td>Blackbody infrared radiation dissociation (BIRD)</td>
<td>Low</td>
<td>IT, FTICR</td>
<td>Low-energy thermal activation method ideal for calculations of energy thresholds. Blackbody infrared radiation dissociation (BIRD).</td>
</tr>
<tr>
<td>Electron capture dissociation (ECD)</td>
<td>Low</td>
<td>FTICR</td>
<td>Low-energy beam of electrons resulting in electron capture at protonation (or cationic) site with subsequent fragmentations. Electron capture dissociation (ECD).</td>
</tr>
<tr>
<td>Electron transfer dissociation (ETD)</td>
<td>Low</td>
<td>IT, FTICR</td>
<td>Employs radical anions to transfer electron. Energetic beams of electrons resulting in electron transfer to ions. Electron transfer dissociation (ETD).</td>
</tr>
<tr>
<td>Surface induced dissociation (SID)</td>
<td>High</td>
<td>Tandem TOF, FTICR</td>
<td>Same as above with precursors of higher translational energies (instrument dependent). Surface induced dissociation (SID).</td>
</tr>
<tr>
<td>Fragmentation types that apply to LC-MSMS</td>
<td></td>
<td></td>
<td>Table 1.3: Fragmentation types that apply to LC-MSMS. (Figure adapted from Sle04)</td>
</tr>
</tbody>
</table>
1.3. **CURRENT SCREENING INSTRUMENTATIONS**

<table>
<thead>
<tr>
<th></th>
<th>TOF</th>
<th>Sector magnet</th>
<th>Quadrupole</th>
<th>Ion trap</th>
<th>FTICR</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mass resolving power</strong></td>
<td>$10^3 - 10^4$</td>
<td>$10^2 - 10^6$</td>
<td>$10^4 - 10^5$</td>
<td>$10^5 - 10^4$</td>
<td>$10^4 - 10^6$</td>
</tr>
<tr>
<td><strong>Mass accuracy</strong> [ppm]</td>
<td>5-50</td>
<td>1-5</td>
<td>100</td>
<td>50-100</td>
<td>1-5</td>
</tr>
<tr>
<td><strong>Mass range</strong> [Da]</td>
<td>&gt; $10^5$</td>
<td>$10^4$</td>
<td>$10^3$</td>
<td>$1.5 \times 10^5$</td>
<td>&gt; $10^4$</td>
</tr>
<tr>
<td><strong>Linear dynamic range</strong></td>
<td>$10^2 - 10^6$</td>
<td>$10^5$</td>
<td>$10^4$</td>
<td>$10^2 - 10^5$</td>
<td>$10^2 - 10^5$</td>
</tr>
<tr>
<td><strong>Speed</strong> [Hz]</td>
<td>$10^1 - 10^4$</td>
<td>0.1-20</td>
<td>1-20</td>
<td>1-30</td>
<td>0.001-10</td>
</tr>
</tbody>
</table>

1. Measure of the ability to distinguish two peaks of slightly different m/z.
2. Ratio of the m/z measurement error to the true m/z.
3. Range of m/z amenable to analysis by a given analyzer.
4. Range over which ion signal is linear with analyte concentration.
5. Time frame of the experiment and ultimately is used to determine the number of spectra per unit time that can be generated.

Table 1.4: available mass analyzers and their individual performance. (See the glossary for more detailed definitions of performance terms.)
1.3.3 LC-MSMS in clinical environment

From the early general unknown screening (GUS\textsuperscript{9}) developments [Wei01, Ven03] to the more recent targeted analysis techniques (such as Multiple Reaction Monitoring (MRM\textsuperscript{10})) [Ger03], LC-MS and then LC-MSMS have proved to be the most valuable alternative among the previously described instrumentations. Though LC-MSMS suffers from drawbacks, it still efficiently addresses the needs for wide detection range, for speed (thanks to automation possibilities), and for specificity and sensitivity.

In the domain of clinical analyses, the broad LC-MSMS detection range can be illustrated by comparing it with other techniques. While the REMEDi system shows low sensitivity and immunoassays low specificity [Mar09], LC-MSMS technology seems to cope well with almost all kind substances. For example, numerous LC-MSMS methods show good identification results for the detection and identification of benzodiazepines [Mar08, Ngw07], which was previously presented as an issue, employing former instrumentations.

To assess the validity of using LC-MSMS as the successor of the REMEDi, Figure 1.8 presents compared performance of a REMEDi machine and triple quadrupole instrument\textsuperscript{11}. This figure shows that most of the substances detected by the REMEDi are also detected by LC-MSMS. It also shows that some substances were detected by LC-MSMS, but not by the REMEDi. This figure will be further detailed in chapter 8.

Finally, when compared to GC-MS systems, LC-MSMS provides a way to detect additional substances. For instance, nonvolatile or thermally labile substances (such as polar and high mass substances) are poorly or never detected by GC-MS analyses, while these substances are perfectly suitable for LC-MSMS.

1.4 Current instrument software solutions (∗)

Along with the increasing number of LC-MSMS applications, the number available software raised to an unmanageable array of tools that are often redundant, but usable for specific experimental designs only. The follow-

\textsuperscript{9}A general unknown screening (GUS) is a drug screening for which no a priori, in terms of substance desired to identify, has been decided.

\textsuperscript{10}MRM is a MS technique that allows to a priori select substances of interest.

\textsuperscript{11}This instrument and its use at the University Hospital of Geneva will be further described in section 8.2.1
1.4. CURRENT INSTRUMENT SOFTWARE SOLUTIONS (*)

Figure 1.8: analysis of 237 real sample between a REMEDI system and a LC-MSMS. (Courtesy of Veronique Viette)

ing sections contain a brief overview of the main software applications in the field of small molecule identification. Beside the described commercial solutions, there exist several open source tools and attempts to offer unified cross-vendors further functionalities [Stu08, Col06, Smi06].

1.4.1 Instrument acquisition (*)

Every MS-instrument vendor provides its own specific machine acquisition software. Even for one constructor, this software is at least partly different from one machine to another. The main functionalities of these software applications include:

- Control data acquisition
- Control machine parameters
- Store acquired raw data
- Visualize acquired data

Beyond these basic functionalities, most vendors extend their software by interacting with other applications. This is the case for the integration
of the platform we present in section 7.4. External software is either developed internally or by third party software developers. A big challenge for these integrations is the transfer of data from one software to the other. To facilitate this process, ongoing effort to create data standards is crucial.

With regards to LC-MSMS standards, raw formats can be transformed into a generic XML format such as mzXML [Ped04], or its recent successor mzML [Kel05]. These formats are widely supported by MS related softwares, though there remain some major issues. One problem is that mzXML and mzML were initially created for proteomics. As a result, the design of these formats is not always adequate for LC-MSMS applications, such as small molecules identification.

Another downside of the transfer of data is the manual processing steps it imposes to users. To answer that issue, a common choice of software vendors is to directly implement bridges from the acquisition software to other tools, so that the process of transferring data can be hidden.

A list of instrument acquisition software of the major MS instrument vendors and their corresponding raw data formats is presented in Figure 1.5.

<table>
<thead>
<tr>
<th>Vendor</th>
<th>Acquisition Software</th>
<th>Raw format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thermo</td>
<td>XCalibur</td>
<td>.RAW file</td>
</tr>
<tr>
<td>Waters</td>
<td>MassLynx</td>
<td>.RAW directory</td>
</tr>
<tr>
<td>Waters</td>
<td>Analyst</td>
<td>.Wiff file</td>
</tr>
<tr>
<td>Waters</td>
<td>Mass-Hunter</td>
<td>.D directory</td>
</tr>
<tr>
<td>Agilent</td>
<td>CompassOpenAccess</td>
<td>.Baf</td>
</tr>
<tr>
<td>Bruker</td>
<td>Dependent on the instrument</td>
<td>lcd</td>
</tr>
</tbody>
</table>

Table 1.5: list of instrument acquisition software from major MS instrument vendors and their corresponding raw data formats.

In this context of multiple proprietary formats, formats handling is a technical but crucial aspect. Isolated format readers exist, but Proteowizard is an initiative that seems to cover a broader area of standards and file formats. Proteowizard\textsuperscript{12} is a modular set of open source, cross-platform tools and software java libraries for proteomics data analysis [Kes08]. Particularly, it addresses the question of file format diversity and proposes open-source readers.

\textsuperscript{12}\url{http://proteowizard.sourceforge.net/}
In complement to *Proteowizard*, the effort we dedicated to address formats support, as well as the production of an open-source software addressing format issues, are discussed in section d).

### 1.4.2 Library search identification (*)&

For the identification of chemical compounds, a library search approach is usually chosen. In this technique, experimental spectra are compared to a reference library. There are two approaches for library search; either the experimental spectra are selected manually and searched against a library or all the experimental spectra from a run are searched against a library and a result containing all the matches is generated. We will call this methods "manual library search" and "automated library search", respectively.

Table 1.6 shows commonly used library search software. *MS Search*, from the National Institute of Standards and Technology (NIST), is to our knowledge the only widely used MS instrument constructor independent solution. However, *MS Search* algorithm is used in several commercial software application such as *ToxID* from Thermo Scientific.

To fulfill general needs, identification software has to fulfill three main functionalities:

1. library search,
2. result visualization,
3. and library management.

The realization of these functionalities is more or less developed, depending on software. The result visualization for instance can be a very limited pdf printout\textsuperscript{13} on the one hand, or a more complex user interface on the other end \textsuperscript{14}. The same applies to library management: this aspect can be very basic, such as indicating a file to use as library, or more complete, with the possibility to create, enrich and edit multiple libraries.

Section 7 of this thesis presents our approach towards these functionalities.

\textsuperscript{13}This is the case for *ToxID* from Thermo Scientific.

\textsuperscript{14}The identification platform presented in this thesis belongs to this category.
Table 1.6: major software platforms used in the field of small molecule identification.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Instrument control</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Distant access</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Quantitation</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Manual library search</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Automated library search</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Library management</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>User management</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>High mass accuracy data</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

a) Library search identification algorithms (*

The manual interpretation of mass spectra is a tedious work that requires skills acquired only by decades of experience [McL93]. While forty years
ago instruments could only acquire a few spectra per hour, modern machines are orders of magnitude faster. Nowadays, the huge amount of produced data marks the end of routine manual interpretation. Even though this approach remains useful in the case of ambiguous identifications, fully automated approaches are the preferred method for routine compound screening.

As previously mentioned, an automated identification can be done by comparing experimentally acquired fragment spectra to annotated spectra libraries. This approach is commonly called library search. There exist a number of library search algorithms (Table 1.7). In the literature, the most cited ones are the PBM and the Dot Product. Mainly adapted to GC-MS data, one or the other of these algorithms seems alternatively to perform better, depending on the study. Since the PBM algorithm is based on GC-MS statistical observations, it has rarely been tested on LC-MSMS data.\(^\text{15}\)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Implementation</th>
<th>Application</th>
<th>Literature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distance Measures</td>
<td>No known</td>
<td>GC-MS, LC-MSMS</td>
<td>[Lay05], [Her71], [Ste94], [Wanan]</td>
</tr>
<tr>
<td>PBM</td>
<td>Palisade Corporation</td>
<td>GC-MS</td>
<td>[McL74b], [McL74a], [Pes75], [Mclan]</td>
</tr>
<tr>
<td>Share peak count</td>
<td>No known</td>
<td>GC-MS, LC-MSMS</td>
<td>[Sin04]</td>
</tr>
<tr>
<td>Dot Product</td>
<td>MS Search (NIST), INCOS, SpectraST</td>
<td>GC-MS, LC-MSMS</td>
<td>[McL74b], [Ste94], [Wanan], [Lam07]</td>
</tr>
<tr>
<td>Ramp</td>
<td>Not available</td>
<td>LC-MSMS</td>
<td>[Obe09a], [Obe09b]</td>
</tr>
</tbody>
</table>

Table 1.7: main algorithms used for library search in GC-MS and LC-MSMS.

The Dot Product is implemented in some commercial platforms (e.g. MS Search and INCOS) and open-source projects (e.g. SpectraST). More generally, to our knowledge, most LC-MSMS manufacturers use a Dot Product.

\(^\text{15}\)Such a test is presented in the Ph.D. thesis of Roman Mylonas [Myl10], section 2.1.
based approach in their respective library search applications. Unfortunately, those companies tend to keep their search approaches confidential, so an exhaustive overview is difficult.

Contrasting with its good performances [Ste94], the Dot Product suffers from several drawbacks. For instance, the artificially high scores granted to spectra made of a few peaks only can be mentioned. Furthermore it is not very robust towards spectral variations. Since LC-MSMS is known to create less reproducible spectra, this is a crucial point for any library search platform [Bri02]. Experimental examples of these drawbacks are presented in section 8.

Section 3.1 describes our algorithmic approach and how it is different from existing ones. Sections 6 and 7 are oriented towards the platform itself and its different components.

b) Spectral libraries and library management (∗)

A spectral library is a library which gathers chemical compounds and MS spectra generated from these compounds. Besides spectral information, such as peak intensities and m/z, spectral libraries often propose information regarding the acquisition machine and conditions. Furthermore, spectra are usually annotated with a chemical compound reference, though the type of this reference varies (e.g. a scientific name, a PubChem identifier (CID), or a Chemical Abstracts Service identifier (CAS)).

Whereas large GC-MS libraries are available, few are adapted to LC-MSMS. The most frequently used are presented in Table 1.8. In addition to these libraries, isolated laboratories and MS constructors tend to build their own small spectral libraries, adapted to targeted.

The scarcity of LC-MSMS spectral libraries can be explained by the cost of building such libraries, as well as by the poor availability of certain substances (e.g. substances subject to legal restrictions). New advances in robust scoring models and library structures could be a solution to this lack, enabling cross-libraries identifications and library enrichments. An example of such application is proposed in section 8.4.

Despite the crucial role of spectral libraries, current software tends to include only basic library management functionalities. In fact, this aspect is almost always restricted to data viewing and editing, which fails to take into account needs for data sharing and cross-application use of libraries, for example.
Table 1.8: major LC-MSMS libraries with their access (√: free access, $: licenced access), thematic, size and reference. Size information indicates the number of chemical compounds (before the "/" sign) and the number of MSMS spectra (after the "/" sign). Size indication for the number of spectra in the case of the HMDB library is missing, since this informations is not provided in published statistics. Moreover, the distinction between MS and MSMS spectra is not made for this source.

c) Result visualization (∗)

Visualization of results is usually integrated to the acquisition software. As mentioned in section 1.4.2, this visualization can be very minimalist such as the pdf report generated by ToxId or Cliquid, or more advanced for "manual library search" tools such as with Analyst or Data Analysis. In these two software applications, results are presented in an interactive way, in which users must select area of interest (usually on the chromatogram) to activate the library search and thus generating results.

Both approaches seem to be insufficient with regards to the need of interactions with data. An ideal software solution should comply with both an expert and a less advanced use. Such a more flexible approach, including an easy identification workflow and the possibility to further explore results, is presented in section chapter 7.

1.4.3 Chemical fragmentation applications

a) Fragmentation rules

Fragmentation rules are complex and often not well understood. Generally, a distinction is made between cleavages (where a molecule is broken into sub-structures) and complex rearrangements (where sub-structures of a molecule are connected to form new structures).
Fragmentation patterns depend on many factors. Among them, Fred McLafferty proposed to concentrate on four aspects; the amount of internal energy that an ion obtains during the fragmentation, the stability of the produced ion, the steric effects, and the charge or radical sites of the parent ion [McL93, Hei06]. As previously mentioned\(^{16}\), the most important of these factors is probably the amount of energy that is transferred to the molecule (i.e. the internal energy). From a general point of view, the higher the energy, the more fragments are obtained. With the increase in energy and thus in fragmentation events, obtained fragments are also usually of lower mass. To the contrary, specific structures such as resonance structures, have the ability to stabilize a chemical compound [Kas03]. Resonance structures illustrate the delocalization of electrons within a chemical structure, allowing multiple intermediates to exist [Ker06]. Aromaticity cycles such as the benzene are classical examples of the resonance phenomenon [Zie08, Mo09].

In the field of small molecules, three main strategies are commonly followed to approach the complexity of fragmentation principles. The first is the expert analysis, where some complex rules are applied to determine the weaknesses and the possible rearrangements of a chemical compound [McL93]. This approach is difficult to implement but is of great help while trying to determine and understand the rules that govern chemical compound fragmentation. The second approach consists in studying large databases of fragmentation spectra, in order to identify recurrent patterns. By doing so it is not only possible to study fragmentation patterns, but also to identify fragments that could be discriminant for a given chemical compound. Finally, the third approach involves the methodic fragmentation of every bound of a chemical structure. This last approach is usually time consuming but allows to obtain an exhaustive representation of possible fragment ions, recombinations notwithstanding.

Observations and development regarding the fragmentation of chemical compounds are presented in section 5. In this section, we do not intend to revolutionize the complex science of spectrum interpretation, but rather to point out potentially interesting directions and applications.

\(^{16}\)See section 1.3.2 on fragmentation devices.
b) Fragmentation tools

Chemical fragmentation related software represents a specialized area of software dedicated to small molecule investigations. It covers aspects related to library annotation, fragmentation pattern modeling, or molecule viewing. Commonly used fragmentation software is described in the following paragraphs:

- **Mass Frontier [Hig05]** is defined as a spectral interpretation software. *Mass Frontier* is provided by Thermo Scientific. It is tightly bound to a broad library of literature documented fragmentation pathways. This provides *Mass Frontier* with the ability to take into account complex mechanisms, such as molecular rearrangement. It can also be used in combination with other Thermo Scientific software, like *MetWorks* or *SIEVE* for metabolomic applications. One of the negative characteristics of *Mass Frontier* is its very high price.

- **MS Interpreter [Ste03a]** is also a spectral interpretation software. *MS Interpreter* is provided by NIST. It was originally developed and used by NIST curators in their analysis of mass spectra. *MS Interpreter* primarily provides the user with information related to spectra (peak mass and intensity, losses between two peaks, isotopic patterns, etc.), but also tries to map this information on a molecule formula if this formula is available. *MS Interpreter* is free but only available for *Windows* users. In addition, it does not provide a batch option to carry multiple spectral interpretations.

- **MS Fragmenter [ACD05, Wil02]** is a fragment predictor. *MS Fragmenter* is provided by ACD Labs. It offers fragment prediction, based on fragmentation rules, depending on the fragmentation technique and the polarity used. It also offers the possibility to visualize fragmentation pathways. On the other hand, *MS Fragmenter* is complicated to use, expensive, and only works on *Windows* environments. Like *MS interpreter*, *MS Fragmenter* does not provide any batch option.

- **Fragment Identificator (FiD) [Hei08]** is an identification tool for molecular fragments. *Fragment Identificator* is provided by the department of computer science from the University of Helsinki. *Fragment Identificator* is free and seems still to be active, since the last publication about it dated of 2008.
Section 5 describes our approach and development, aiming at addressing basic aspects of chemical fragmentation.

1.4.4 Software engineering and design

Finally, due to conventional software engineering challenges, and the need for inter-disciplinary knowledge, developing life science software is a complex task. Particularly, needs tend to rapidly evolve or even completely change. In addition, in the field of life sciences, software applications are often used in multiple configuration (i.e. batch mode versus manual, expert versus technician, or routine versus exploratory, for example), adding complexity to the design and development. Moreover, this kind of software is an area where software engineering must wisely combine production and exploratory aspects. This last aspect is probably the most important, though it seems to be often forgotten.

These challenging aspects represent only part of the complexity encountered while developing a software application. More details about software engineering are discussed in sections 2.4 and 6.

1.5 SmileMS: a small molecule identification platform

This thesis presents explorations and developments that led to the conception of SmileMS, an identification platform employed in the field of small molecules. Particularly, four areas are emphasized: the identification algorithm, the issue of spectral libraries, the approach of molecular fragmentation mechanisms, and the challenge of software engineering. These challenges are further described in the next section.
Chapter 2
Challenges

"He who knows all the answers has not been asked all the questions."
Confucius

Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Identification of small molecules</td>
<td>46</td>
</tr>
<tr>
<td>2.1.1</td>
<td>A first identification algorithm</td>
<td>46</td>
</tr>
<tr>
<td>2.1.2</td>
<td>Algorithm refinement: when close compounds become indistinguishable</td>
<td>47</td>
</tr>
<tr>
<td>2.2</td>
<td>The scarcity of spectral libraries</td>
<td>49</td>
</tr>
<tr>
<td>2.2.1</td>
<td>Towards a multi-application conception of libraries</td>
<td>49</td>
</tr>
<tr>
<td>2.2.2</td>
<td>Sharing spectral library data</td>
<td>49</td>
</tr>
<tr>
<td>2.3</td>
<td>Further investigation: LC-MSMS Fragmentation patterns</td>
<td>50</td>
</tr>
<tr>
<td>2.3.1</td>
<td>Understanding fragmentation patterns</td>
<td>50</td>
</tr>
<tr>
<td>2.3.2</td>
<td>Fragmentation software</td>
<td>51</td>
</tr>
<tr>
<td>2.4</td>
<td><em>SmileMS</em>: a small molecule identification platform</td>
<td>51</td>
</tr>
<tr>
<td>2.4.1</td>
<td>Software platform engineering</td>
<td>52</td>
</tr>
<tr>
<td>2.4.2</td>
<td>The identification platform</td>
<td>53</td>
</tr>
<tr>
<td>2.5</td>
<td>Study case: the emergency department of the University Hospital of Geneva</td>
<td>53</td>
</tr>
</tbody>
</table>
This section of my thesis presents the challenges addressed in the next chapters. Particularly, this section exposes issues associated with: the identification of small molecule using on LC-MSMS, the scarcity of spectral libraries adapted to this technology, the question of better understanding chemical fragmentation mechanisms, and the conception of software applied to data exploration and routine identification of small molecules. This section finally introduces several challenges imposed by the University Hospital of Geneva, the first application of our identification platform: SmileMS.

2.1 Identification of small molecules

A first challenge resides in the creation of scoring models adapted to LC-MSMS.

2.1.1 A first identification algorithm

Modern instrumentation, and especially LC-MSMS, have led to an increasing complexity in produced data. Whereas dot-product algorithms where sufficient to cope with spectra generated by GC-MS, the variability created by LC-MSMS instruments requests new approaches.

This consideration grows even stronger with modern laboratories usually cumulating several different instruments. By doing so, they again increase the variability of generated spectra. Figure 2.1 shows spectra of morphine, acquired on two different machines\(^1\) and with three different collision energies for the second instrument. Even though elements are conserved among the four spectra, this figure highlights the variability of LC-MSMS spectra across instruments and acquisition parameters. Straight dot-product comparison with a reference database becomes worthless in such cases. To address this situation, new robust algorithms have to be developed.

Another concern is the trend to look for ever broader range of molecules. As it has been previously mentioned, targeted screening is becoming a standard in drug screening investigations. Although this technique engenders clean spectra, it limits the detection to searched chemical compounds. This approach is clearly not adequate to fields where new molecules appear

\(^1\)With two different fragmentation mechanisms in this case
Figure 2.1: Spectra of morphine, acquired on two different machines and three collision energies for the second instrument. This figure highlights the variability of product spectra across instruments and acquisition parameters.

more rapidly than means to detect them. Anti-doping analysis and food control testing are among these fields. Clinical toxicology also tends to ask for an always increasing range of screened molecules. For such applications, algorithms that can support GUS identification should be conceived.

2.1.2 Algorithm refinement: when close compounds become indistinguishable

Specific solutions must be found to deal with particular circumstances. A classical example is the detection and identification of very close substances. Figure 2.2 presents $MS_2$ (left hand) and $MS_3$ (right hand) fragmentation spectra of an extreme case. In this figure, the $MS_3$ spectrum of compound X is almost indiscernible from its metabolite. Though this case exposes a
very hard challenge\(^2\), it highlights the need for highly accurate scoring capabilities.

![Figure 2.2: Left hand: Product ion mass spectra of (A) parent compound X and (B) metabolite A obtained from ion trap mass spectrometer. Right hand: Product ion MS\(^3\) spectra of (A) parent compound X and (B) metabolite A obtained from ion trap mass spectrometer. Figure retrieved from [Ma06]](image-url)

Algorithms such as the presented one, \(X\text{-Rank}\)^3, cannot address every aspects at the same time. The example of metabolites clearly illustrates this situation. If an algorithm were conceived to detect minimal variations such as the ones encountered in Figure 2.2, it is likely that this algorithm would fail to be flexible enough to handle LC-MSMS variability. An ideal algorithm should then tackle the majority of situations and relies on additional metrics to disambiguate specific cases.

In that respect, the case of close compounds disambiguation should to be address in a complement to the scoring. The main challenges of this situation are:

1. to find a more stringent metric to discriminate close chemical compounds,

2. to combine multiple scores, in order to gain in identification power, without degrading the original scoring model.

\(^2\)In addition to the closeness of spectra, the metabolite and the substance MS\(^3\)’s are isobaric.

\(^3\)X-Rank is part of this doctorate work and will be presented in the eponymous section.
2.2 The scarcity of spectral libraries

The second challenge addressed in this thesis concerns spectral reference libraries. Spectral libraries adapted to LC-MSMS are rare and often MS manufacturer- and/or application-specific. This weak adaptability of libraries can be explained by the information they contain, as well as by the organization of this information. Another consequence of the heterogeneous content of spectral libraries is the lack of sharing mechanisms, allowing to exchange them among collaborating laboratories.

2.2.1 Towards a multi-application conception of libraries

The content issue hindering a broad usability of libraries particularly regards two types of information.

First, spectra generated on different instruments can vary drastically for a same chemical compound. This first aspect limits the use of libraries acquired with a technology different from the one routinely employed.

The second type of information regards metrics that strongly bind reference spectra to the analytical process surrounding their MS acquisition. Particularly, information such as retention time or chemical annotation tend to restrict the use of spectra within the context of an analytical method or a naming system, for example.

To address these limitations, there is a need to find a flexible way to organize information. The issue of chemical annotation should also be addressed, in order to find a standard of chemical compound identifier.

Finally, from a more technical point of view, libraries are distributed in various formats. Despite the availability of open-source standards, such as mzXML or mzML, MS constructors and research groups continue to use proprietary and diverse other library formats. This behavior hinders the possibility to use libraries on multiple systems, without having access to adapted file readers and convertors.

2.2.2 Sharing spectral library data

Sharing libraries, or part of a library, could be a potential solution to the scarcity of spectral libraries, by providing a better availability of existing libraries. Indeed, a central issue, while creating a new spectral library, is the fact that pure substances are often expensive and sometimes difficult to ob-
tain. Consequently, it seems natural to share data acquired in a laboratory, whether for free or through a business scheme.

The flaw of this exchange strategy is that no mechanism exists to achieve this sharing of knowledge. This is partly due to the limits of library adaptability to multiple applications, but also to the lack of sharing platform in the domain of small molecule identification.

An illustration of the need of library sharing is the situation encountered by two laboratory with which we collaborated: the HUG and the food control department. Substances traditionally found in food control analyses are becoming of interest for drug screening. The ideal solution for these two governmental institutions would be to share their reference libraries, but once again, this is not possible due to the exposed limitations.

2.3 Further investigation: LC-MSMS

Fragmentation patterns

An additional concern of this thesis is the approach of theoretical chemical fragmentation. Chemical compound fragmentation is of great interest for LC-MSMS identification, since it represents one of the central phenomenon at play in MS-based technologies. Indeed, fragment ions are separated according to their mass over charge ratio and eventually detected, in order to measure the intensity of a fragment (i.e. the number of fragment of the same sort).

Recall that fragmentation can occur at multiple steps, but principally during the ionization or in a fragmentation cell.

2.3.1 Understanding fragmentation patterns

Understanding the way a molecule is fragmented in MS instruments has for long been a prime objective for chemists. If it were possible to understand and mimic small molecule fragmentation, that would have a huge impact on an array of applications, such as bioorganic chemistry [Leo94], drug design [Sch05] [Jor04], or galenic formulation. It would also be easier to identify unknown spectra generated from a mass spectrometer. In the extreme, it could open the possibility of building theoretical spectral libraries.

However, contrary to protein fragmentation [Pai05], small molecule fragmentation is difficult to predict. More precisely, factors exposed by Fred
2.4. **SmileMS: A small molecule identification platform**

McLafferty\(^4\) are extremely difficult to accurately model [Hei06, Rog00, Sho01]. One of the numerous challenges in this field is then to find alternative ways to approach the mechanisms at play during the fragmentation of small molecules.

### 2.3.2 Fragmentation software

As mentioned in section 1.4.3, tools that compute and help visualizing potential fragmentation patterns exist. To our knowledge, *Mass Frontier* [Hig05] from Thermo Scientific, *MS Interpreter* [Ste03a] from NIST, and *MS Fragmenter* [ACD05, Wil02] from ACD Labs are the most mature initiatives.

Even though these tools can cope with part of the complexity of small molecule fragmentation, there exist lacks that need to be addressed. First, these systems are conceived to explore a few simple molecules, whereas high-throughput and no complexity restriction are required to routinely use fragmentation tools in an identification context. Rapidity factor is to consider seriously in such environment. Second, they are mainly developed to assist in the evaluation of spectra, and do not provide information, whether for routine annotation or for the identification of spectra. Finally, though methodologies used represent a step forward in the understanding of theoretical fragmentation, they are still not accurate enough for routine applications. [Wol99] for example, reports a noticeable help from *Mass Frontier* in a very restrained context\(^5\), but highlights a great number of predicted fragments compared to the number of peaks really obtained.

In this context, there is a need for software applications that could help in the routine annotation of spectra and during the identification process.

2.4 **SmileMS: a small molecule identification platform**

The previously mentioned challenges are all mandatory questions for the conception of an efficient software identification platform in the field of small molecules. In addition the conception itself rises several challenges. This challenges are described in the following paragraphs.

---

\(^4\)Recall section 1.4.3.

\(^5\)Only one chemical compound is studied.
2.4.1 Software platform engineering

Statistics in software development and management alone highlight the need for proper software engineering.

- 35% of requirements change throughout the software lifecycle [Jon96].
- 45% of delivered features are never used [Joh02].
- 82% of projects cited incomplete and unstable requirements as the number one reason for failure [Tay00].
- Large projects costing more than $10 million are successful exactly zero percent of the time [Joh99].

One of the first challenges while developing a software application is to properly identify initial requirements. In the case of a scientific software application, exploration of data is one of these requirements. Managing scientific exploration and discovery, as well as the subsequent transfer of knowledge to a production environment, is thus a crucial challenge.

Design and implementation represent the next software engineering challenge. While design is recognized to be at the heart of a high-quality product, implementation is often regarded as a manual and tiresome work. However, the pair design/implementation guarantees an efficient software application. This is even more important for high-throughput applications that have to treat large amount of data on a daily basis.

In addition, specific challenges are linked to the MS nature of the project. The question of data standards is often a blocking aspect. From the well-known standard initiatives, such as the HUPO Proteomics Standards Initiative (PSI), to proprietary standards, multiple file format exist. These standards have to be wisely handled to ensure the usability of the application in various situations. Besides, useful development should comply with the scientific philosophy, by being accessible and profitable for the scientific community.

From another point of view, highly heterogeneous environments ask for strict integration strategies. Clinical toxicology offers almost as many environments as the number of existing laboratories. For example, these environments can be defined by the type of instrument used, the network configuration, internal policies, or legal requirements. The role of integration is then to ensure the correct functioning of individual software pieces,
when assembled and confronted to different environments. In this context, interactions with third-party MS software applications are also to consider.

Finally, testing and maintenance procedures are essential to guarantee proper functioning of an application through time. Particularly, testing is crucial to avoid side effect on identification scoring. Whereas it is fairly easy to detect an application malfunctioning, it can be far more complicated to spot a shift in generated scores, for example.

2.4.2 The identification platform

The functionalities of the identification platform itself present some interesting challenges. For instance, workflows of data processing can be arranged in different manner, depending on the application. Particularly, the question of dimensionality is important to mention. MS analyses comport multiple steps that generate as many kind of information. Each of these category of information can be considered a dimension in which experimental data are projected. In order to obtain an adapted segregation, these dimensions must be carefully handled.

The handling of experimental data and results is also an important challenge for any software. In the presented project, one of the requirements is to produce a system simple enough to rapidly execute analyses, but which also enables advanced users to access detailed results.

2.5 Study case: the emergency department of the University Hospital of Geneva

Finally, the University Hospital of Geneva was the ground for the first real application of our identification platform. This application rose an array of questions, from multiple perspectives.

First of all, the use of SmileMS at the hospital was the initial application of theoretical models to real identifications. We had to train the algorithm and compare the complete solution to previously used software. Another evaluation consisted in the comparison of MRM and GUS generated data.

From a design point of view, we also had to comply with needs and expectations from physicians, technicians and researchers.

Besides, the hospital laboratory had the opportunity to operate multiple MS instruments. This represented a challenge in terms of identification, sharing, security, library and standards.
In addition, one of these instruments was used in the context of a software integration with Bruker Daltonics. This software integration also implied the building of a spectral library specifically adapted to the analytical design.

Finally, the timescale in itself was a challenge, since a complete solution including SmileMS had to be used in routine at the emergency department of this hospital from March 2010.

**Chapter summary**

In this chapter, we introduced the challenges addressed in this thesis. Particularly, we detailed challenges in the following areas:

1. the identification of small molecules,
2. the availability of spectral libraries,
3. the investigation of LC-MSMS fragmentation patterns,
4. the development of a software platform.

Our approach towards these areas is developed in chapters 3 (Identification), 4 (libraries), 5 (fragmentation patterns), 6 and 7 (software platform). After these sections, chapter 8 presents an application of the platform in the clinical laboratory of the HUG.
Chapter 3

X-Rank scoring model

"As far as the laws of mathematics refer to reality, they are not certain, and as far as they are certain, they do not refer to reality."

Albert Einstein

Contents

3.1 Algorithm description ........................................ 56
  3.1.1 Preliminary definitions .............................. 56
  3.1.2 Observations ........................................... 57
  3.1.3 Scoring model: the log-likelihood ratio ........ 60
  3.1.4 Examples ................................................ 61

3.2 Scoring model performance (*) .............................. 61
  3.2.1 Four comparisons with MS Search ............... 62
  3.2.2 Test mix identification .............................. 65
  3.2.3 Preliminary conclusion .............................. 67

3.3 Scoring model enhancement ................................ 69
  3.3.1 Chemical compounds score-based clustering .... 69
  3.3.2 Intra-compound peak distribution ............... 72
  3.3.3 Score combination: an exploration of Weka classifiers ........................................... 74
As mentioned in section 2.1.1, LC-MSMS acquisitions lead to an increasing complexity in the produced data. This causes existing algorithmic approaches to be no more applicable thus creating a need for new strategies.

Moreover, a unique approach cannot be applied to cope with all situations. Specific scoring models have to be applied to handle peculiar situations.

The next sections describe the development of X-Rank, a powerful and robust algorithm, and explore solutions to improve its performance in specific situations.

3.1 Algorithm description

This section presents X-Rank as an identification algorithm based on statistical observations. We describe how these observations are translated into a predicting score, to form a scoring model.

The principle of X-Rank algorithm is first to sort peak intensities of a spectrum and then, to establish a correlation between the two sorted spectra. Finally, the algorithm relies on a scoring model to discriminate fragmentation spectra from different chemical compounds.

Unlike most published methods, our scoring model does not take into account absolute nor relative intensities. A study on peptide identification methods showed improved results when using ranks instead of peak intensities as a robust metric [Mag04]. Oberacher et al. reached a similar conclusion in the context of small molecule identification [Obe09b]. The accuracy of identification was shown to be improved when reducing the importance of peak intensities [Obe09a, Obe09b].

3.1.1 Preliminary definitions

Suppose $S$ and $S'$ are two fragmentation spectra. Those two measures may originate from the same chemical compound ($H_1$, the alternative hypothesis) or from different compounds ($H_0$, the null hypothesis).

$S$ can be defined as a set of fragment ion peaks $\{f_1, f_2, \ldots f_{\text{size}(S)}\}$. $f_i \sim S'$ indicates that $f_i$ matches one fragment of $S'$ (i.e. the mass of $f_i$ differs from that of the matching peak in $S'$ by less than a given tolerance $\tau$). $f_i \sim f'_j$ indicates that fragment $f_i$ matches the specific fragment $f'_j$ from $S'$. Let $f_{X(1)}$
be the most intense fragment of \( S \) (\( f_{\lambda(2)} \) is the second most intense fragment and so on).

The first step is to model probabilities for each hypothesis (i.e. \( H_1 \) and \( H_0 \)) that a peak of \( S \) matches one of \( S' \) according to their respective ranks. This can be respectively stated as in:

\[
P(f_{\lambda(i)} \sim f'_{\lambda(j)} | H_1) \tag{3.1}
\]

\[
P(f_{\lambda(i)} \sim f'_{\lambda(j)} | H_0) \tag{3.2}
\]

The theorem of composite probabilities can be applied to obtain:

\[
P(f_{\lambda(i)} \sim f'_{\lambda(j)} | H_x) = P(f_{\lambda(i)} \sim S' | H_x) \cdot P(f_{\lambda(i)} \sim f'_{\lambda(j)} | f_{\lambda(i)} \sim S', H_x) \tag{3.3}
\]

### 3.1.2 Observations

The experimental distributions of the two probabilities on the right side of equation 3.3 can be computed empirically given a large set of observations (i.e. manually validated identification spectra against a library of chemical compound spectra).

We considered, as a test set, the high quality spectrum library built by Wolfgang Weinmann [Mue05]. Two spectra of the same compound will account for the correct hypothesis \( H_1 \). Two spectra of different compounds will account for the random hypothesis \( H_0 \).

Figure 3.1 (left) shows that \( P(f_{\lambda(i)} \sim S' | H_1) \) can be modeled by fitting a negative exponential curve. Since we did not find any correlation between \( \lambda \) (i.e. the rank of the fragment) and the estimated probability that two fragments match for \( H_1 \) (Figure 3.1 right), we simply approximated this value by the median; 0.065.

For correct matches, Figure 3.2 (left) shows experimental distributions for \( P(f_{\lambda(i)} \sim f'_{\lambda(j)} | f_{\lambda(i)} \sim S', H_1) \), for selected values of \( i \). In the same figure, we show how these distributions can be fitted by “Gamma-like” distributions (plain lines).

A Gamma distribution is a statistical distribution that involves two parameters; a scale parameter that determines the position of the curve on the \( x \)-axis (i.e. \( \kappa \) on Figure 3.3), and a second parameter that determines the shape of the curve (i.e. \( \theta \) on Figure 3.3). Figure 3.3 presents some examples of Gamma distributions.

In the case of \( X-Rank \), we use “Gamma-like” distributions, including a third parameter which represents a vertical shift of the curve. This choice is
CHAPTER 3. X-RANK SCORING MODEL

Figure 3.1: observation and fit of fragment spectra matches in a set of manually validated observations. Left: density of the probability, depending on intensity rank, that two fragments from spectra of the same chemical substance match. The experimental distribution is fitted by a negative exponential distribution. This case will account for the correct hypothesis $H_1$. This is denoted $P(f_{\lambda(i)} \sim S|H_1)$. Right: probability that two fragments from spectra of two different chemical substances match. This case will account for the random hypothesis $H_0$, it is denoted $P(f_{\lambda(i)} \sim S'|H_0)$.

justified to fit random match noise, and thus to better fit to the experimental data.

In contrast, for wrong matches, experimental distributions for $P(f_{\lambda(i)} \sim f'_{\lambda(j)}|f_{\lambda(i)} \sim S', H_0)$, shown in Figure 3.2 right, can be fitted with negative exponential functions. Fitting these density functions with explicit functions results in a simpler model and thus avoids over fitting. In these distributions, only the thirty most intense peaks were kept in each spectrum. In practice lower intensity signals (i.e. higher ranks) are poorly informative.
3.1. ALGORITHM DESCRIPTION

Figure 3.2: observation and fit of fragment spectra matches from a large set of manually validated observations (for an overview only the first 3 ranks of spectra $S$ are displayed). The green circles represent the estimated probability that the first rank in spectrum $S$ matches the fragment of rank $x$ in spectrum $S'$. The red plus signs and the blue asterisk signs represent the estimated probability of the second, respectively the third, rank of the spectrum $S$. Left: in this figure $S$ and $S'$ come from the same chemical compound. This is denoted $P(f_{\lambda(i)} \sim f'_{\lambda(j)} | f_{\lambda(i)} \sim S', H_1)$. These distributions can be fitted by Gamma-like distributions (plain lines). Right: in this figure, $S$ and $S'$ come from two different chemical compounds. This is denoted $P(f_{\lambda(i)} \sim f'_{\lambda(j)} | f_{\lambda(i)} \sim S', H_0)$. These distributions can be fitted by negative exponential distributions (plain lines).

Interestingly, we noticed similar behavior for other types of data sets. More data, as well as examples derived from peptide identification are given in the thesis of Roman Mylonas [Myl10], section 6.1.

a) Observations summary

To summarize, the following parameters need to be computed when a scoring is learned:

- one parameter for the negative exponential function presented in Figure 3.2 left,
CHAPTER 3. X-RANK SCORING MODEL

Figure 3.3: examples of Gamma distributions. (Figure retrieved from http://en.wikipedia.org/wiki/Gamma_distribution)

- one fixed value (presented in Figure 3.2 right),
- a set of $\kappa$, $\theta$, and a parameter representing the $y$-axis shift, for each gamma curve (i.e. for each considered rank) (Figure 3.1 left),
- and one second negative exponential parameter (Figure 3.1 right).

In total, the number of computed parameters is equal to $3 + 3$ times the number of considered ranks. With the usual settings (i.e. 30 ranks are considered), the total number of computed parameters is thus 93. In practice, these parameters are learned from representative datasets (See thesis of Roman Mylonas [Myl10] for further details, section 3.1.).

3.1.3 Scoring model: the log-likelihood ratio

Given the observed alignment between two spectra $S \sim S'$, we can now compute the global probabilities $P(S \sim S'|H_0)$ and $P(S \sim S'|H_1)$ of this observation being random or correct. If we assume that matches between
3.2. SCORING MODEL PERFORMANCE (*)

individual fragment peaks are independent. In practice, peaks are not independent, however, this simplification is used by other models [?], we can factorize for $H_0$ (respectively $H_1$):

$$P(S \sim S' | H_0) = \prod_{i=0}^{n} \left\{ \begin{array}{ll} P(f_{\lambda(i)} \sim f'_{\lambda'(j)} | H_0), & \text{if } f_{\lambda(i)} \sim f'_{\lambda'(j)} \\ 1 - P(f_{\lambda(i)} \sim S' | H_0), & \text{if } f_{\lambda(i)} \not\sim S' \end{array} \right. \tag{3.4}$$

Deciding whether an observation is random or not is now reduced to a simple hypothesis testing problem. The Neyman-Pearson Lemma shows that the optimal statistics, for deciding between the null hypothesis (random match) and the alternative hypothesis (correct match), is the likelihood ratio, that is, the probability of a correct match divided by the probability of a random match. Consequently, we define the score of a match as:

$$\sigma(S, S') = \log \frac{P(S \sim S' | H_1)}{P(S \sim S' | H_0)} \tag{3.5}$$

In practice, the logarithm is taken for numerical convenience.

3.1.4 Examples

Figure 3.4 illustrates the computation of a match (for simplification only the three first ranks of the spectrum $S$ are considered). The first rank of spectrum $S$ matches the first rank of spectrum $S'$ and the second matches the fourth one. This is denoted as $f_{\lambda(1)} \sim f'_{\lambda'(1)}$ and $f_{\lambda(2)} \sim f'_{\lambda'(4)}$. The third rank does not have a matching fragment and is consequently denoted as $f_{\lambda(3)} \not\sim S'$. According to equation 3.4, the probability $P(S \sim S' | H_0)$ can thus be expressed as $P(f_{\lambda(1)} \sim f'_{\lambda'(1)} | H_0) \cdot P(f_{\lambda(2)} \sim f'_{\lambda'(4)} | H_0) \cdot P(f_{\lambda(3)} \not\sim S')$ (respectively $H_1$).

The score we obtain is not symmetric. It depends which spectra are considered for $S$ and $S'$. In practice, to obtain a symmetric score, the score is simply computed twice, once straightly and once with inverse $S$ and $S'$. In addition, the final score value is rescaled between 0 and 1$^1$.

3.2 Scoring model performance (*)

In order to assess the performance of the algorithm, we conducted two different experiments. For the first one, two datasets were searched against

$^1$Further details on MS Search are provided in the thesis of Roman Mylonas [Myl10], section 2.3.
CHAPTER 3. X-RANK SCORING MODEL

Figure 3.4: (*) illustration of a score computation (for a clearer view only the three first ranks of the spectrum $S$ are considered). In this example, the first rank of spectrum $S$ matches the first rank of spectrum $S'$ and the second matches the fourth one. This is denoted as $f_{\lambda(1)} \sim f'_{\lambda'(1)}$, respectively $f_{\lambda(2)} \sim f'_{\lambda'(4)}$. The third rank does not have a matching fragment and is consequently denoted as $f_{\lambda(3)} \not\sim S'$. This results in $P(S \sim S'|H_0) = P(f_{\lambda(1)} \sim f'_{\lambda'(1)}|H_0) \cdot P(f_{\lambda(2)} \sim f'_{\lambda'(4)}|H_0) \cdot (1 - P(f_{\lambda(3)} \not\sim S'))$ (respectively $H_1$).

each other and against itself in a leave-one-out manner (LOO).

3.2.1 Four comparisons with MS Search

Each of the four identification tasks was conducted using X-Rank and MS Search tool from NIST². Because the MS Search tool does not allow for any parameter training on specific data, X-Rank parameters have been trained only with QTRAP data, voluntarily avoiding re-training for each compari-

²Further details on MS Search are provided in the thesis of Roman Mylonas [Myl10], section 2.1.
son. In practice, X-Rank is actually trained to optimize specific situations, yet this example aims at highlighting cross-platform performance of X-Rank.

Neither X-Rank nor MS Search include the precursor ion in score computation. The identification was conducted as described in Table 3.1. Indications in Table 3.1 refer to results presented in the following paragraphs. The two used datasets are:

- Weinmann data set: this data set consists of 1050 product ion spectra acquired on a Applied Biosystems / MDS Sciex QTRAP, from 344 different substances. The data mainly originate from the commercially available Weinmann library enriched by additional compounds acquired by Geneva University Hospital (GUH) on the same instrument under similar conditions. The acquisition method consists in MRM / EPI runs with three different collision energies; 20, 35 and 40 eV [Vie09].

- IT-NIST data set: product ion spectra acquired on a IT machine and containing at least 5 fragmentation peaks were extracted from the NIST LC-MSMS 2008 library. Fragmentation spectra of this library were acquired under different conditions and collected by NIST. 4504 spectra from 1171 different substances were obtained.

<table>
<thead>
<tr>
<th>Experimental</th>
<th>Reference</th>
<th>IT</th>
<th>QTRAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>IT</td>
<td>Figure 3.5 A</td>
<td>Figure 3.5 B</td>
<td></td>
</tr>
<tr>
<td>QTRAP</td>
<td>Figure 3.5 C</td>
<td>Figure 3.5 D</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.1: four different comparative tests were undertaken; the IT set was identified against the IT library and the QTRAP library, the QTRAP set was identified against the QTRAP ion trap library and the IT library.

Figure 3.5 presents corresponding Receiver Operating Characteristic (ROC) [Par04] curves of X-Rank and MS Search performance. Red curves represent identifications conducted with X-Rank, whereas blue curves represent identification conducted with MS Search. On the whole, X-Rank appears as a more specific and sensitive method, especially for cross-platform identification. X-Rank shows an important improvement over the other algorithm.
This good performance opens the possibility of spectra identifying using heterogeneous libraries.

- Figure 3.5 A shows the comparison of MS Search and X-Rank while identifying ion trap data in a leave one-out-manner. Along with Figure 3.5 D, this figure illustrates the easiest cases: test data are identified against data generated with the same experimental design. All along, both specificity and sensitivity are improved with X-Rank. Since this situation is rather straightforward, the slight improvement on the upper-left corner of the chart is nonetheless significant, while it indicates an increase of correct identifications given a smaller false positive rate, which becomes a key point for high-throughput analysis. At most, 3932 identifications were produced by X-Rank, while only 3916 by MS Search.

- Figure 3.5 B shows the comparison of MS Search and X-Rank while identifying ion trap data against a QTRAP library. It displays the performance of the two algorithms while identifying a test data set acquired with a fragmentation type against a reference data set acquired with a distinct fragmentation type. Although other parameters differ, fragmentation is the most important one in this situation. Overall, results are less impressive than in the previous case, but still highlight a significant improvement compared to the existing identification algorithm. The curve of MS Search does not reach the one hundred percent value. This indicates that MS Search does not manage to identify all the substances even with a low score. At most, 188 identifications were produced by X-Rank, while only 174 by MS Search.

- Figure 3.5 C shows the comparison of MS Search and X-Rank while identifying QTRAP data against an ion trap library. This figure demonstrates clear improvement with X-Rank. Improvement spreads across low false positive rate up to the maximum false positive rate. Like in Figure 3.5 B, the curve representing the performance of MS Search does not reach the one hundred percent value. This indicates that MS Search does not manage to identify all the substances even with a low score. At most, 172 identifications were produced by X-Rank, while only 162 by MS Search.

- Figure 3.5 D shows the comparison of MS Search and X-Rank while identifying QTRAP data in a leave one-out-manner. This figure is
very similar to Figure 3.5 A. X-Rank performs better for the low positive rate section. MS Search performs slightly better for a false positive rate between 15 and 30%. Finally, the two algorithms obtain equal performance for a false positive rate above 30%. At most, 1049 identifications were produced by X-Rank, compared to the similar 1044 by MS Search.

3.2.2 Test mix identification

For the second experiment, a publicly available test mix for drug analysis and acquired on an Applied Biosystem QTRAP 4000, was identified against the aforementioned Weinmann data set. This test mix, developed by Restek with the help of Applied Biosystems | MDS Sciex, is routinely used for the assessment of Applied Biosystem’s Cliquid software. The test mix consists of the eight forensic drugs, presented in Table 3.2. As for the previously presented comparisons, the identification was conducted on both X-Rank and MS Search.

<table>
<thead>
<tr>
<th>Chemical substance</th>
<th>Concentration (in µg/mL)</th>
</tr>
</thead>
<tbody>
<tr>
<td>amiodarone</td>
<td>10</td>
</tr>
<tr>
<td>amphetamine</td>
<td>10</td>
</tr>
<tr>
<td>caffeine</td>
<td>10</td>
</tr>
<tr>
<td>codeine</td>
<td>10</td>
</tr>
<tr>
<td>diazepam</td>
<td>10</td>
</tr>
<tr>
<td>doxepine</td>
<td>10</td>
</tr>
<tr>
<td>haloperidol</td>
<td>1</td>
</tr>
<tr>
<td>morphine</td>
<td>10</td>
</tr>
</tbody>
</table>

Table 3.2: publicly available test mix for drug analysis composition. This test mix, developed by Restek with the help of Applied Biosystems | MDS Sciex, consists of the eight forensic drugs.

Table 3.3 shows on a three columns table the comparison of SmileMS and MS Search for this test mix identification. Columns one and two show two different MS Search scores. The first one is described as a probability score, while the second one is described as a matching score. The third column corresponds to identification with X-Rank. Correct identifications are
Figure 3.5: four comparison cases; A shows ion trap data identified in a leave one-out-manner, B shows ion trap data identified against a QTRAP library, C shows QTRAP data against an ion trap library and finally, D shows QTRAP data identification in a leave one-out-manner. Each figure shows performance in red for X-Rank and in blue for MS Search. On the whole, X-Rank appears as a more specific and sensitive method, especially for cross-platform identification.
highlighted in green, according to the known test mix content. Names presented are retrieved without modification from Pubchem except for the D5-Amphetamine. In column one and two, D5-Amphetamine could be considered green, since this substance was not one of the searched substances, but used as an internal standard. The other internal standard, the D5-Doxepin is never present. Our interpretation is that the D5-Doxepin is wrongly identified as the Doxepin since no filter on the precursor mass was used. The same could apply for the D5-Amphetamine in the case of X-Rank.

This table clearly shows the much better behavior of X-Rank. First, the eight searched compounds are found in the first positions using X-Rank. It is far from being the case using MS Search. Second, It is possible to apply a score threshold with X-Rank. For example, thanks to a clearer decrease in score values, a threshold put at 0.5 leads to only one false negative (i.e. the amphetamine). On the other hand, putting a score threshold to MS Search leads to unusable results.

### 3.2.3 Preliminary conclusion

For the two presented studies, X-Rank showed better performance when compared to MS Search.

In the first study, though the difference between the two tools is rather small for standard identifications, it becomes bigger in the cases of cross-platform uses.

The second performance study showed a good discrimination of correct and wrong identifications, in the case of X-Rank. Indeed, it is possible to apply a threshold that completely separates true positives from false positive identifications. This is not the case for the two metrics proposed by MS Search. Applying a fixed threshold is important in toxicology screening. Without such a metric, it is often difficult to judge if an identification is correct or not.

An even better behavior would be to directly obtain a value of true or false, while analyzing the match between two spectra. This kind of approach is developed in the next sections.
CHAPTER 3. X-RANK SCORING MODEL

Table 3.3: Three-column table comparison of SmileMS and MS Search for a test mix of identifications. The first column is described as a probability score, while the second one is described as a matching score. The third column corresponds to X-Rank identification. Correct identifications are highlighted in green, according to the known test mix content. This table clearly shows the better behavior of X-Rank.

<table>
<thead>
<tr>
<th>Drug</th>
<th>MS Search Probability</th>
<th>MS Search Matching</th>
<th>X-Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acetaminophen</td>
<td>98.11</td>
<td>Tolbutamide</td>
<td>98.74</td>
</tr>
<tr>
<td>Haloperidol</td>
<td>96.8</td>
<td>Prilocaine</td>
<td>97.79</td>
</tr>
<tr>
<td>Parathion</td>
<td>87.2</td>
<td>Morphine</td>
<td>737</td>
</tr>
<tr>
<td>Tolbutamide</td>
<td>97.68</td>
<td>Diazepam</td>
<td>97.68</td>
</tr>
<tr>
<td>Ketoprofen</td>
<td>97.34</td>
<td>Diazepam</td>
<td>97.68</td>
</tr>
<tr>
<td>Prilocaine</td>
<td>97.1</td>
<td>Ketoprofen</td>
<td>97.34</td>
</tr>
<tr>
<td>Amantadine</td>
<td>96.5</td>
<td>Tocainide</td>
<td>96</td>
</tr>
<tr>
<td>Phenyldpropranalmine</td>
<td>96.4</td>
<td>Tocainide</td>
<td>96.4</td>
</tr>
<tr>
<td>Caffeine</td>
<td>89.6</td>
<td>Gliclazide</td>
<td>89.73</td>
</tr>
<tr>
<td>Amiodarone</td>
<td>95.9</td>
<td>Parathion</td>
<td>86.11</td>
</tr>
<tr>
<td>Glimepiride</td>
<td>95.9</td>
<td>Prilocaine</td>
<td>499</td>
</tr>
<tr>
<td>Tramadol</td>
<td>95.8</td>
<td>Benzocaine</td>
<td>95.9</td>
</tr>
<tr>
<td>Parathion</td>
<td>95.8</td>
<td>Nicotinic acid hydrazide</td>
<td>95.9</td>
</tr>
<tr>
<td>Doxepin</td>
<td>93.24</td>
<td>Tocainide</td>
<td>96.99</td>
</tr>
<tr>
<td>Venlafaxine</td>
<td>93.24</td>
<td>Prilocaine</td>
<td>93.24</td>
</tr>
<tr>
<td>Pseudoephedrine</td>
<td>93.73</td>
<td>Benzocaine</td>
<td>93.73</td>
</tr>
<tr>
<td>Tramadol</td>
<td>93.73</td>
<td>Tocainide</td>
<td>93.73</td>
</tr>
<tr>
<td>Tocainide</td>
<td>93.11</td>
<td>Pseudoephedrine</td>
<td>93.73</td>
</tr>
<tr>
<td>Tocainide</td>
<td>92.69</td>
<td>Cetirizine</td>
<td>92.69</td>
</tr>
<tr>
<td>Propylsalicylic acid</td>
<td>92.69</td>
<td>Cetirizine</td>
<td>92.69</td>
</tr>
<tr>
<td>Penbutolol</td>
<td>92.69</td>
<td>Caffeine</td>
<td>92.69</td>
</tr>
<tr>
<td>Benzocaine</td>
<td>92.69</td>
<td>Benzocaine</td>
<td>92.69</td>
</tr>
<tr>
<td>Pseudoephedrine</td>
<td>92.69</td>
<td>Caffeine</td>
<td>92.69</td>
</tr>
<tr>
<td>Venlafaxine</td>
<td>92.69</td>
<td>Benzocaine</td>
<td>92.69</td>
</tr>
<tr>
<td>Mescaline</td>
<td>92.69</td>
<td>Caffeine</td>
<td>92.69</td>
</tr>
<tr>
<td>Acetaminophen</td>
<td>92.69</td>
<td>Benzocaine</td>
<td>92.69</td>
</tr>
<tr>
<td>Haloperidol</td>
<td>92.69</td>
<td>Caffeine</td>
<td>92.69</td>
</tr>
<tr>
<td>Parathion</td>
<td>92.69</td>
<td>Benzocaine</td>
<td>92.69</td>
</tr>
<tr>
<td>Tolbutamide</td>
<td>92.69</td>
<td>Caffeine</td>
<td>92.69</td>
</tr>
<tr>
<td>Ketoprofen</td>
<td>92.69</td>
<td>Benzocaine</td>
<td>92.69</td>
</tr>
<tr>
<td>Prilocaine</td>
<td>92.69</td>
<td>Caffeine</td>
<td>92.69</td>
</tr>
<tr>
<td>Amantadine</td>
<td>92.69</td>
<td>Benzocaine</td>
<td>92.69</td>
</tr>
<tr>
<td>Phenyldpropranalmine</td>
<td>92.69</td>
<td>Caffeine</td>
<td>92.69</td>
</tr>
<tr>
<td>Caffeine</td>
<td>92.69</td>
<td>Benzocaine</td>
<td>92.69</td>
</tr>
<tr>
<td>Amiodarone</td>
<td>92.69</td>
<td>Caffeine</td>
<td>92.69</td>
</tr>
<tr>
<td>Glimepiride</td>
<td>92.69</td>
<td>Benzocaine</td>
<td>92.69</td>
</tr>
<tr>
<td>Tramadol</td>
<td>92.69</td>
<td>Caffeine</td>
<td>92.69</td>
</tr>
<tr>
<td>Parathion</td>
<td>92.69</td>
<td>Benzocaine</td>
<td>92.69</td>
</tr>
<tr>
<td>Doxepin</td>
<td>92.69</td>
<td>Caffeine</td>
<td>92.69</td>
</tr>
<tr>
<td>Venlafaxine</td>
<td>92.69</td>
<td>Benzocaine</td>
<td>92.69</td>
</tr>
<tr>
<td>Pseudoephedrine</td>
<td>92.69</td>
<td>Caffeine</td>
<td>92.69</td>
</tr>
<tr>
<td>Tramadol</td>
<td>92.69</td>
<td>Benzocaine</td>
<td>92.69</td>
</tr>
<tr>
<td>Tocainide</td>
<td>92.69</td>
<td>Caffeine</td>
<td>92.69</td>
</tr>
<tr>
<td>Tocainide</td>
<td>92.69</td>
<td>Benzocaine</td>
<td>92.69</td>
</tr>
<tr>
<td>Pseudoephedrine</td>
<td>92.69</td>
<td>Caffeine</td>
<td>92.69</td>
</tr>
<tr>
<td>Tramadol</td>
<td>92.69</td>
<td>Benzocaine</td>
<td>92.69</td>
</tr>
<tr>
<td>Tocainide</td>
<td>92.69</td>
<td>Caffeine</td>
<td>92.69</td>
</tr>
<tr>
<td>Tocainide</td>
<td>92.69</td>
<td>Benzocaine</td>
<td>92.69</td>
</tr>
<tr>
<td>Pseudoephedrine</td>
<td>92.69</td>
<td>Caffeine</td>
<td>92.69</td>
</tr>
<tr>
<td>Tramadol</td>
<td>92.69</td>
<td>Benzocaine</td>
<td>92.69</td>
</tr>
<tr>
<td>Tocainide</td>
<td>92.69</td>
<td>Caffeine</td>
<td>92.69</td>
</tr>
<tr>
<td>Tocainide</td>
<td>92.69</td>
<td>Benzocaine</td>
<td>92.69</td>
</tr>
<tr>
<td>Pseudoephedrine</td>
<td>92.69</td>
<td>Caffeine</td>
<td>92.69</td>
</tr>
<tr>
<td>Tramadol</td>
<td>92.69</td>
<td>Benzocaine</td>
<td>92.69</td>
</tr>
<tr>
<td>Tocainide</td>
<td>92.69</td>
<td>Caffeine</td>
<td>92.69</td>
</tr>
<tr>
<td>Tocainide</td>
<td>92.69</td>
<td>Benzocaine</td>
<td>92.69</td>
</tr>
<tr>
<td>Pseudoephedrine</td>
<td>92.69</td>
<td>Caffeine</td>
<td>92.69</td>
</tr>
<tr>
<td>Tramadol</td>
<td>92.69</td>
<td>Benzocaine</td>
<td>92.69</td>
</tr>
<tr>
<td>Tocainide</td>
<td>92.69</td>
<td>Caffeine</td>
<td>92.69</td>
</tr>
<tr>
<td>Tocainide</td>
<td>92.69</td>
<td>Benzocaine</td>
<td>92.69</td>
</tr>
<tr>
<td>Pseudoephedrine</td>
<td>92.69</td>
<td>Caffeine</td>
<td>92.69</td>
</tr>
<tr>
<td>Tramadol</td>
<td>92.69</td>
<td>Benzocaine</td>
<td>92.69</td>
</tr>
<tr>
<td>Tocainide</td>
<td>92.69</td>
<td>Caffeine</td>
<td>92.69</td>
</tr>
<tr>
<td>Tocainide</td>
<td>92.69</td>
<td>Benzocaine</td>
<td>92.69</td>
</tr>
<tr>
<td>Pseudoephedrine</td>
<td>92.69</td>
<td>Caffeine</td>
<td>92.69</td>
</tr>
<tr>
<td>Tramadol</td>
<td>92.69</td>
<td>Benzocaine</td>
<td>92.69</td>
</tr>
<tr>
<td>Tocainide</td>
<td>92.69</td>
<td>Caffeine</td>
<td>92.69</td>
</tr>
<tr>
<td>Tocainide</td>
<td>92.69</td>
<td>Benzocaine</td>
<td>92.69</td>
</tr>
<tr>
<td>Pseudoephedrine</td>
<td>92.69</td>
<td>Caffeine</td>
<td>92.69</td>
</tr>
<tr>
<td>Tramadol</td>
<td>92.69</td>
<td>Benzocaine</td>
<td>92.69</td>
</tr>
<tr>
<td>Tocainide</td>
<td>92.69</td>
<td>Caffeine</td>
<td>92.69</td>
</tr>
<tr>
<td>Tocainide</td>
<td>92.69</td>
<td>Benzocaine</td>
<td>92.69</td>
</tr>
<tr>
<td>Pseudoephedrine</td>
<td>92.69</td>
<td>Caffeine</td>
<td>92.69</td>
</tr>
<tr>
<td>Tramadol</td>
<td>92.69</td>
<td>Benzocaine</td>
<td>92.69</td>
</tr>
<tr>
<td>Tocainide</td>
<td>92.69</td>
<td>Caffeine</td>
<td>92.69</td>
</tr>
<tr>
<td>Tocainide</td>
<td>92.69</td>
<td>Benzocaine</td>
<td>92.69</td>
</tr>
<tr>
<td>Pseudoephedrine</td>
<td>92.69</td>
<td>Caffeine</td>
<td>92.69</td>
</tr>
<tr>
<td>Tramadol</td>
<td>92.69</td>
<td>Benzocaine</td>
<td>92.69</td>
</tr>
<tr>
<td>Tocainide</td>
<td>92.69</td>
<td>Caffeine</td>
<td>92.69</td>
</tr>
<tr>
<td>Tocainide</td>
<td>92.69</td>
<td>Benzocaine</td>
<td>92.69</td>
</tr>
<tr>
<td>Pseudoephedrine</td>
<td>92.69</td>
<td>Caffeine</td>
<td>92.69</td>
</tr>
<tr>
<td>Tramadol</td>
<td>92.69</td>
<td>Benzocaine</td>
<td>92.69</td>
</tr>
<tr>
<td>Tocainide</td>
<td>92.69</td>
<td>Caffeine</td>
<td>92.69</td>
</tr>
<tr>
<td>Tocainide</td>
<td>92.69</td>
<td>Benzocaine</td>
<td>92.69</td>
</tr>
<tr>
<td>Pseudoephedrine</td>
<td>92.69</td>
<td>Caffeine</td>
<td>92.69</td>
</tr>
<tr>
<td>Tramadol</td>
<td>92.69</td>
<td>Benzocaine</td>
<td>92.69</td>
</tr>
<tr>
<td>Tocainide</td>
<td>92.69</td>
<td>Caffeine</td>
<td>92.69</td>
</tr>
<tr>
<td>Tocainide</td>
<td>92.69</td>
<td>Benzocaine</td>
<td>92.69</td>
</tr>
<tr>
<td>Pseudoephedrine</td>
<td>92.69</td>
<td>Caffeine</td>
<td>92.69</td>
</tr>
<tr>
<td>Tramadol</td>
<td>92.69</td>
<td>Benzocaine</td>
<td>92.69</td>
</tr>
<tr>
<td>Tocainide</td>
<td>92.69</td>
<td>Caffeine</td>
<td>92.69</td>
</tr>
</tbody>
</table>
3.3 Scoring model enhancement

This section aims at presenting improvements to the previous scoring model, when confronted to particularly challenging identifications. One obvious way to increase the discrimination power is to use metrics of retention time and precursor mass\(^3\). However, we focussed on improving the scoring model itself, keeping the possibility to include other metrics as *a posteriori* filters.

3.3.1 Chemical compounds score-based clustering

One approach to test scoring capabilities is to cluster results and detect abnormalities. Figure 3.6 shows the clustering analysis of a subset of a spectral library constructed by Wolfgang Weinmann and completed by the HUG\(^4\). We chose a dendrogram representation of the clusters (as opposed to a standard xy plots) to ease molecule names reading. Branches with numbers instead of names are spectra for which no matching name was found in the PubChem repository.

The distance matrix that was used to establish the cluster analysis was based on a \(1/score\) metric. This means that pairs of spectra that obtain a high score are attributed a low distance matrix coefficient and thus are closer on the dendrogram. An alternative solution would be to use \(\log(score)\). The disadvantage of this second approach is the non-linearity that the \(\log\) imposes. Proper visualization of results is rendered difficult in this case. Computation of the dendrogram was carried out using the *Hierarchical Clustering* (hclust) package of R.

Two types of situations are of peculiar interest:

1. distant branches with common names: indicates that a low score is attributed to a pair of spectra of the same substance,

2. close branches with different names: indicates that a high score is attributed to a pair of spectra of two different substances.

To illustrate these two situations, consider the following examples:

---

\(^3\)See thesis of Roman Mylonas [Myl10], section 3.4 for further details.

\(^4\)This library was constructed using a QTRAP from Applied Biosystem. From this point onward, this library will be called Weinmann/HUG.
Figure 3.6: Clustering analysis of a subset of the spectral library constructed by Wolfgang Weinmann and completed by the HUG. The distance matrix that was used to establish the cluster analysis was based on a $1/\text{score}$ metric.

1. the large distance between two Ofloxacins (spectra 441 and 445) and another one (spectrum 443) in blue on Figure 3.6,

2. the apparent closeness of Sulpiride (spectra 489 and 493) and the Naltrexone (spectrum 273) in red on Figure 3.6.

Spectra of these two cases are illustrated on Figure 3.7 left and right. Distance between Ofloxacins is explained by the discrepancy of fragmentation patterns. Closeness of Naltrexone and Sulpiride is explained by similar fragments found for these two substances.

In order to improve the discriminating power of X-Rank, two goals were pursued:
3.3. SCORING MODEL ENHANCEMENT

Figure 3.7: left: three spectra of Ofloxacins (441, 443, and 445). Right: one spectrum of Naltrexone (273), and two of Sulpiride (489 and 493).

1. Gather distant but similar identifications.

2. Segregate close structures, but dissimilar compounds.

Two strategies can be applied to overcome the first situation. First, scoring can be flexible enough to handle spectral variability. This applies within a certain limit. If spectra are too different, then a good match score is meaningless, even if they correspond to the same compound. The second strategy consists in including the broader possible variety of spectra from a given compound to the library. Consequently, the likelihood to come across a completely new form of spectrum while identifying unknown samples is decreased. In practice, both strategies are employed at the same time.

The second situation is harder. Firstly it appears frequently, especially when dealing with metabolites or compounds from a same family (i.e. sharing a similar molecular skeleton structure). Figure 3.8 presents molecular structures and spectra of two close benzodiazepines (Oxazepam and Demoxepam). Secondly, the risk of mistaking the intra-compound variability of spectra for the identification of two different substance is high (i.e. one compound and its metabolite, for instance).

To address this second challenge, a two-step approach was defined. It consists in:
1. Detect cases where top-ranked identifications obtain close scores. This highlights a potential identification conflict. Table 3.4, Score id. A shows a clear score distribution, whereas Score id. B shows a potentially conflictual situation.

2. Apply a second, more discriminating, scoring to discriminate between close candidates.

### 3.3.2 Intra-compound peak distribution

In order to differentiate close identifications, a more discriminating signature of compounds can be defined. As it is shown in Figure 3.7, for instance, spectra from a given chemical substance share a certain number of fragmentation peaks. Figure 3.9 presents the distribution of the percentage of shared peaks within spectra of a given compound, for the substances of the Weinmann/HUG spectral library. This figure reads that the great majority of spectra share more than 20% of their peaks with other spectra from the same substance, for instance.

One possible stringent approach towards close-substances discrimination consists in focusing on fragmentation peaks present in every spectrum of a given substance. The obtained collection of conserved peaks forms a signature for the studied substance.
### Table 3.4: Score id. A: scores distribution of a clear analysis. Interval from 0.7 to 1 (orange cells) contains only one identification. The next one appears far behind at around 0.4. Score id. B: scores distribution of an ambiguous analysis. Interval from 0.7 to 1 (orange cells) contains three close identifications.

<table>
<thead>
<tr>
<th>A candidate molecule</th>
<th>Score id. A</th>
<th>Score id. B</th>
<th>B candidate molecule</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carbimazole</td>
<td>0.91</td>
<td>0.91</td>
<td>Bergapten</td>
</tr>
<tr>
<td>Carbimazole</td>
<td>0.40</td>
<td>0.84</td>
<td>Methoxsalen</td>
</tr>
<tr>
<td>Carbimazole</td>
<td>0.12</td>
<td>0.75</td>
<td>Methoxsalen</td>
</tr>
<tr>
<td>Aminophenazone</td>
<td>0.08</td>
<td>0.41</td>
<td>Bergapten</td>
</tr>
<tr>
<td>Debrisoquine</td>
<td>0.07</td>
<td>0.12</td>
<td>Flumazenil</td>
</tr>
<tr>
<td>Moclobemide</td>
<td>0.06</td>
<td>0.12</td>
<td>Methoxsalen</td>
</tr>
<tr>
<td>Glaunex</td>
<td>0.06</td>
<td>0.12</td>
<td>Flumazenil</td>
</tr>
<tr>
<td>Norephedrine</td>
<td>0.05</td>
<td>0.11</td>
<td>Nortriptyline</td>
</tr>
<tr>
<td>Isoephedrine</td>
<td>0.05</td>
<td>0.10</td>
<td>Bergapten</td>
</tr>
<tr>
<td>Corindolan</td>
<td>0.05</td>
<td>0.09</td>
<td>Cafestol</td>
</tr>
</tbody>
</table>

![Figure 3.9: Distribution of the percentage of shared peaks within spectra of a given compound, for the substances of the Weinmann/HUG spectral library.](image-url)
The easiest metric that can be extracted from this principle is the counting of peaks from an experimental spectrum that match peaks from the consensus signature.

The idea behind this metric is first to select peaks that are conserved among every spectra of a substance. Then, if the studied data set is representative enough, it is likely that these peaks will also be present in the acquisition of a new spectrum, if this spectrum comes from the same substance. This approach makes sense only if the library contains enough spectra to represent the various fragmentation patterns a chemical compound can produce.

In practice, only the three most conserved peaks were conserved for the calculation. The maximum score an experimental spectrum can obtain is 3.

3.3.3 Score combination: an exploration of Weka classifiers

Combining a targeted strategy with a global scoring is far from obvious. Rather than modifying the existing algorithm, three approaches were considered; a decision tree, a neural network and a supported vector machine.
In order to incorporate the consensus signature strategy to X-Rank, we employed Weka software.

We apply the three approaches and assess false negative and false positive compared to X-Rank alone. The three presented classifiers were trained with a test set consisting of 104 spectra acquired at the University of Geneva. Presented performance are obtained by applying classifiers on 312 spectra acquired at the forensic center in Lausanne identified against the Weinmann / HUG spectral library.

a) Decision tree

The decision tree is one of the simplest classifier provided by Weka. We applied this approach to the X-Rank scoring, completed by the signature score. Figure 3.11 shows the optimal tree conformation proposed by Weka, based on X-Rank score and number of shared peaks.

The confusion matrix generated by this decision tree is presented in Table 3.5. The classifier identified 243 matches as true (174 plus 69) and 69 (20 plus 49) as false. Among the proposed trues, 69 were wrongly classified, whereas 20 of the proposed false were in fact true.

When compared to X-Rank score, a threshold of 0.46 must be applied to obtain the same number of false positives (69). This generates 35 false negative results. From the false negative point of view, a threshold of 0.37 generates 20 false negatives and 89 false positives.

<table>
<thead>
<tr>
<th></th>
<th>Classified as true</th>
<th>Classified as false</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>174</td>
<td>20</td>
</tr>
<tr>
<td>False</td>
<td>69</td>
<td>49</td>
</tr>
</tbody>
</table>

Table 3.5: Decision tree classifier performance.

b) Neural network

Neural network analysis was the second classifier used to combine X-Rank score and the signature metric. Table 3.6 highlights performance of this classifier. 261 matches were identified as true (185 plus 76) and 51 (9 plus 42)
among the proposed trues, 76 were wrongly classified, whereas 9 of the proposed false were in fact true.

When compared to X-Rank score, a threshold of 0.434 must be applied to generate 76 false positive, with the consequence of generating 26 false negatives. A threshold of 0.329 must be set to produce 9 false negative, and consequently 107 false positives.

<table>
<thead>
<tr>
<th>Classified as true</th>
<th>Classified as false</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>185</td>
</tr>
<tr>
<td>False</td>
<td>76</td>
</tr>
</tbody>
</table>

Table 3.6: Neural network classifier performance.
c) Support vector machine

Support vector machine is the last tool of Weka that was used to classify identification. The confusion matrix generated by the SVN analysis is presented in Table 3.7. The algorithm labeled 274 matches as true (191 plus 83) and 38 (3 plus 35) as false. Among the proposed trues, 83 were wrongly classified, whereas 3 of the proposed false were in fact true.

When compared to X-Rank score only, in order to obtain the same number of false positives (83), a threshold of 0.406 must be applied. This generates 21 false negative results. From the false negative point of view, a threshold of 0.313 generates 3 false negatives and 118 false positives.

<table>
<thead>
<tr>
<th></th>
<th>Classified as true</th>
<th>Classified as false</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>191</td>
<td>3</td>
</tr>
<tr>
<td>False</td>
<td>83</td>
<td>35</td>
</tr>
</tbody>
</table>

Table 3.7: SVM classifier performance.

d) Scoring model enhancement performance

On the whole, the three presented approaches improved X-Rank performance. In terms of correctly classified instances, the neural network approach is the most effective (with 73 % of correctly classified instances), closely followed by the SVM classifier (with 72 % of correctly classified instances), and finally the decision tree with 71 % of correctly classified instances).

It is also important to notice that while improving false positive rate for example, our combination of metrics doesn’t negatively affect false negative rate. This non-degradation of the base scoring was one of our concerns when choosing a way to combine different discriminating values.

Table 3.8 summarizes performance in percent of each classifier compared to X-Rank scores for corresponding levels of false and true negatives. This comparison shows the best identification power using a SVM classifier.

Many other scoring function refinements are possible. The direction presented in this section represents a way of investigation that could be extended with an array of other potential improvements.

---

8LibSVM implementation was employed.
Table 3.8: FN gain: false negative decrease compared to X-Rank for an equivalent number of false positive. FP gain: false positive decrease compared to X-Rank for an equivalent number of false negative.

However, it is easy and dangerous to focus on a peculiar and difficult identification, forgetting the bigger picture of the many other easy identifications. In this context, it is not trivial to manipulate scoring improvements, without damaging global performance. Rather than modifying scoring functions to add new developments, we strongly recommend the use of a combination approach that take into account individual scoring metrics.

Chapter summary

In this chapter, we introduced two aspects: X-Rank and possible improvements.

1. **X-Rank** is a library-search identification algorithm based on a statistical model. This model involves rank of intensities, which is a more robust measure of intensity. X-Rank showed better performance compared to existing identification algorithms and clearly allows easier routine analysis when compared to its best available competitor.

2. **Improvements** are needed to address particularly challenging situations. One of these situations is the distinction of very close molecular structures. A more stringent metric was described, involving conserved peaks among spectra acquired from a given chemical compound. The problematic of combining such a metric with the existing scoring is also approached by using three ways: a decision tree, a neural network, and a supported vector machine. This last approach highlighted the best performance.
Small molecule LC-MSMS identification is often associated with sparse availability of spectral libraries. Though numerous large libraries exist in the world of GC-MS, only rare public initiatives are available for LC-MSMS. The main ones are briefly exposed in the introductory chapter of this thesis.

In addition to these main libraries, an array of individual laboratories build their own spectral libraries, whether to commercialize them or for
their personal use. Hans H. Maurer, Wolfgang Weinmann, Herbert Oberacher and their respective teams are the most visible examples of laboratories working towards the building of new LC-MSMS libraries.

Paradoxically, no tool seems to fulfill the needs associated with spectral library building. Consequently, there is a need for facilitating tools, that would allow to

1. import libraries,
2. explore them,
3. modify them,
4. share them.

The following sections describe our architecture of libraries, insisting on the chemical compound naming and the strategy deployed to adapt peripheral information to multiple applications. We also discuss strategies to enable library sharing.

In other words, the key motivation of this chapter is to approach one of the major weaknesses of LC-MSMS: the poor availability of spectral libraries.

4.1 Library architecture and interactions

This section describes what we define as a library. We then expose our strategy towards multi-application libraries and chemical compound naming.

4.1.1 Structure and definitions

In order to clearly expose our vision of spectral libraries, as well as interactions with them, let us define some preliminary elements. From the library point of view, a fragmentation spectrum $S$ is define as:

$$S = (P, C)$$

(4.1)

where $P$ designates a collection of peaks$^1$ and $C$ a collection of contextual information. $P$ and $C$ are respectively defined as:

$$P = \{(\omega_i, \zeta_i), i = 1...n\}$$

(4.2)

$^1$For information, these peaks correspond to $\{f_1, f_2, ... f_{size(S)}\}$ in section 3 dedicated to X-Rank.
4.1. LIBRARY ARCHITECTURE AND INTERACTIONS

\[ C = \begin{pmatrix} retentionTime_j \\
collisionEnergy_j \\
ionization_j \\
title_j \\
contributor_j \\
implement_j \\
implement_j \end{pmatrix}, i = 1..m \]  \hspace{1cm} (4.3)

where \( \omega \) and \( \zeta \) designate mass over charge ratio (m/z) and intensity values. retentionTime, collisionEnergy, ionization, title, contributor, and instrument are elements defining contextual information of the spectrum.

In the following sections, \( C \), represents the collection of contexts associated with spectrum \( S_i \).

With these definitions in mind, a library \( L \) is defined as:

\[ L = \{ (S_i, mol_i) \}_{i=1..|L|} \]  \hspace{1cm} (4.4)

where \( mol \) is the molecule linked to the spectrum. Note that \( mol_1 \) and \( mol_2 \) can represent the same chemical molecule so that \( S_1 \) and \( S_2 \) are generated from the same compound, possibly under different experimental conditions.

a) Library queries

We can now describe several interactions with these libraries.

It is possible to select spectra according to diverse criteria. We can define a subset of a spectra collection (\( S_c \)) belonging to a library (\( L \)) and acquired from a molecule (\( mol \)):

\[ S_c(L, mol) = \{ (s_i, mol_i) \in L | mol_i = mol \} \]  \hspace{1cm} (4.5)

Equation 4.6 expresses a more restricted selection by adding contextual information (context) to the query.

\[ S_c(L, mol|context) = \{ (s_i, mol_i) \in L | mol_i = mol, context \in C_i \} \]  \hspace{1cm} (4.6)

It is finally possible to enrich a library spectrum, by adding new contextual information to an existing spectrum.

\[ S_i \leftarrow (P_i, \{C_i, C'\}) \]  \hspace{1cm} (4.7)
4.1.2 Layers of information and multi-application libraries

Imagine a cocaine spectrum acquired at the University Hospital of Geneva (HUG), associated with a retention time of 3.7[min] under a given protocol. In parallel, the University Hospital of Vaud (CHUV) is interested in using this spectrum as a reference spectrum for its own investigations. However, acquisitions of cocaine lead to a retention time of 4.5[min] with the protocol used at this second hospital. The only way to benefit from the HUG acquisition consists in enriching the CHUV library with the information related to the CHUV methodology, keeping the spectrum (basically masses and intensities) of the HUG.

Layers of information is an approach that apply this principle, coping with application-specific content. It consists in differently considering areas of information related to spectra.

Part of this information defines the spectra itself and thus cannot be modified. This typically concerns peak intensities and mass over charge ratios. These values form the fragment ion peaks and are considered intrinsic to the MS acquisition. Changing them would mean changing the nature of data.

Other parts of information, such as the retention time value, are highly related to the analytical process. This kind of information strongly bind spectra to the context in which it has been acquired.

Finally, a last part of information is used to give extra material about spectra (e.g. its title, the person who did the acquisition, the compound to which it is linked, and so on).

To address multiple applications, the last two parts of information must be modifiable, in order to fit in with other analytical processes (e.g. a new LC column) and other annotation systems (e.g. the use of drug names instead of chemical compound names). The layer system allows to keep core information ($P$) intact, while adding layers of contextual information $C$.

Each of this layer is dedicated and selectively loaded for a given type of analytical method. Figure 4.1 is a schematic representation of this system.

The advantage of this approach is that acquisitions of reference spectra can be used in multiple context. For details about the design and the implementation of the library system, see section 6.2.2.
4.1. LIBRARY ARCHITECTURE AND INTERACTIONS

4.1.3 Chemical compound annotation

Another central aspect of libraries is the annotation of spectra with chemical information. Analysts seldom talk about spectrum themselves, usually preferring to mention their corresponding molecule. In addition, relying on a stable and comprehensive annotation system allows to search against multiple libraries, since spectra of these libraries become comparable (i.e. they are referred to by a common system of unique identifiers).

The problem is that chemical compound annotation is not trivial, since no thorough unique identifier really exists. Chemical Abstract Service (CAS) registry number and PubChem Compound Identifier number (CID) were two candidates for this role, but showed important flaws.

First of all, both identifiers fail to assign a unique reference to a chemical compound. For example, the CID 2244, corresponding to aspirin, points to a multitude of CAS\(^2\). On the other hand, amphetamine can be assigned to several CID\(^3\), for example.

In addition, the two identifiers fail to address specific situations such as isotop-marked internal standards. D5-Amphetamine and D3-Clomipramine

\(^2\) 11126-35-5, 11126-37-7, 2349-94-2, 26914-13-6, or 98201-60-6 are some of these CAS.

\(^3\) 3007, 222898, or 10220277 are some of these CID. In reality, only the first of these identifier corresponds to the amphetamine. Other identifiers are labeled amphetamine, but correspond to mixtures, in fact.
are two examples of frequently employed internal standards that cannot be bound to a chemical compound identifier.

As a result, an internal identifier system is used in the SmileMS platform. This allows to handle ambiguous annotations such as the previously mentioned aspirin and amphetamine. It also keeps the door open to other identifier strategies, based on drug names, for example. In order to attribute displayable names to chemical compounds, the system provided by PubChem is employed for the majority of chemical compounds. Cases such as internal standards are addressed by letting users define their own annotations. This would not have been possible if we had based our chemical annotations on PubChem or the Chemical Abstract Service only.

4.2 Library management and sharing

This section highlights the basic library management functionalities of our platform, as well as some related examples of applications. It also presents the concept of library sharing.

4.2.1 Library building and management

While developing SmileMS, we placed libraries at the center of the platform. We thus developed tools to create, visualize, update, modify and export libraries.

In the first place, these tools were internally used to test our scoring hypotheses and to serve as basic mechanisms to support other developments.

Quality management was another major use of library browsing and editing. By consulting spectral information, we, and later users, could evaluate the pertinence of this information. It was then possible to modify this information, in order to improve its accuracy or to adapt it to new applications.

The use of these functionalities in real applications is briefly illustrated by the four selected examples that follow. These examples demonstrate the applicability of our library structure to multiple applications, as well as the possibility to import libraries from various formats.
4.2. LIBRARY MANAGEMENT AND SHARING

a) Four selected examples

The HUG
Taking the example of the HUG, these capabilities were used to build an ion trap spectral library composed of 416 $MS_2$ and $MS_3$ spectra of clinical toxicology interest\(^4\). The platform was also used to correct annotations and to further enrich this library. Finally, the ion trap library was used with the QTRAP instrument, highlighting the advantage of a generic library format that enables the use of libraries with multiple instruments.

Centre Universitaire Romand de Médecine Légale (CURML)
At the forensic center in Lausanne, a formerly built library of 128 $MS_2$ was imported in SmileMS. So far, this library has been highly modified through the platform interface and has been used for more than 600 analyses.

Food control center
The food control center in Geneva is, among other, responsible for assessing the content in active substances claimed by drugs sold over the Internet. In this context, a spectral library consisting of 124 $MS_2$ spectra was imported and used via SmileMS.

Proteomics application
Finally, in his chapter dedicated to proteomics applications, Roman Mylonas achieved an identification comparison between SpectraST and SmileMS. This compared identification involved as many as 3'176 $MS_n$ peptide spectra.

4.2.2 Library sharing
As mentioned in section 2.2.2, sharing library data is an interesting way of coping with legal restrictions and cost limitations\(^5\). Let us recall the example of Geneva and the interest of sharing library data between the HUG and the food control department. Chemical compounds found in food control investigations became of interest for medical drug screening.

\(^4\)See chapter 8 for further details.
\(^5\)When a laboratory wants to build a library, it can be difficult to obtain some substances, because of their restricted access or due to their cost.
a) Basic sharing mechanism

The most basic sharing mechanism consists in being able to import and export libraries. Even though basic, this possibility enables to share reference data with the scientific community. It also enables to break application- or format-specific limitations, since users can import libraries from a specific format, update this library, and eventually share it with colleagues.

On the other hand, sharing libraries implies potential legal limitations\(^6\), as well as some technical issue, such as versioning and standards concerns.

Versioning issues

Versioning issues are a major concern of sharing mechanisms. Two levels of versioning are possible.

The first one is the library versioning, where each library possesses a version number. In this system, a newly imported library will replace information from a similar (maybe older) library. If not any older library exists, the new library will be imported directly. This model corresponds to the current status of all spectral libraries. NIST, for example, frequently updates its library, forcing users that employ a former version to replace their old library by the new one, if they are interested in the update.

The second level of versioning takes place at the spectra level. In this model, each spectra has an unique identifier. When importing a new spectrum, three cases exist:

- the spectrum does not exist in the library; it is directly imported,
- the spectrum already exists; it is not imported,
- the spectrum already exists, but has been modified by the user. In this last case a proper behavior has to be chosen. If modifications only concern extra-information, the newly extra-information is inserted as a new layer. If the modification is more profound, the user could choose to consider the new spectrum a different spectrum.

This second model of versioning answers to an important question: the stability of generated results. Suppose for example that an analysis was carried out using version 1 of a certain library, at the HUG. In this analysis, amphetamine appears as a detected compound, which leads to a certain

\(^6\) Certain libraries cannot be redistributed.
medical behavior. In an update to version 2 of the library, the amphetamine spectrum is modified. The HUG cannot afford to see its analysis modified or to lose the amphetamine identification if it re-analyses the same sample. For this reason, a versioning system must be implemented.

In the current version of SmileMS, newly imported libraries are directly registered as new libraries, without impacting on former ones. To search against older and more recent libraries, users can create profiles that specify old and new libraries. This possibility is ensured by providing inter-libraries compatibility. In addition, when adding a new spectrum to existing libraries, if this spectrum already exists, a contextual layer will be added to the existing spectrum, keeping track of the older information.

These two strategies present the advantage of avoiding synchronization issues. They also ensure the coherence of former results throughout time.

4.2.3 Formats support

The support of various library formats is another technical but mandatory step towards a usable library system. In order to be compliant with existing libraries, the platform provides readers for the most common library formats. These formats include CLQ and MDB (Applied Biosystems), MSP (NIST and Thermo Scientific), LIBRARY (Bruker Daltonics), and a large range of spectra formats.

Since the identification process of SmileMs is partly based on an aggregation system, one challenge resides in the mapping of reference spectra with the chemical compounds, while loading a new library.

This is achieved either by reading compound specific information where it exists (this is the case for formats proposed by Applied Biosystems for instance), or by using the spectrum name (this second technique is adapted to the LIBRARY format from Bruker Daltonics). Although these strategies showed good results where correctly annotated libraries are read, users have the possibility to change the aggregation attributes afterwards to ensure a correct behavior of the system.

7 Once a library is imported, compound annotation is the main piece of information that enables inter-libraries coherence.

8 see section 7.2.2, paragraph 7.2.3 for further details
a) Sharing platform

The term sharing platform designates a more advanced and dedicated tool, that would concentrate on sharing-related aspects only. The following sections present three concepts of sharing platforms.

Library repository

The first library sharing model is the repository model. This model can be compared to open-source repositories, such as the Advanced Packaging Tool (APT) system provided by Debian\(^9\).

The principle of these repositories is to offer a place to developers who want to share their project with the rest of the community. Users can install a software and then upgrade seamlessly, or roll back to a given version. Similarly, we could imagine a model in which MS actors could submit and retrieve spectra.

This kind of model relies on the voluntary sharing of information. Submitters should accept to give their libraries, without guaranteed return.

The book library model

An alternative to the repository model relies on considering spectral libraries as book libraries. Thinking about classical literature, one easily conceives that the ambition of gathering every existing book in his or her personal library is an illusion. Yet, to the contrary, it is fairly easy to access these books by borrowing them at the local library.

The same reasoning applies to spectral libraries. Instead of trying to gather the whole knowledge in-house, a specific request on relevant data is preferred in this model. Several spectral libraries, such as MassBank\(^10\) or METLIN [SMi05] already propose programatic access.

The main advantage of this model is first the access of very large knowledge bases, without needing extensive internal computer resources. It also transfers the versioning concern to the institute which provides the spectral library. Finally, bioinformaticians tend to have gained experience in database remote access through other applications, such as Genomics or Proteomics. This would probably facilitate the approach of such a system.

On the other hand, this approach deeply relies on external resources,\(^9\)See http://en.wikipedia.org/wiki/Advanced_Packaging_Tool for further details.\(^10\)Web address: www.massbank.jp/
which asks the question of trust in these resources. Particularly needs for reproducibility of former results and permanent availability of identification systems should be carefully studied. In addition, the need for infrastructure, such as efficient network connections, seems to be difficultly answerable at the time. Even though the situation will probably evolve towards a greater availability of internet connections, routine laboratories are for most of them isolated from the external world, nowadays.

The iTunes model

Finally, the iTunes sharing model represents a much more advanced model of data sharing. iTunes is a commercial software produced by Apple. It aims at providing a platform to share and sell media content. It takes the form of a repository where users can get (sometimes for money) entire or part of a singer discography, for instance. iTunes takes aspects such as versioning or intellectual property into account by implementing unique identifiers for every single item presented on this platform. Users who retrieve information from this platform are allowed to use this material only for granted applications.

Similarly, an equivalent platform applied to spectral libraries would represent an ideal situation for spectral data sharing. Such a platform would consist of a repository where people could upload and download libraries or individual spectra. The possibility of downloading individual spectra would be a potential solution for laboratories that cannot access specific pure substances, whether because of their price or because of restricted accesses. Uploads and downloads should also be possible in various format to enable multiple applications. Furthermore, peer-reviewing and rating of the quality of data should be implemented to provide users with a way to estimate data quality before downloading them. In that respect, advances done within this project in terms of data visualization and annotation are a step forward in this direction. Finally, financial models could be developed, in order to encourage people to share (or sell) their data through the platform, thus increasing the number of proposed spectra.

4.2.4 Prospect

The ideal systems probably lies between the iTunes model and the book library one. Indeed, it is highly convenient for performance and trust reasons to keep library data on-site. However, when chemical compounds are
missing and cannot be identified, it would be beneficial for the analysis to be able to access external resources to enlarge the search space.

The balance between internal and external data strongly depends on applications. At the HUG for example, a library of little more than 133 substances seems to cover more than 90% of analysis needs. This application would be more turned towards the iTusnes model which offers specific help for specific cases. On the other hand Serge Rudaz, from the laboratory of pharmaceutical analytical chemistry in Geneva, argued that only a few percentage of metabolic identifications are established against classical in-house libraries. This second example would be better addressed by the book library model or even more exploratory methods such as theoretical fragmentation.

**Chapter summary**

In this chapter, we introduced:

- the concept of spectral libraries as well as interactions proposed for these libraries.
- the choice of turning towards an internal system for a unique chemical identifier system.
- the importance and implications of an efficient library management tool, throughout four selected examples.
- divers models of library sharing, from a simple repository to much more advanced sharing platforms.
Chapter 5

Chemical Compound Fragmentation

"Almost all aspects of life are engineered at the molecular level, and without understanding molecules we can only have a very sketchy understanding of life itself."

Francis Crick

Contents

5.1 Context ................................................. 92
5.2 Fragmentation-energy driven fragmentation ............ 93
  5.2.1 Fragmentation pattern observation ................. 93
  5.2.2 Introducing fragmentation graph .................. 93
5.3 Software applications ................................... 97
  5.3.1 A visual fragmentation tool ......................... 98
  5.3.2 Fragmentation graph exploration .................. 98
  5.3.3 Fragmentation algorithm .......................... 99
  5.3.4 CDK Framework ................................ 106
  5.3.5 Achievement and prospect ......................... 107
5.1 Context

While peptide fragmentation is a well described process, small molecule tandem MS produces complex fragmentation patterns and is a lot more obscure. Hopefully, the new tendency to acquire chemical compounds under multiple fragmentation energies leads to a more global picture of molecular fragmentation patterns. Massbank\(^1\), for example, is a publicly available high resolution spectral library that proposes spectra acquired with different instruments and under multiple fragmentation energies.

A better understanding of mechanisms underlying the fragmentation of small molecules would enrich the identification process by several aspects. A first benefit resides in the automated annotation of fragmentation spectra (i.e. assigning a spectrum peak to a molecular fragment). Annotating peaks would mean that users could assess the pertinence of an identification. Furthermore, understanding these processes could play a much broader role in the future of identification platforms. For example, it could help refining identifications by eliminating candidates with incoherent peak patterns. It could also be useful in discriminating close compounds, such as metabolites of a same substance. Finally, mastering fragmentation mechanisms could lead to suggesting candidates when no match is found in the searched spectral library.

In the first part of this chapter, we present observations about fragmentation patterns, throughout multiple fragmentation energies. We also discuss how these observations can be helpful as a complement in an identification context. We then introduce the notion of fragmentation graph. This graph is a kind of genealogical representation of a molecule and its fragments.

In the second part of the chapter we present the development of a tool that helps visualizing and interacting with molecular structures. This tool is presented in section 5.3.1. We also describe another software application (section 5.3.2) that creates and displays the fragmentation graph of a molecule.

These investigations and developments were motivated by the fact that none of the tools\(^2\) discussed in sections 1.4.3 and 2.3 were suitable for integration in an identification platform. They were either not fast enough, not scalable, not integrable, not adapted to specific molecular structure format,\(^3\)

\(^1\) URL: www.massbank.jp
\(^2\) Mass Frontier, MS Interpreter, MS Fragmenter, and Fragment Identificator (FiD).
or not user-friendly. In addition, none of them can create a visual representation of a *fragmentation graph*. Building algorithms and software pieces were also crucial to understand the matter and test our hypotheses.

### 5.2 Fragmentation-energy driven fragmentation

This section describes a behavior that we observed between fragmentation energies and generated fragments. We will also introduce the notion of *fragmentation graph* through a manually constructed example, and correlate this example with our observation.

This approach is the preliminary study to development that aims at automatically generating *fragmentation graphs*.

#### 5.2.1 Fragmentation pattern observation

We first studied the influence of collision energy on fragmentation patterns, noticing a systematic behavior across spectra acquired under different collision energies. Figure 5.1 illustrates this behavior, showing six fragmentation spectra of the same molecule, acquired at increasing collision energies. When fragmentation energy increases, high mass peaks tend to disappear, breaking into lower mass peaks.

The observation illustrates that certain peaks are linked to others. For example, when peak of m/z 132 diminishes in intensity, peaks of m/z 130 and 117 increase in intensity. This behavior is particularly visible between 30 [eV] spectrum and 45 [eV] one, and then between this spectrum and 60 [eV] spectrum.

This mass transfer can obviously be explained by looking at the molecular fragmentation patterns. A molecular structure, corresponding to a peak, will be broken into sub-structures, corresponding to other peaks. This mechanism decreases the intensity of the first peak and increases those of the other peaks. The architecture of these pathways can be represented as a *fragmentation graph*.

#### 5.2.2 Introducing fragmentation graph

The next step of our investigation consisted in presenting possible fragmentation pathways of molecules, the nicotine, in our example. Taken together,

---

3This development is presented in section 5.3.
these fragmentation pathways form the fragmentation graph of a molecule. This fragmentation graph will be annotated with collision energy information later in this section.

By combining fragmentation information from the literature [Wil06, Smy07, Pel07], we obtained a simplified graph (Figure 5.2), expressed in terms of successive fragments. Figure 5.2 shows possible fragmentation states of the nicotine. In this graph, the nodes represent the molecular fragments and are annotated with the fragment mass. The edges represent the fragmentation events between two molecular structures.

We then plotted intensities of fragments (i.e. nodes of the graph) through successive fragmentation spectra and obtained the annotated graph pre-
Figure 5.2: reconstituted fragmentation graph representing major mass states obtained after fragmentation. The fragmentation information used to build this graph was gathered in the literature [Wil06, Smy07, Pel07].

Presented in Figure 5.3. This figure tells us, for example, that the peak of m/z 106 was present at fragmentation energies \(30[eV], 45[eV], 60[eV], 75[eV]\), and \(90[eV]\), with a maximal intensity at \(45[eV]\).

We notice that the behavior previously observed is now obvious. For instance, the precursor mass (163 m/z) strongly appears at \(15[eV]\) and \(30[eV]\) spectra and diminishes afterwards. In the same time, the fragment of 132 m/z is absent at \(15[eV]\), but appears and later disappears at and after \(30[eV]\). Finally, fragments 130 m/z and 117 m/z are a little more shifted in terms of fragmentation energy, with a maximum of presence at respectively \(45[eV]\) and \(60/70[eV]\).

Two observations can be made from Figure 5.3:

1. intensities of peaks follow a logical pattern throughout fragmentations at different energies,

2. a transfer can clearly be observed from parent nodes to children.
a) **Fragmentation graph and automated spectra annotation**

Since nodes of the fragmentation graph correspond to fragment of the molecule, this graph can be useful for automated annotation of spectra.

Let us consider the situation where a user tries to assign a molecule substructure of the nicotine to a fragment of m/z 132, acquired at a fragmentation energy of $30\, eV$. In this example, the task seems pretty easy since only one fragment corresponds to this m/z value. However, in more complex cases, this may not be the case, leading to a potential misinterpretation of the fragment associated to peak m/z 132.

One possible manner of confirming the assignment of peak m/z 132 to a molecular fragment consists in considering the context of the studied spectrum (*i.e.* spectra acquired under different fragmentation energies). If the mass pointed out really corresponds to the node 132 of our graph, we would expect increasing intensities of peaks m/z 130 and m/z 117 at immediate higher fragmentation energies. Similarly, we would expect a
5.3. SOFTWARE APPLICATIONS

high intensity for peak m/z 163, at lower fragmentation energies. Figure 5.4 presents the evolution of the intensities of peaks 163, 132, 130, and 117 in relation to fragmentation energies.

![Figure 5.4: evolution of selected peak intensities in relation to fragmentation energy.](image)

Nowadays, this contextual evaluation is possible thanks to experimental design including acquisitions at various fragmentation energies. As mentioned in the "Context" section of this chapter, Massbank provides such kind of spectra collections. The main issue remains, to our knowledge, the lack of software performing automated computation of fragmentation graphs.

### 5.3 Software applications

This section presents the two software applications we developed in the field of small molecule structure investigation. The first one is a visualization tool that helps understanding and manipulating molecule structures, while the second one aims at building fragmentation graphs such as the one presented in the previous section. Both tools are based on the Chemistry Development Kit (CDK) presented in section 5.3.4.
5.3.1 A visual fragmentation tool

It is interesting to note that no free or even affordable tool proposing an easy way to manipulate molecular structures seems to exist. Though not entirely open-source, software presented in the following sections can be obtained upon request.

In order to gain understanding in how a molecule is fragmented, we first developed a tool that helps visualizing and manipulating molecules. Figure 5.5 is a screenshot of this software tool. It allows to enter a molecule CID or SMILE, download it from Pubchem, and display the identified molecule. Users can then manually select sub-structures, displaying the sub-fragment as well as its formula, mass and other metrics.

Despite its rather simple features, this tool proved to be of great help in understanding what parts of molecules were more likely to fragment. In addition, it turned out to help in the manual process of spectra annotation.

Figure 5.5: web-based tool that helps visualizing and interacting with molecules basic fragmentation.

5.3.2 Fragmentation graph exploration

The next logical step was to develop a tool that generates graphs approaching Figure 5.2 and Figure 5.3. Figure 5.6 presents this software application.

---

4 Please address your requests to info@smilems.ch.
5 Terms CID and SMILE have been previously discussed in this thesis. A definition can be found in the glossary section.
Just as in 5.3.1, users can feed the system with a molecule CID or SMILE. The molecule is drawn and completely fragmented. The graph of successive fragments is shown on the right of Figure 5.6. In this model, rearrangements are not taken into account, though the system allows to implement fragmentation rules. In this instance, the tool does not allow to fragment aromatic cycles, for example.

Figure 5.6: fragmentation graph tool. Users can select a molecule by entering a Pubchem CID or a SMILE. The molecule is then highlighted, along with its fragmentation graph. Users can finally select nodes, which presents the concerned fragment.

Compared to the figures previously extracted from the literature, this graph presents far too many possibilities. This is due to the important number of predicted fragments that in fact do not exist. This behavior could be improved by adding rules to the system, although these rules could also be automatically learned from annotated spectra.

5.3.3 Fragmentation algorithm

In this section, we present the method used by the previously described fragmentation tool to produce fragmentation graphs. This approach, based
on a connectivity matrix, is very fast, has a low memory footprint, and can be enhanced by taking additional fragmentation rules into account.

a) Preliminary concepts and definitions

In this first sub-section, we define elements manipulated during successive fragmentations of a molecule. These elements are: a molecule (which is defined as a graph), a fragment (which is a sub-part of a molecule), a collection of fragments, and finally the fragmentation graph.

Molecule

First, a molecule can be represented as an undirected graph $G_m$, composed of vertices $V_{G_m}$ (Equation 5.1) and edges $E_{G_m}$ (Equation 5.2).

$$V_{G_m} = \{ V_i | i = 1...n_{G_m} \} \quad (5.1)$$

Each vertex $V_i$ from the collection $V_{G_m}$ represents an atom of the molecule. In practice, atoms of hydrogen are implicit, except when they play a role in the molecule conformation. $n_{G_m} (= |V_{G_m}|)$ is the number of atoms in $G_m$.

$$E_{G_m} = \{ E_i = (k_i, l_i, P_i) | i = 1...n; k_i, l_i \in [1, n_{G_m}]; l_i > k_i \} \quad (5.2)$$

Each edge $E_i$ from the collection $E_{G_m}$ represents a bond of the molecule. This bond is defined as an undirected link between $V_{k_i}$ and $V_{l_i}$ (which explains condition $l_i > k_i$). Each bond also contains a list of properties $P_i$ which are typically the order of the bond or its stereochemistry.

Individual fragment

Let $F$ be a fragment of $G_m$.

$$F = \{ \xi_1, ..., \xi_m_F | \xi_i \in [1, ..., n_{G_m}]; \xi_i \neq \xi_j \} \quad (5.3)$$

where $\xi_1, ..., \xi_{m_F}$ designate indices of $V_{G_m}$. The $\xi_i \neq \xi_j$ condition ensures that a single atom is not considered twice.

With these definitions, $SG(F, G_m)$ is a subgraph of $G_m$ which contains:

1. the vertices $\{ V_{\xi_i} | \xi_i \in F \}$,
2. the edges $\{ E_i = (k_i, l_i, P_i) | E_i \in E_{G_m}; k_i, l_i \in F \}$.

Where $SG(F, G_m)$ must be a connex graph. The second condition ensures that all relevant $G_m$ bonds are kept.
$SG$ could be extended by adding extra bonds to reconnect it. That would model fragment recombination. This step is beyond the topic presented in this chapter.

**Collection of fragments**
The complete collection of fragments derived from a molecule $F_{G_m}$ is expressed as:

$$F_{G_m} = \{ F_i | F_i \text{ is a fragment of } G_m; \text{ conditions } C(F_i, G_m) \text{ are satisfied} \} \quad (5.4)$$

Conditions $C(F_i, G_m)$ currently implemented are:

- bonds broken for generating the fragment are not part of an aromatic cycle,
- the cumulated mass of $F_i$ atoms is greater than a minimum value,
- the total number of bonds broken for deriving the fragment is below a given threshold.

These conditions tend to limit the graph complexity and to avoid unrealistic chemical configurations. Additional conditions could be added to this list.

**Fragmentation graph $G_f(G_m)$**

Finally, we represent fragmentation genealogy (the whole molecule being the root of this genealogy). In order to follow a fragmentation pathway, we need to track down the succession of fragmentation events (i.e. bond breaking in this limited model) and construct a graph of successive fragments or fragmentation graph $G_f$. $G_f$ is therefore naturally a directed acyclic graph (DAG). $G_f$ is not a tree as some fragments can be produced following different paths. For example, the $G_f(G_m)$ contains:

1. the vertices $F$, where $F$ is the collection of elements presented in equation 5.4,

2. the edges $\{(F_{ki}, F_{li}, B_i) | F_{ki}, F_{li} \in F; F_{li} \subset F_{ki}\}$

where $B_i$ represents broken bonds explaining the transition between $F_{ki}$ and $F_{li}$. $F_{li} \subset F_{ki}$ expresses the fact that a sub-fragment must be part of its parent.
b) Algorithm building fragmentation graph $G_f(G_m)$

This section presents the algorithm used to build the basic (i.e. bond breaking) fragmentation of molecules. The aim of this algorithm is to generate the fragmentation graph $G_f(G_m)$. Concepts are presented in pseudo-code format.

Preliminary objects and functions

First, a connectivity symmetric matrix $M$ is created. $M$ is of size $n_{G_m} \times n_{G_m}$, where columns and rows correspond to numbered atoms of the molecule. The content of $M$ represents bonds between atoms of the molecule. $M$ is a matrix of bits, where true represents a bond and false indicated that no bond link two atoms:

$$M[i, j] = true \leftrightarrow \exists (i, j, P) \in \{\text{Edges of } F\} \lor (j, i, P) \in \{\text{Edges of } F\} \quad (5.5)$$

From a programming point of view, bit matrices can be represented as arrays of bitsets in the major languages. This offers a light memory representation and fast low level primitives.

An example of such a connectivity matrix is presented in Figure 5.7. In this figure, $H$ atoms have been neglected. The matrix only shows true (i.e. 1) values. All empty cells must be considered false (i.e. 0).

In this matrix, elements are individually accessible, the vector of boolean (later referred as bitset) $M[i, j]$, or can be access as entire row $M[i, \ast]$. $M[i, \ast]$ represents the collection $(M[i, 1], M[i, 2], \ldots, M[i, n_{G_m}])$.

Several operations are also possible both on individual element or on vectors. These operations are typically operators: AND ($\land$), OR ($\lor$), and NOT ($\lnot$). In addition, the function $\text{indices}(x)$ returns indices of true element from a vector, e.g.:

$$\text{indices}\{1, 1, 0, 0, 1, 0\} \rightarrow 0, 1, 5 \quad (5.6)$$

Based on the connectivity matrix, as well as on the above-defined operation, two functions are designed to extract information from $M$:

1. function $\text{connectedNodes}(M, i)$ returns indices of all connected nodes (whether directly or indirectly) to the $i^{th}$ atom of molecule described by $M$, 

$$\text{connectedNodes}(M, i)$$
Figure 5.7: connectivity matrix of the nicotine molecule. In this figure, $H$ atoms have been neglected. The matrix only shows true (i.e. 1) values. All empty cells must be considered false (i.e. 0).

2. function \texttt{subGraphs($G_m$)} returns an array of arrays of indices representing the different (if multiple graphs exist) connex graph found in the matrix (i.e. a representation of a list of the disconnected fragments).

These two functions are described in algorithmic sections 5.3.1 and 5.3.2.

\begin{algorithm}
\textbf{Algorithm 5.3.1: CONNECTEDNODES($M, iNode$)}
\begin{algorithmic}
\State $toVisit \leftarrow false \times n_{G_m}$
\State $visited \leftarrow false \times n_{G_m}$
\# Two vectors of $n_{G_m}$ false values.
\State $bsToVisit[i\text{Node}] \leftarrow true$
\While{(any\_true\_value\_in(toVisit))}
\State $i \leftarrow first\_index\_of\_true\_value(toVisit)$
\State $toVisit \leftarrow (M[i, \ast] \land ! visited) \lor toVisit$
\State $visited[i] \leftarrow true$
\EndWhile
\State \textbf{return} indices(visited)
\end{algorithmic}
\end{algorithm}
Algorithm 5.3.2: \textsc{subGraphs}()

\begin{verbatim}
 toVisit ← true × nGm
 # A vector of nGm true values.
 while (any_true_value_in(toVisit)) do
   i ← first_index_of_true_value(toVisit)
   subgraphs.add(connectedNodes(i))
   for (j in subi) do
     toVisit[j] ← false
 return subgraphs
\end{verbatim}

In addition to the connectivity matrix and its operations and functions, we identified \textit{breakable bonds}. These bonds indicate points of the molecule that could be broken to potentially obtain sub-fragments. To be breakable, a bond must comply with some conditions. Conditions currently considered are:

- the bond is not part of an aromatic cycle,
- one of the two atoms of the bond is not a hydrogen.

We note $BB(G_m)$ the collection of these breakable bonds for the molecule represented by $G_m$.

\textbf{Calculation of the fragmentation graph} $G_f(G_m)$

The building process of $G_f(G_m)$ is presented in this section. Recall from the previously described connectivity matrix $M$ that such a matrix describe a molecule or a molecular fragment. Three basic functions are involved in this process.

1. the function $\text{createNode}(M)$ creates a new node in $G_f(G_m)$, representing molecular fragment described by $M$,
2. the function $\text{node}(M)$ returns the node of $G_m$ associated to the fragment $M$,
3. the function $\text{link}(M, M', bb)$ establishes a link between nodes associated to $M$ and $M'$. It is complemented by $bb$ describing which bond(s) was/were broken to derive $M'$ from $M$. 

We can now describe the heart of the fragmentation algorithm: the function $\text{break}(M, BB, Level)$. This function actually operates the breaking of fragment represented by $M$, in order to create new nodes and edges to $G_f(G_m)$.

**Algorithm 5.3.3:** $\text{BREAK}(M, BB, Level)$

```
for (l = 1...Level) do
    success_bb ← { }
    for (bb set of l bonds in BB) do
        # Loop over a combination of 1,2,...,l breakable bonds.
        if (∃ bb' ∈ success_bb | bb' ⊂ bb) do
            next
        
        M' ← disconnect(M, bb)
        # Set falses in the connectivity matrix.
        sg ← subgraphs(M')
        if (|sg| = 2) do
            # The molecule is broken in 2 connex parts.
            success_bb.add(bb)
            BB' ← BB \ bb,
            # bb will not be taken into account later.
            for (M_F in sg) do
                if (not ∃ node(M_F)) do
                    break(M_F, BB', Level − l)
                    # Recursively break the fragment if it has
                    # not yet been taken into account.
                    link(M, M_F, bb)
```

Finally, the whole fragmentation graph is built by calling:

$$\text{break}(G_m, BB(G_m), Level_{max})$$ (5.7)
5.3.4 CDK Framework

Tools presented in section 5.3.1 and 5.3.2 use the Chemistry Development Kit\(^6\) (CDK) [Ste03b, Ste06] as their skeleton. Among others, CDK provides developers with the following functionalities, that were useful in our project.

- reads common molecule file formats (e.g. SDF\(^7\)),
- reads SMILE\(^8\) string formatted molecules,
- computes dozen of metrics (called descriptors in CDK) for atoms and groups of atoms,
- computes 2D and 3D coordinates of atoms within a molecule,
- renders images (e.g. PNG format),
- handles implicit and explicit \(H\),
- provides a complete description of atoms and bonds,
- provides information about bonds (e.g. is it a simple, a double or a triple bond?) and about their position (e.g. is this bond within an aromatic cycle?),
- proposes functions to handle molecules.

CDK is also an open-source initiative that counts almost 50 frequent contributors. It is an active project and can be considered one of the most complete development framework for chemistry-related development\(^9\).

Finally, is must be mentioned that CDK proposes a fragmentation tool developed by Rajarshi Guha\(^10\). Unfortunately, this small software application is very slow and suffers from a lack of scalability (i.e. large molecules leads to extensive calculation time and memory overflow). This tool is also difficult to integrate into other applications and is not adapted to exploratory work (mainly due to the previously mentioned limitations).

---

\(^6\)The Chemistry Development Kit is an open source Java library for Chemoinformatics and Bioinformatics. See [http://sourceforge.net/apps/mediawiki/cdk](http://sourceforge.net/apps/mediawiki/cdk) for further details.

\(^7\)SDF is an open standard text format that describes molecules and their structure.

\(^8\)The SMILE or simplified molecular input line entry specification is a specification for unambiguously describing the structure of chemical molecules using short ASCII strings. This nomenclature is at the origin of the name SmileMS.

\(^9\)CDK is an equivalent of bioperl for sequence analysis, for example.

\(^10\)see [http://rguha.net/code/java/#frags](http://rguha.net/code/java/#frags)
5.3. SOFTWARE APPLICATIONS

5.3.5 Achievement and prospect

Although we did not contribute significantly to the understanding of fragmentation, the observations and the developed tools and algorithms are one step ahead this long process.

The presented algorithmic approach appears efficient enough to permit a routine exploratory use of the results generated by the software. In addition, our development takes advantage of the publicly available CDK framework and is easily integrable. This development should be considered a starting point to further developments in the direction of better understanding and using chemical compound fragmentation. For example, edges of the fragmentation graph could be associated with probabilities (or a notion of cost) to break the related bonds, creating likely and unlikely paths throughout the fragmentation graph.

It is also interesting to consider this development within the global picture of our small molecule identification platform. If it took advantage of library mechanisms, for example, we could imagine a system in which users would manually annotate library spectra using our visualizing tool. These annotations could be further used to train an improved version of our fragmentation engine.

Chapter summary

In this chapter, we presented:

1. observations about molecular fragmentation patterns; we also briefly discussed potential implications of this observation and of a better understanding of chemical fragmentation,

2. a simple molecular visualization tool, based on CDK framework, available upon request,

3. a more advanced software application that fragments a molecule and constructs its fragmentation graph. This software piece was conceived as an exploratory tool that allows developer to implement their own fragmentation rules,

4. the underlying algorithmic concepts of our fragmentation software, paving the path for further investigations.
Chapter 6

Exploratory software engineering

“There are two ways of constructing a software design: One way is to make it so simple that there are obviously no deficiencies, and the other way is to make it so complicated that there are no obvious deficiencies. The first method is far more difficult.”

C. A. R. Hoare

Contents

6.1 Context ................................................................. 110
   6.1.1 Engineering of scientific exploration ........ 110
   6.1.2 Dry and Wet cohabitation .................. 111
   6.1.3 Classical software engineering methodologies 112
6.2 SmileMS software development .................. 116
   6.2.1 Preliminary steps ................................. 116
   6.2.2 General design .................................. 119
   6.2.3 Scoring model and algorithmic design .... 131
   6.2.4 Implementation ............................... 132
   6.2.5 Development and build environment .... 138
6.1 Context

Just as standard software do, bioinformatics software seems to suffer from common software engineering pitfalls. Particularly, it often fails to meet requirements of several types of users, are difficult to interface or to make evolve, and are old-fashioned in terms of technology and interactivity, for example.

In the following sections, we describe a challenging aspect of SmileMS development: its interaction with scientific exploration. We also introduce software engineering methods, as well as the core modules of the platform.

6.1.1 Engineering of scientific exploration

Our project highlights a crucial aspect of bioinformatics development: the scientific, or data analysis, perspective. Another central aspect of such kind of development is the need for adaptability of the development process itself, but also of the produced software.

First of all, since one of the core pieces of the platform is its identification scoring, high performance of this part of the software must be guaranteed and maintained throughout the evolution of the software.

In addition, the creation of scoring metrics and other data-based developments\(^1\) implies exploratory phases. These phases ask for specialized engineering approaches and are difficult to quantify, in terms of resource and outcome. The continuous exploration also implies extensive validations with users and tests of performance for newly developed algorithms.

Moreover, scientific exploration is often coupled with graphical representation of data\(^2\). In that respect, a pertinent platform should support such visual exploratory integrated modules.

\(^1\)Our investigations on molecular fragmentation is another example of data-based development.

\(^2\)Again the fragmentation of molecules is a good example of investigations that need graphical representation.
Finally, like many other software applications, SmileMS is used and developed in a changing environment. Every new scientific challenge leads to exploratory phases and potential changes in the software. This aspect increases even more the need for adaptability.

a) Production of scientific advances

The transition from scientific exploration to production code was also pondered. In that respect, we retained three elements to consider:

- **From prototype to product**: transforming data exploration and algorithmic models into production software is one of the challenges.

- **Keep production objectives**: on the other hand, data exploration must be done without being overwhelmed by off-target considerations. In that regard, the balance between scientific investigations and production must be carefully set, since the two elements are indissociable in this project.

- **Further investigate**: it is often needed to stop production of code and investigate encountered issues. This unforeseeable dimension has to be integrated in development plan.

6.1.2  *Dry and Wet* cohabitation

Beyond the organization of the scientific side of the development, our project originates from a group of the University Hospital of Geneva (*HUG*), set to define a new analytical method for the identification of small molecules\(^3\).

Our early implication within this project created a continuous exchange between our team (the *dry*-lab part of the project) and researchers at the hospital (the *wet*-lab part) (Figure 6.1). As a result, elements of the analytical methodology developed in the *wet*-lab constantly influenced our choices, while our investigation on experimental data provided clues as to how the analytical method could be improved to generate clearer results.

This positive feedback loop oriented experimental acquisitions towards targeted screening, for example. After comparing identification results from *MRM* and *GUS* data, the first type of acquisition seemed to better correspond to what the *wet*-lab team expected. Interestingly, this led us to

---

\(^3\)See section 8 for more details.
also consider the GUS perspective, improving our management of extensive sets of experimental data generated by this technique. This example clearly illustrates our respective influence: each side taking into account tasks of the other and adapting development to permanent changes.

In addition to the close collaboration with the HUG, other collaborations showed the same dynamic. The forensic center in Lausanne is another example of such collaboration. This example is further described in the thesis of Roman Mylonas [Myl10], chapter 5.

### 6.1.3 Classical software engineering methodologies

In order to position our approach within existing methodologies, the following paragraphs describe classical approaches in software engineering.

#### a) Software process models

A software process model is the set of activities\(^4\) leading to the production of a software product [Som06]. The most famous software process models are the waterfall, the iterative development, the prototyping, the exploratory model, the spiral model, and the reuse model [CTG98].

\(^4\)Typical activities are: specifications, development, validation, and evolution.
The waterfall model

The waterfall model is a sequential model which takes five fundamental activities, visited as a downwards flow (Figure 6.2) [Som06].

[Diagram of the waterfall model]

Figure 6.2: the waterfall model takes five fundamental activities and represents them as a downwards flow. (Figure retrieved from [Som06])

Iterative development

The iterative development is a cyclic model where process activities are repeated in response to external pressures and change requests [Lar03] (Figure 6.3). This method was initially conceived to address weaknesses of the waterfall model, particularly its lack of adaptability to change. Agile methods (developed later in this chapter) are part of the iterative development family.

Prototyping

Prototyping is a model based on the assumption that requirements at the beginning of a project are often difficult to determine. Usually, a simplified version of the proposed system is presented to customers in order to obtain feedback and to refine this system [CTG98]. In our explorations, once a direction seemed promising, we used prototyping to establish a first draft of functionality, coded in a language different from the production language.
CHAPTER 6. EXPLORATORY SOFTWARE ENGINEERING

Figure 6.3: the iterative development is a cyclic model where process activities are repeated in response to external pressures and change requests. (Figure retrieved from [Spe05])

The exploratory model

The exploratory model is a model for which no precise specification is made, due to poor or absent identification of requirements. A first assumption is made on how the system should be developed. Rapid iterations are then achieved to quickly incorporate suggested changes [CTG98] [Sam01]. In the scope of our project, this model is used to explore new scorings, chemical fragmentation theories, and other data-related situations.

The spiral model

The spiral model is a risk-based model that incorporates elements from the prototyping approach and the waterfall model. A first prototype is conceived and then improved, with each consecutive step being designed according to the waterfall methodology (see Figure 6.4) [Boe88].

The reuse model

In the reuse model, systems are constructed from existing components rather than from newly developed ones [Som06]. Existing components can be either software pieces already developed by the development team, open-source projects, or commercial initiatives. This model is central in the modularization of the development presented in the following sections.

Combination of models

Combining models is a common software engineering practice. Since process models were designed for specific environments, new conditions often require combinations of models or even the creation of new ones. Organi-
6.1. CONTEXT

The spiral model is a risk-based model that incorporates elements from the prototyping approach and the waterfall model.

Organizational environment as well as the type of application are the two main aspects to identify, when selecting a process model.

b) Agile methods

Agile methods is a popular family of methods that follow the agile philosophy [Sho07, Aoy98]. Agile methods combine elements from the previously described process models, insisting on recommendations including version control, coding standards, and weekly demonstrations to stakeholders.

The "Manifesto for Agile Software Development" describes the key strengths of agile development. This methodology first emphasizes on individuals and interactions over processes and tools. It also supports working software over comprehensive documentation and customer collaboration over contract negotiation. Finally, agile aims at responding to change rather than following a plan. This last aspect is probably the most important reason why we adopted agile methodology for the development of SmileMS.

Numerous agile implementations were developed, among which extreme programming and scrum seem to be more widely adopted [Dee10, Cop01].

---

5Mainly users, but also other stakeholders.
Our approach
As suggested in the process model presentations, the development presented in the following sections follows a logic a model combination. Prototyping and exploratory model were combined for data investigation, whereas others combinations such as agile methods and the reuse model served the more global development of SmileMS.

6.2 SmileMS software development
Congruently to the first claim of the agile manifesto\(^6\), recent years have shown that successful software projects centrally involve end users in their conception [Dee10]. In other words, identifying stakeholders and determining fundamental requirements with them is crucial. Coming from people, these requirements are likely to evolve over time, again implying that the software engineering methodology must be adaptable to this change.

In SmileMS, a typical example is the scoring model itself. Initially adapted to a very specific set of data, improvements and many laboratory collaborations made it more broadly adaptable. Under the influence of these laboratories, the representation of scores also went from an abstract open scaled number to a value between 0 and 1. These two examples are only a fraction of changes undertaken, but they show how adaptability and people proximity are crucial to SmileMS.

In the following sections, we expose how we tried to avoid common software engineering pitfalls. We also describe the main parts of SmileMS architecture.

6.2.1 Preliminary steps
a) Identifying the stakeholders
The first step in our methodology was to better understand, beyond the large scale picture, our environment and people who would interact with SmileMS. Stakeholders belong to various classes, depending on their interactions with the platform:

- **Users of the identification platform**: they have different skills (from a routine technician to an expert in mass spectrometry), different ad-

\(^6\)Individuals and interactions over processes and tools.
ministrative functions (from the basic end user to the IT specialist), and are from different domains (mainly from clinical toxicology, but also from forensic, food control, general toxicology, or environmental chemistry).

- **Institution using the platform:** they are also from various domain and various countries. The University Hospital of Geneva, the forensic center in Lausanne and the toxicology laboratories of Professors Hans H. Maurer and Wolfgang Weinmann in Germany are among this category.

- **Collaborations and integrations:** they mainly concern MS constructors. Currently, Bruker Daltonics and Applied Biosystems represent the two major actors of this category.

- **Development team and product management:** this category is composed of people developing and managing the platform, at Geneva Bioinformatics.

- **external sources:** they are academic and highly related to the project with the involvement of the Swiss Institute of Bioinformatics, for example, or more external such as Proteowizard.

- **Other groups that develop similar systems:** this category gathers academic or commercial initiatives that develop identification platforms or algorithms.

  An example of such initiatives is the research group of Herbert Oberacher, at the Hospital of Innsbruck, which develops an identification algorithm for small molecules [Obe09a, Obe09b].

b) **User requirements**

The identification of requirements is very important, since stakeholders are the ones who hold the knowledge, the culture, the questions and the challenges encountered in the field of small molecule identification. Most of the users’ requirements were learned from our two closest collaborations; the University Hospital of Geneva (HUG), and the forensic center in Lausanne.

- **Objective:** the system is to be routinely used in production from the first quarter of 2010.
In the case of the HUG, this is mandatory, since the platform aims at replacing a part of its existing emergency unit identification system. At the forensic center in Lausanne, SmileMS should contribute in a move from GC-MS platforms to LC-MSMS ones.

- **Utilization:** the platform must comply with multiple and environments (i.e. other than the clinical toxicology) and user profiles.

  For example, the HUG, distinguishes two user levels: technicians and administrators. The first category has to rapidly access analysis reports, while the second is concerned by all aspects of the platform, from libraries administration to the setting of users.

- **Environment:** due to the heterogeneity of MS instruments, the platform must be adapted to different systems.

  Again, taking the example of the HUG, two instruments from two constructors and two technologies are employed. This aspect is further described in section 8.

- **Remote access:** identification software tends to be installed on the acquisition computer of MS instruments. Though it is usually preferred to review results in a quieter environment, at home for instance. To allow this behavior, SmileMS must be accessible both locally and remotely.

c) **Performance requirements**

In addition to usability requirements, we identified requirements related to performance.

- **Identification efficiency:** the pertinence of the identification is one of the main performance requirements. The identification must be discriminant and performance not negatively affected by future development\(^7\).

- **Speed and memory usage:** with the size of libraries and experimental data increasing, tens of million of comparisons\(^8\) must be achieved in a reasonable time and space.

\(^7\)Non regression of performance is typically addressed by unit testing (see section 6.2.6)\(^8\)A MS run that generates 1’000 spectra, identified against a medium library containing 10’000 spectra generates 10 million of comparisons.
• **Reliability**: To comply with routine use, the system must be accessible at any time. Emergency use of SmileMS is even more representative of this point.

d) **Production requirements**

Finally, unlike isolated scripting solutions, building a software product implies production requirements. Three major aspects of these requirements are: a clear deployment plan, test and maintenance procedures, and a versioning system which keeps track of changes and software versions.

### 6.2.2 General design

The goal of software design is to answer requirements by designing a coherent software system.

a) **Multi-tier architecture**

SmileMS platform presents four layers: *data*, *business logic*, *web service*, and *interface*. Figure 6.5 shows this layers in respectively green, red, yellow, and blue.

This figure also highlights packages and interactions between these packages. Not all interactions are represented to keep the figure readable. For example, the *Manager* package interacts with numerous packages. Similarly, packages on the left of the figure (*Access, Licence, Tag, Exception, Helpers*) tend to play a transversal role.

Set of packages can be further classified as taking part in global mechanisms. In Figure 6.5 a subdivision is proposed with groups related to:

- user and security,
- Tagging
- analysis workflow,
- library sharing

Except for the tagging system\(^9\), parts involved in these macro-functions are detailed in the following paragraphs and in section 6.2.4 dedicated to implementation.

---

\(^9\)Tagging system was judged not important enough to be detailed in this document.
Figure 6.5: SmileMS platform presents four layers: the data layer, the business logic layer, the web service layer, and the interface layer. Packages are represented taking part in a global function and according to their involvement in the different layers.
6.2. SMILEMS SOFTWARE DEVELOPMENT

User interface (UI) layer

In Figure 6.5, the interface of the platform follows a logic of components and views. Each function is handled by a specialized component (e.g. Server configuration, JobViewer, ResultViewer, Submission, and Library sharing and browsing). This ensures a clear separation of tasks and renders the maintenance easier. The modularization also permits a dynamic evolution of the platform.

In addition to interface modules presented in Figure 6.5, specific interfaces exist. This is the case for the fragmentation project and exploratory initiatives.

Web service layer

The web service layer is a web-accessible translation of business logic functionalities.

The MVC model (Figure 6.6) plays a central role at this level. In this model, requests sent to the web service layer are not directly handled. Instead, they are received by a controller which asks business services to make an action. Subsequently, the result of this action is exposed in a view.

Figure 6.6: the MVC pattern and interactions between its elements. (illustration retrieved from [Sun10b])
The sequence diagram presented in Figure 6.7 shows interactions between the interface and the web service layer during the submission process. Firstly, a job is requested to the ManagerController which delegates this action, creating a new ToxicoJob. Secondly, data are submitted to the web service layer, which consequently updates the ToxicoJob. Finally, the identification is requested, by having ManagerController start a new identification through SubmissionEngine (in red on the figure).\footnote{Steps related to the SubmissionEngine are detailed in section a).}

![Sequence diagram of interactions between the interface and the web service layer during the submission process.](image)

**Business logic layer**

The business logic layer is situated between the data layer and the web service layer. In this layer, data are manipulated and converted into content
suitable for presentation [McL02].

Figure 6.8 completes Figure 6.7\textsuperscript{11}, presenting the submission engine. This entity is responsible for handling the actual identification of experimental data.

When an identification is started, a cascade of creation is executed, ending with the creation of an engine, that handles scoring(s) and library spectra. The submissionEngine then provides experimental data and a set of parameters (i.e. the profile) to use. Finally, submissionEngine requests the generation of the identification. As a result, submission operates a license verification to ensure that the application is genuinely used; If so, submission asks the engine to generate the identification.

During the identification, the engine analyses each pair of library and experimental spectra. The first step in this process is to verify that a comparison is worth doing. To achieve this verification, we implemented two optional Filters:

1. an alignment of spectra converted into bitsets must lead to a minimum number of matches defined by a threshold parameter\textsuperscript{12},

2. a filter on a precursor mass window\textsuperscript{13}.

If a pair of spectra passes these two filters, it is scored.

Generated scores are stacked in a list, along with information regarding the pair of spectra. This information is formatted into a result by the engine and reported to the submission.

Multiple scoring models can be used at the same time. Scores are then reported for each scoring model.

To achieve this concurrent scoring, scoring models implement a common interface that defines mandatory interactions for a scoring object (e.g. generate and return a score). Adding new scoring doesn’t imply modifying the code, since the dependencies injection provided by Spring Framework allows external setting of this parameter\textsuperscript{14}.

Multiple scorings make model combinations, as well as comparisons of scoring models performance during exploratory phases easy to achieve.

\textsuperscript{11}SubmissionEngine (in red on the figure) correspond to the same entity in Figure 6.7.

\textsuperscript{12}Further details on this approach can be found in [Myl10], section 2.4.

\textsuperscript{13}Note that this precursor mass window filter is different from the one applied \textit{a posteriori} during the identification. Contrary to its \textit{a priori} version, it cannot be removed once set.

\textsuperscript{14}See paragraph "Dependencies injection" from section 6.2.4 for further details.
CHAPTER 6. EXPLORATORY SOFTWARE ENGINEERING

Figure 6.8: sequence diagram of the identification process business logic layer. After an identification is started, a cascade of creation leads to the creation of *ScoringModel* object that actually operates the matching of spectra. The result of this matching is then reported up to the *SubmissionManager*.

Data layer

"The foundation of any application is the data that it contains and utilizes. Without data, there is no need for an application!"[McL02]. This section discusses data structures used in SmileMS.

Storage

In total, four types of data need to be stored:

- data related to fragmentation spectra and libraries,
- data related to chemical compounds,
- data related to identification results,
• and administration of the server; mainly users dans their privileges.

Information is stored on two different supports: a relational database and XML files. Data stored in the database are frequently accessed (i.e. spectra and libraries, compounds, and administrative information), while information stored in XML files is less often accessed and can be of important size (i.e. it mainly concerns identification results).

**Fragmentation spectra and libraries**

Spectra and library structures are presented in the same section since they share most of their content. Some classes presented in this section belong to the JPL project. Additional information about this project is provided in section 6.2.4.

Figure 6.9 is a relational scheme of spectrum and library object. In this system, a LC-MSMS runs (JPLRunLcmsms) contain 0 or more chromatographic peaks (JPLMS1LCSpectrum) and 0 or more peak lists (JPLPeakList). A chromatographic peak provides information regarding its retention time (JPLRetentionTime). A peak list contains descriptive information (i.e. a title, the number of peaks, a MS level) and references to lists of masses (JPLPeakList_mzs) and intensities (JPLPeakList_intensities). Peak lists contain 1 or more set of spectrum-related information. Finally, peak lists can also point to a filtered version of themselves (JPLFilteringResults) and a parent peak (JPLMSPeak).

Library structure largely overlaps with spectra data structure. It is constituted by a DataSource, which gathers sets of MsSpectrumInfo and subsequent peak lists.

The one-to-many relationship that links peak lists to additional information allows using specific sets of parameters per spectrum, depending on the targeted application. As explained in section 4.1.1 paragraph 4.1.2, this approach is part of the solution proposed for multi-application libraries.

**Chemical compounds**

As mentioned in section 4.1.1, spectra are bound to chemical compounds. Figure 6.10 describes the data structure of chemical compounds.

---

Information related to the acquisition instrument, the contributor, the library to which the spectra belongs, and so on.
Compounds are uniquely defined (COMPOUND_ID) and are linked to collections of descriptive entities (i.e. descriptors (CompoundDescriptor), names (CompoundName), and reference to other systems (CompoundXref)). They also possess a $\text{CID}^{16}$ value (COMPOUND_CID) that is used to retrieve specific information through Pubchem services.

---

16See section 4.1.1 paragraph 4.1.3.
Identification results

Results are saved so that they can be subsequently reviewed. A result contains information regarding the job, the parameters, library and the experimental spectra references, as well as scores generated by the different scoring schemes.

Information related to the identification job (e.g. date, title, ...) is stored in the database, whereas actual identification results (e.g. list of identified compounds) are saved in a XML flat file.

The XML format presents the advantage of being readable with the naked eye and easily parsable by third-party scripts. Results are also much bigger and less frequently read than spectra. As a result, they do not necessitate a database storage.

Figure 6.11 shows an example of XML result.

Users and rights

Users and their privileges are accessed throughout the software application to determine whether a user is granted to access objects, methods, or entire portions of the application.
Figure 6.11: XML structure SmileMS results. The XML provides information about the job, the experimental and reference spectra, as well as about the scores generated by each used scoring.

The model presented in Figure 6.12 utilizes concepts from Spring Security\textsuperscript{17}. Users (\texttt{USER\_USER}) are defined as belonging to a group (\texttt{USER\_GROUP}) which can also belongs to other groups. Groups are then associated a role (\texttt{USER\_ROLE}).

Each user is associated a security identity (\texttt{acl\_sid}) which will be used to specify user rights on objects. Securable objects are defined as \texttt{acl\_object\_identity}'s, and are owned by a \texttt{acl\_sid}. Each right defined for this object is repre-

\textsuperscript{17}See section 6.2.4 for more details.
sented as an acl_entry, which basically defines the user to which the right is granted and the level of this right (e.g. read, write, delete, and admin).

Figure 6.12: data structure of users and their privileges. This system is largely inspired by the one proposed by Spring Security.

Object Relational Mapping (ORM)

Aforementioned entities must be storable, readable, deletable, and updateable. Since the support for data storage is not the same for every entity\(^\text{18}\), we opted for a Data Access Object (DAO) approach. DAOs implement the access mechanism required to work with the data source. The data source can be a relational database, a LDAP repository, a flat file (e.g. XML file), or whatever persistence support[Sun10a]. Interactions between a DAO and its environment are described in Figure 6.13. Basically, a business logic object instantiates a DAO which accesses the

\(^{18}\text{And is likely to change throughout the project life cycle.}\)
data source, whether it is a file, a database, or any other source of persistence. The DAO then creates a TransferObject, which contains information retrieved from the data source\textsuperscript{19}. This object is returned to the business logic layer. Any update of data passes through the DAO.

Figure 6.13: representation of a Data Access Object (DAO). (Figure retrieve and adapted from Sun website, at the following address: [Sun10a])

In the context of this project, DAOs concern three types of files (i.e. raw spectra files, property files, and XML files) and a relational database.

\textsuperscript{19}In java, the TransferObject is often called a POJO, which is the acronym for Plain Old Java Object.
6.2.3 Scoring model and algorithmic design

A specialized and important part of the design relies on how to manage algorithmic aspects, and more broadly data analysis. This rises the concern of managing explanatory aspects in projects involving scientific research.

a) Exploratory and integration environment

The first strategy we employed relies on principles of modularization and components proposed by agile.

As previously mentioned, a minimal definition of input and output parameters of scoring models was proposed. New scoring models have to comply with this basic definition to be used in the context of SmileMS. If they do, the system will easily incorporate them, providing mechanisms to manipulate experimental and reference spectra.

This strategy allows:

1. creating an environment for exploration and performance probing of new scoring models,
2. rapidly integrating successful explorations.

This principle of modularization was also applied to spectrum fragmentation and other prospective aspects.

b) From prototype to production

Once the environment set, two phases are observed during the development of new scorings:

1. a highly exploratory phase, where most attempts fail,
2. a production phase where only successful approaches are retained and implemented.

At this stage, the prototyping model is employed. As explained in the introductory part of this chapter, prototyping is useful for situations in which requirements are difficult to determine. In the case of a scoring model, the only clear requirement is the improvement of the identification power, whether globally or for specific cases.

To achieve this goal, prototypes are tested in Perl and R. During this phase, the exploratory model is followed. In this second model, a first idea
is tested and further improvements or modifications are implemented, depending on obtained results. This cycle is repeated until a seemingly efficient solution is found, based on its benchmarking against the production scoring.

Prototyping in a programming language different from the production one ensures that successful attempts are re-coded and not copied from exploratory development. In addition, the knowledge gained during the Perl and R exploratory phase is used to produce tests that will ensure proper behavior of the re-coded scoring model. Finally, Perl and R are highly adapted to data manipulation and exploration.

Once several scoring models have been prototyped and tested, successful ones are implemented in Java to be part of the production code.

Again, this process is applicable to other explorations than scoring.

6.2.4 Implementation

After a first design stage, comes the implementation of designed concepts. This section discusses implementation aspects of the platform. It does not aim at describing every piece of code, but rather at mentioning important aspects of the four aforementioned layers: interface, web service, business logic, and data.

a) User Interface (UI) layer

The interface layer is developed in Adobe Flex. Adobe Flex is a software development kit released by Adobe Systems for the development and deployment of cross-platform rich Internet applications based on the Adobe Flash platform [Wik10a]. As presented in Figure 6.14, this technology is well adapted to graphical representation. It is also partially\(^\text{20}\) classified as an object oriented language and proposes easy integration of web services. Finally, Adobe Flex runs in a virtual machine thus showing similar aspect and behavior on all web browsers.

These characteristics make of Flex a programming language greatly adapted to the presented platform, as well as to exploratory aspects needing graphical representations.

Before opting for Adobe Flex, we investigated the javascript/Java framework Google Web Toolkit (GWT) provided by Google. This option proved

\(^{20}\)Despite an important effort, some paradigms of oriented programing are not implemented in flex. The absence of abstract classes is an example.
6.2. SMILEMS SOFTWARE DEVELOPMENT

Figure 6.14: examples of Flex development. Top, left: representation of a fragmentation graph. Top, middle: exploration of a molecule structure. Top right: visualization of a fragmentation spectrum. Bottom: interactive chromatogram, with dots representing the different identifications.

to be less mature and not similarly interpreted by the panel of existing web browsers. After some prototypes in GWT, we decided not to use this technology.

b) Web service layer

Recall that the web service layer is the portion of the application directly accessed by the external world (i.e. the user interface in most cases\textsuperscript{21}). Web service development follows the MVC framework provided by Spring framework\textsuperscript{22}.

\textsuperscript{21}Web services are also accessed by the light client (section ??) and the integration with Bruker Daltonic software (section 7.4), for instance.

\textsuperscript{22}Spring framework is an open source application framework for the Java platform and .NET Framework (Spring.NET).
Spring framework MVC was initially developed to address Jakarta Struts web framework and other web frameworks weaknesses. Being tightly linked to Spring Framework\textsuperscript{23}, it seemed natural to use this MVC implementation in the web service layer. In addition, Spring Framework MVC is well adapted to unit testing.

c) Business logic layer

The business logic layer is really the core implementation of SmileMS. Just as its web service counterpart, the business logic layer is widely based on Spring Framework. Transversal security aspects are based on another module of Spring toolkit: the Spring Security framework\textsuperscript{24}.

Spring framework

Two recurrent issues encountered while developing the identification platform are:

1. how to deal with different configurations, depending on the context of the application,

2. how to operate transversal operations.

Spring framework answers efficiently these two concerns by introducing implementations of: dependency injection and aspect-oriented programming.

Dependency injection

Formerly called "inversion of control", dependency injection allows giving an object a set of dependencies, rather than having this object instantiate its dependencies by itself (Figure 6.15). In the Spring Framework model, this operation is achieved externally, via a XML file\textsuperscript{25}.

Reversing the dependency process allows loose coupling\textsuperscript{26} of entities and separating implementations from their configuration within a complete system. Multiple configurations, depending on the context of

\textsuperscript{23}Technology used in the business layer of the platform

\textsuperscript{24}This framework has for long been known under the name of Acegi Security.

\textsuperscript{25}Alternatively, it can be done using annotation, directly in the source code.

\textsuperscript{26}If an object only knows about its dependencies by their interface (not their implementation or how they were instantiated) then the dependency can be swapped out with a different implementation without the depending object knowing the difference\cite{Wal05}.\]
dependency injection involves giving an object its dependencies as opposed to an object having to acquire those dependencies on its own (Figure retrieved from [Wal05]).

application, can then be created. Behaviors can also be modified without re-compiling the project.

In practice, we centrally use this functionality for scoring models, filters, exports, or security aspects.

Aspect-oriented programing (AOP)

A cross-cutting concern can be described as any functionality that affects multiple points of an application [Wal05]. Figure 6.16 presents this situation. Security, Transactions, and Other define three actions that affect the application in a transversal manner. AOP helps to modularize cross-cutting concerns.

In practice, we use AOP for transaction aspects, through hibernate\textsuperscript{27}, and for security concerns. This last aspect is further described in the next section.

Spring Security framework

Spring Security framework is the last piece of the Spring Framework that we employed. Spring Security provides a comprehensive security solution, handling authentication and authorization, at both the web request level

\textsuperscript{27}See paragraph "Hibernate" for more details.
Figure 6.16: Aspect Oriented Programming (AOP). Aspects (Security, Transactions, and Other in this figure) modularize crosscutting concerns, applying logic that spans multiple application objects (Figure retrieved from [Wal05]).

and at the method invocation level [Wal05]. In addition, Spring Security offers a complete Access Control List (ACL) system that enables to grant authority to users at the data level.

Spring Security is based on the two previously described concepts; dependency injection and aspect-oriented programming.

d) Data layer

The data layer is the last piece of implementation presented in this document. Two aspects particularly concern the implementation of this layer: readers of spectral data and database accesses.

The Java Proteomics Library (JPL)

JPL is an open source initiative consisting of java classes employed in proteomic investigations\(^{28}\). Though aimed at proteomics, JPL presents an important effort in the direction of MSMS spectra object representation. It can be considered a java and more evolved version of InSilicoSpectro\(^{29}\). JPL also deals with experimental data from LC-MSMS experiments such as experimental chromatograms and \(MS_n\) spectra.

\(^{28}\)See [http://sourceforge.net/projects/javaprotlib/](http://sourceforge.net/projects/javaprotlib/) for details about the JPL project
Besides being the skeleton of spectra data structure, JPL plays an important role in data extraction by providing readers for an array of peak list formats. We contributed to the development of this toolbox, by implementing data structures related to fragmentation spectra and LC-MSMS experiments, as well as developing new file readers.

JPL is an open-source initiative freely available for the scientific community.

**Proprietary data extraction**

We investigated two directions to handle proprietary data extraction. The first one was a in-house development of readers for Applied Biosystems and Thermo Scientific standards, while the second one was the use of open source initiatives.

On the one hand, XenoBol is a project developed by Ivan Topolski during his master’s degree internship in our lab. Its aim is to directly import raw data produced by a tandem mass spectrometer. XenoBol is composed of software tools that communicate with the proprietary APIs used by software controlling instruments. Tools are used to import spectra from the proprietary formats used in Analyst by Biodata and in Excalibur by Thermo. As these complex APIs rely on Microsoft’s OLE/COM object calling format, XenoBol was programed using the DispHelper C/C++ COM helper library. This makes it functional either with Microsoft Windows or with Winelib under Linux.\(^{30}\)

On the other hand, open-source converters are developed by universities and independent groups. Using these converters presents the advantage of outsourcing their development. Also, in most of the cases, these projects are conceived by experts in the field of standards and mass spectrometry instruments. This ensures a better coverage of format specificities, as well as a better reactivity to format change. *mzWiff* from Proteomecenter [Wik10b] and the *ProteoWizard* suite [Kes08] are two examples of open source converters.

After evaluating both options, we opted for an open source solution, since it offers better support and reactivity.

\(^{30}\)For more information on XenoBol, please refer to [Top09]
Chapter 6. Exploratory Software Engineering

Hibernate

Finally, SmileMS relies on Hibernate to handle databases interactions. Hibernate is an open source persistence framework that allows the standard implementation of persistent classes, irrespective of how these classes are rendered persistent. Like Spring framework, Hibernate is based on aspect-oriented programming.

Combined with Apache Lucene\textsuperscript{31}, Hibernate allows the execution of full text and advanced searches. These capabilities are centrally used by SmileMS, particularly retrieving of former analyses.

6.2.5 Development and build environment

a) Deployment

Newly produced code is not directly used in a production environment. This would threaten the stability of the platform and expose obvious bugs to users. Instead, the deployment is achieved in three steps.

1. new development is deployed on an development server; this server is used by the development team to run integration tests,

2. if new development is stable on the development server, it is deployed on the beta server; the beta server is a replica of the production one, in terms of data; this ensures proper testing on this instance, in order to mitigate risks when going to the production server,

3. the production server is the last stage of the deployment plan; this server is employed by real users.

b) Third-party environments

Third-party environments is an important part of SmileMS. As a platform aiming at working in tight collaboration with MS constructor software, SmileMS is confronted to various standards and pieces of software.

We decided to dedicate virtual machines\textsuperscript{32} to each of the major MS constructors. Newly produced code and installers are frequently tested on each

\textsuperscript{31}Apache Lucene is a high-performance, full-featured text search engine library written entirely in Java. For more details, see http://lucene.apache.org/java/docs/.

\textsuperscript{32}A virtual machine is a software implementation of a computer that executes programs like a physical machine.
virtual machine, in order to ensure proper functioning of SmileMS with regards to third-party software.

### 6.2.6 Testing

Last but not least, software verification and validation aims at showing that the application conforms to its specifications [Som06].

With iterative methodologies, the role of testing is even more profound. Tests are used to ensure proper functioning of the software, but also play a role in the requirement analysis, in documenting the software and during the design stage.

#### a) Regression testing

Regression tests (or unit tests) are conceived to test a precise functionality. They are called regression tests, because they guarantee that new development does not negatively impact on former features. Regression tests tend to appear increasingly early in the production of modern software. They are even produced to test a functionality identified in the requirement analysis before this functionality is even implemented.

Iterations between creating new test cases and implementing corresponding functions focus the development on clear objectives and facilitate a broad test coverage.

In practice, we began by coding functionalities and subsequently implement corresponding tests. However, due to external pressure, we rapidly neglected the testing phase, resulting in poor test coverage. To counteract this negative trend, we decided to implement tests in parallel to functionalities, in an iterative manner. This practice was greatly beneficial to recent development.

#### b) System testing

System testing is the testing of a software application, as a whole. These tests mainly involve human activity. People try every functionalities of SmileMS and report misbehaviors.

A more automated way of testing application consists in using test software such as Selenium\(^{33}\). This kind of application offers the possibility of

\(^{33}\)More information on Selenium can be found at the following address: [http://seleniumhq.org/](http://seleniumhq.org/)
testing web applications by automatically interacting with the interface, expecting a certain result as a consequence of these interactions.  

Finally, beta testers represent the last step of system testing. At this stage, the software is used by real users, who provide feedback on potential bugs and improvement. Beta testers are a very important source of information, whether they identify trends of use or detect bugs. As mentioned in the introductory part of this section, the research team at the HUG and the forensic center in Lausanne have been central to this testing and improvement process.

6.2.7 Project management software

In order to plan and coordinate software development, we used an open source solution called Trac\textsuperscript{35}. This web-based application is inspired by the wiki\textsuperscript{36} trend.

On top of Trac, we use another web-based application called AGILO\textsuperscript{37}, which is dedicated to Scrum software management.

6.3 Achievement and prospect

The combination of agile methodology for production purposes, and prototyping and the exploratory model for exploratory aspects proved to be efficient in developing SmileMS. Particularly, the changing environment generated by the constant exploration of data and our collaboration in the conception of a new analytical method at the HUG were properly handled by these combinations of software engineering methodologies.

From a technical point of view, at least two points would be differentially achieved if we had the opportunity to rethink our approach. First, though the modularization capabilities provided by Spring Framework were beneficial for the project, we would emphasize this dimension even more by using OSGI technology. OSGI is the dynamic module system for Java,

\textsuperscript{34}Support of Flex/Flax application was very poor in the past, but recent test frameworks tend to provide ways to test these kinds of applications.

\textsuperscript{35}http://trac.edgewall.org/

\textsuperscript{36}A wiki is a website that allows the easy creation and editing of any number of interlinked web pages via a web browser using a simplified markup language or a WYSIWYG text editor.

\textsuperscript{37}http://www.agile42.com/cms/pages/agilo/
that integrates aspects such as module management and updating\textsuperscript{38}. We had the opportunity to develop our fragmentation tool using this technology and thus saw how it is valuable to consider sub-projects autonomous modules.

The second aspect we would change in a second version of SmileMS is the conception of unit tests. We recently discovered that tests can be used to design, document, make evolve and test functionalities, rather than "simply" to achieve this last point. We would probably employ unit tests in this more globally way if we had the opportunity to rebuild SmileMS.

\begin{center}
\textbf{Chapter summary}
\end{center}

In this chapter, we introduced:

1. the context of our development, characterized by the important place of scientific exploration and a changing environment,

2. our interaction with the University Hospital of Geneva (HUG) in the conception of a new analytical method and how this interaction impacted on the evolution of SmileMS,

3. several classical software engineering methods and their principal characteristics,

4. a description of the main steps of SmileMS development,

5. a brief summary of how the approach we used successfully addressed our needs, as well as what we would change should we have the opportunity to rethink the approach.

\textsuperscript{38}See \url{http://www.osgi.org/Main/HomePage} for further information.
Chapter 7

SmileMs Identification Platform: concepts and usage (*)

“Software is like entropy: It is difficult to grasp, weighs nothing, and obeys the Second Law of Thermodynamics; i.e., it always increases.”

Norman Ralph Augustine, U.S. aerospace businessman

Contents

7.1 Platform overview ........................................ 144
7.2 Data processing .............................................. 144
  7.2.1 Data extraction and noise filtering ............. 145
  7.2.2 Identification process ............................... 147
  7.2.3 Aggregation and compound naming ............ 147
7.3 Interface usage ............................................... 148
  7.3.1 The desktop ........................................... 149
  7.3.2 Results validation ................................. 150
  7.3.3 Library management ............................... 150
  7.3.4 Administration and security ..................... 153
7.4 Bruker Daltonics: an integration case study ........ 154
This chapter presents the different parts of SmileMS. It also highlights the challenges we faced, as well as the actions we took to overcome them. Whereas chapter 6 presented the path we followed from an original idea to a production software, this chapter describes concepts and usage of this production software.

7.1 Platform overview

As presented in chapter 6, SmileMS identification platform is built of different modules. Five major aspects of these modules are presented in the following sections.

1. **Data processing**: the identification workflow from data reading to generation of aggregated results.

2. **User interface**: the different components of the user interface.

3. **Library management**: two possibilities to interact with library content.

4. **Administration and security**: platform administration and access privileges.

5. **Integration**: side-tools and strategies developed to interact with third-party software.

7.2 Data processing

Data processing is achieved in five steps (Figure 7.1):

1. spectra files are read,
2. treated to reduce spectral noise,
3. matched and scored,
4. results are filtered,
5. and finally aggregated.
7.2. DATA PROCESSING

Except for noise reduction, each of these steps can be manipulate by the user. Details of these stages are given in the following sections.

Figure 7.1: data processing is achieved in five steps: data are read, treated to reduce spectral noise, scored, filtered, and finally aggregated.

7.2.1 Data extraction and noise filtering

a) Data extraction

Data extraction is the first step of the data processing. With regards to challenges created by LC-MSMS instrumentations, this first step aims to address the immense variability of standards used by the different constructors and open-source initiatives.

Two directions were followed to cope with this question. The first consists in developing in-house converters: the XenoBol project. The second direction, which is the one that was finally elected, consists in taking advantage of existing open-source converters\(^1\).

Spectra file formats currently supported by the identification platform are presented in Table 7.1.

b) Noise filtering

Noise filtering consists in removing noise from imported spectra. This step is important to allow X-Rank to work in many different conditions. A basic

\(^1\)See section d) for further details on the two types of implementations.
### Open-source formats

<table>
<thead>
<tr>
<th>MzXML</th>
<th>WIFF (Applied biosystems)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MGF</td>
<td>RAW (Thermo scientific)</td>
</tr>
<tr>
<td>JPL</td>
<td>XML (Bruker Daltonics)</td>
</tr>
</tbody>
</table>

Table 7.1: major file formats currently supported by SmileMS.

noise reduction filter is implemented in the identification platform:

- look for the most intense peak,
- remove peaks in the neighborhood of this peak, within a pre-defined window,
- look for the second most intense peak,
- ...

A schematic representation of this process is presented in Figure 7.2.

![Figure 7.2: noise reduction by peaks filtering.](image)

The software architecture allows to plug more complex filters. For instance, de-isotopisation or spectra clustering could be undertaken. Further filtering approaches are developed in the thesis of Roman Mylonas [Myl10], section 3.2.
7.2.2 Identification process

The identification process consists of the two following steps of data processing: score generation and post-scoring filtering.

Multiple metrics are taken into account during these two steps. Particularly, chromatographic, MS1, and MS2 information are used by the scoring and the filtering module to produce an identification.

Chromatographic information

is employed to annotate a fragmentation spectrum with the retention time. During the post-scoring filtering, this value is compared to a reference value stored in the library. Users can chose to apply this filter or not.

MS1 information

is used in a similar way to chromatographic information. Similarly to the previous case, this value is compared to a reference value, during the post-scoring filtering. Again, users can chose to apply this filter or not.

MSn information

is the core metric used by X-Rank. It consists in fragmentation spectra obtained from chemical compounds present in the injected sample.

Additional metrics

such as MRM transitions, accurate mass, or internal standard information should be considered in a future version of the platform.

7.2.3 Aggregation and compound naming

The aggregation of matches presents identification information in a useful way. Instead of displaying a crude list of spectra matches, it constructs a consolidated list of potential compounds.

The aggregation consists of three steps (Figure 7.3). First, a list of potential hits is obtained from X-Rank. Second, the best match for each experimental spectrum is determined based on its score, retention time and precursor mass information. The compound identifier linked to the reference spectrum is also noted. Finally, experimental spectra that match reference spectra generated from a common compound are grouped, in order to form
a list of compounds. A score can then be attributed to the compound match itself.

Figure 7.3: aggregation consists on three steps: first, a list of potential hits is obtained from X-Rank, second, the best match for each experimental spectra is retained, and finally, experimental spectra that match reference spectra generated from a common compound are grouped in order to form a list of compounds.

Compound names are attributed through the PubChem index or via manual annotation, if no name can be retrieve in this index. This dual system presents the advantage of allowing to chose a specific name if the one proposed by PubChem is not suitable.

7.3 Interface usage

The user interface is divided into four main areas of functionalities:

- desktop,
- results validation,
- library management,
- administration and security.
7.3. INTERFACE USAGE

Figure 7.4: the SmileMS desktop (A) allows an user to make a new submissions or access an acquired job. By clicking on a job, a more detailed result view is shown in a new tab (B). Details about the identifications, such as the matching spectra (C), can be visualized. Precursor and retention time filters can be changed dynamically (D). In SmileMS the main views (desktop, library, and administration view) are accessible from within any window.

This topics are further developed in the following sections.

7.3.1 The desktop

The desktop is the first page a user sees after login. This view serves to either submit new jobs, or to access previous ones.

\[\text{The web access of this resource can be found at the following address: http://pubchem.ncbi.nlm.nih.gov/}\]
CHAPTER 7. SMILEMS IDENTIFICATION PLATFORM: CONCEPTS
AND USAGE (*)

Submission of new data is kept simple. The user can select an available profile and load spectra for submission. The profile contains a certain number of informations used for the identification (e.g. which library to use and which search parameters).

Once a job is completed, its results are directly exportable to PDF report. The job can also be accessed for more detailed investigations by clicking on it. This opens the results validation view, which is described in the next section.

7.3.2 Results validation

This view helps the user to verify the identification in more detail. Results are on the one hand visualized in a chromatogram, and on the other hand in a list of compounds (Figure 7.4, A).

The chromatogram shows different colors for each identified compound. This makes it easy to locate a compound in the chromatogram and estimate its concentration.

The second view, the list of compounds, is expendable to different levels of detail. It includes information such as X-Rank score, precursor and retention time differences and MS-levels. By selecting a match of interest, a spectrum viewer is opened. The two spectra (experimental and reference) are aligned in a mirror mode for an easy comparison (Figure 7.4, C).

Users with administrator privileges have access to further functionalities. They are allowed to change the filter values for the precursor masses and the retention times (Figure 7.4, D). They also have the possibility to add and edit the matching spectra (see Section 7.3.3).

Validated results can be exported as either a PDF report, or an Excel sheet. In a routine identification environment, the PDF report is often used to keep track of the results (typically the doctors in a hospital prefer results in a paper format).

7.3.3 Library management

Users interact with libraries via the result of an identification, to enrich libraries, or through a dedicated module for general administration.
7.3. INTERFACE USAGE

a) Library enrichment

While reviewing SmileMS results, users can assess the quality of spectra by visualizing them in a so-called spectrum viewer (Figure 7.5 A). If an experimental spectrum is found to be interesting during this review, users can choose to add this spectrum to an existing spectral library (Figure 7.5 B). Before validating this choice, they can also check, and if needed correct, the information related to the spectrum they are going to insert (Figure 7.5 C).

In the same interface, this last step of validating/editing information is also available for matching reference spectra.

Figure 7.5: the library enrichment workflow enables to A) visualize spectra of interest, B) select interesting ones as candidates for inserts in existing libraries, and C) check and correct annotations of these spectra.
b) Library management module

The second way to interact with libraries is the library management dedicated module. Via this module, users browse the content of the installed spectral libraries (Figure 7.6 A). They can also obtain the information related to spectra and display them on a spectrum viewer (Figure 7.6 B and C). Finally, users can export a pdf report of a library (Figure 7.6 D) and import new ones (Figure 7.6 E).

Figure 7.6: the library management dedicated module allows to A) browse existing libraries, B) obtain spectra information, C) display spectra, D) export a pdf report of a library, and E) import new libraries.
7.3. INTERFACE USAGE

7.3.4 Administration and security

_SmileMS_ provides a number of administrational tools, allowing the user to manage his work environment.

**a) Server administration**

The server administration gives access to three different tasks: the profile, the user and the license management. The profiles are, as already mentioned before (Section 7.3.1), the entity which lets the user chose the search parameters. Each user has the right to create or edit his own profiles. The profiles compromise the following informations:

- libraries to be searched against,
- a minimum required _X-Rank_ score for the results,
- a minimum number of required matches between two spectra (this helps to speed up the identification),
- default values for the precursor mass and retention time filters.

The second management task, the user management, allows to change user information. This includes names, login, password and e-mail address. Users with administrator rights have also the possibility to modify other users, and to create new ones.

**b) Role- and User-based security**

In _SmileMS_ each user has its own desktop. The data specific to a user include his runs, the profiles, and the user information. Currently there are two different types of users for _SmileMS_. There is a normal user and a superuser with administrator rights. Some of the functionalities are only available for the administrator users (e.g. user creation, library editing). In addition, the administrators have the right to access the jobs from all other users.

The separation between normal and super users is very convenient in the context of a routine laboratory. While the normal users make submissions and access their results, their supervisor has also the possibility to access the results from his colleagues. In case of uncertainty, the superuser can directly open the doubtful identification and verify the results himself.
7.4 Bruker Daltonics: an integration case study

In addition to the standalone version, *SmileMS* exists as an integrated software. Such integration was conducted with MS manufacturer Bruker Daltonics. In this version of *SmileMS*, the main functionalities of the platform are accessed through its web services, bypassing the user interface for actions such as the submission.

This integration work facilitates the identification of MS acquisitions with *SmileMS*. Whereas a normal workflow would imply an extraction and a conversion of raw data, followed by a submission to the identification platform, the integrated workflow seamlessly executes these actions. In addition, it allows *SmileMS* to access low level information from the proprietary software.

Figure 7.7 illustrates the acquisition and identification workflow of an analysis conducted on a Bruker Daltonics instrument.

![Figure 7.7](image)

**Figure 7.7:** the Bruker Daltonics integration workflow facilitates interactions between Bruker instruments and *SmileMS*. Data acquired on this brand of instrument are automatically submitted to the identification platform. Once the identification is complete, the analyst is contacted by email.
7.4. **BRUKER DALTONICS: AN INTEGRATION CASE STUDY**

First, data are acquired on the MS instrument (A). This instrument is controlled via standard Bruker Daltonics software suite (B) which automatically communicates with the *Open Access* Bruker client (C). Finally, this client interacts with *SmileMS* by sending experimental data and other information, retrieving the generated result, and sending an email to the analyst, indicating the completion of his or her analysis.

**Summary**

In this chapter, we presented:

1. the data processing workflow of *SmileMS*, including spectra reading, filtering, scoring and aggregation of the results.

2. *SmileMS* user interface, consisting of a desktop, results validation, library management, administration and security.

3. the integration with a MS instrument software from Bruker Daltonics.
Chapter 8

Case Study: the University Hospital of Geneva (HUG)

"Medical scientists are nice people, but you should not let them treat you."

August Bier, German surgeon

Contents

8.1 LC-MSMS as an alternative to the REMEDI .......................... 158
8.2 The HUG environment ...................................................... 160
  8.2.1 Analytical methodology ............................................. 160
  8.2.2 Algorithm training ..................................................... 161
  8.2.3 Multiple instruments handling ..................................... 161
  8.2.4 Poor spectra handling ............................................... 163
  8.2.5 MRM versus GUS ....................................................... 164
8.3 Comparison with existing software ........................................ 166
  8.3.1 Comparison with Analyst ............................................. 166
  8.3.2 Comparison with Cliquid ............................................ 167
8.4 Libraries ........................................................................ 169
  8.4.1 Building of library ..................................................... 169
  8.4.2 Library enrichment .................................................... 171
The University Hospital of Geneva (HUG) is one of the biggest hospitals in Switzerland. It is an environment for both research and patient care. The HUG employs 8'401 people, cumulates 48314 admissions per year, for a total of 733’176 days of hospitalization.\(^1\)

The laboratory of clinical toxicology is responsible for all toxicological analyses ordered by the physicians and the emergency service. It has to quickly deliver reliable and accurate results. Usually, such analyses take the form of broad blood or urine drug screenings.

Due to REMEDi discontinuation and limitations, the laboratory of toxicology has to find a suitable alternative to this analytical instrument. The replacement methodology is to be routinely used for emergency samples analysis from March 2010.

This chapter presents the use of *SmileMS* at the HUG. We expose the context of this collaboration, initially created by the discontinuation of the REMEDi instrument. We also introduce the analysis workflow developed at the HUG and how *X-Rank* was trained to suit this analytical environment, composed of multiple MS instruments. We finally discuss the comparison of *SmileMS* with previously employed software, as well as the question of library building in the toxicology laboratory of the HUG.

## 8.1 LC-MSMS as an alternative to the REMEDi

### Current analytical methods

An array of analytical techniques are employed at the toxicological laboratory of the HUG. The main ones are REMEDi, immunoassay tests, and GC-MS. These techniques are further described in section 1.3, each of them highlighting pros and cons. Consequently, they tend to be used in a complementary manner, creating a complexity of use and an decreased speed of analysis. As a result, the REMEDi was used as a first line screening test, leading to further investigations, in cases of specific suspicions. Today, the REMEDi has to be replaced by another polyvalent instrument, due to its discontinuation.

\(^1\)Statistics were retrieved from the website of the HUG at the following address: [http://www.hug-ge.ch/hug_enbref/chiffres-cles_2009.html](http://www.hug-ge.ch/hug_enbref/chiffres-cles_2009.html)
LC-MSMS

LC-MSMS was chosen as the alternative to REMEDI. It offers a broader range of detection for drugs in various matrices and allows for increased speed of analysis. By enabling screening for a broader range of molecules, it also diminishes the need for other analytical tests. Moreover, considerations in terms of cost, robustness, generic procedure, and ease of use were also satisfied by LC-MSMS.

Comparison of REMEDI and QTRAP instruments

Using SmileMS as the identification software, a comparison study between the REMEDI and the triple quadrupole mass spectrometer showed convincing results in favor of using LC-MSMS\(^2\). 146 real serum samples were tested. In total, 77 chemical compounds were identified: 49.2\% of them were identified by both REMEDI and the LC-MSMS technique, 10.9\% only by the REMEDI and 39.9\% only by LC-MSMS.

Substances found only by the REMEDI highlight the need to have complementary approaches, in order to detect the whole range of searched substances.

To further look into the benzodiazepine class, 91 sample positively tested for this class of molecules using immunoassays, were analyzed. LC-MSMS could confirm the presence of benzodiazepines in 80 cases, where REMEDI positively identified these chemical compounds in only 7 samples.

These results are summarized in a table presented in the introductory part of this thesis (Figure 1.8).

Interestingly, retention time information was not needed in most of the cases, with the LC-MSMS method. One exception to that claim is the case of the very close chemical structures of oxazepam and the demoxepam\(^3\). These two molecules, as well as their respective spectra, are presented in Figure 3.8, chapter 3.

---

\(^2\)Comparison performed by Veronique Viette and her colleagues.

\(^3\)Further details can be found on the poster presented by Veronique Viette for the inauguration of the SCAHT [Vie09].
8.2 The HUG environment

8.2.1 Analytical methodology\(^4\)

After patients’ admission at the hospital and samples collection, the newly developed analytical method involves four stages that finally lead to a list of detected chemical compounds (Figure 8.1). These four steps are:

1. a solid phase extraction,
2. a liquid chromatography separation,
3. a tandem mass spectrometry analysis,
4. an automated data analysis and library search\(^5\).

![Figure 8.1: schematic representation of the analysis workflow deployed at the HUG toxicological laboratory.](image)

For the third stage of this process (i.e. the tandem MS analysis), two classes of MS instruments are used. The first one is the hybrid triple quadrupole linear ion trap (a 3200 QTRAP\(^6\) instrument). This machine enables the detection and identification via information-dependent acquisition (IDA) of a

---

\(^4\)Material presented in this section has been mainly produced by Veronique Viette and her colleagues. Further details will be published in a submitted article entitled “A multi-target screening analysis using fast liquid chromatography-hybrid mass spectrometry”.

\(^5\)This step is conducted using SmileMS

\(^6\)From Applied Biosystems.
8.2. THE HUG ENVIRONMENT

drug, in a single MS run. The second class of instruments is the 3D ion trap
(an Esquire HCT and an AmaZon\textsuperscript{7}). These two 3D ion trap mass spectrometers produce rather similar spectra when properly parametrized\textsuperscript{8}.

The automated data analysis and library search were achieved using SmileMS. The platform was used with the two mass spectrometers and by about ten users. Among them two were administrator of the platform, whereas the others were routine technicians.

8.2.2 Algorithm training

\textit{X-Rank} parameters and model were trained for QTRAP data acquired at the HUG, in order to maximize performance. Since spectra from the Weinmann library were generated using the same instrument, this library was used in a leave-one-out manner to achieve this training. The statistical distributions we obtained are presented in section 3.1 of this document, Figure 3.1 and Figure 3.2.

Matching tolerance $\tau$ (\textit{i.e.} the mass of a peak differs from that of the matching peak by less than a given tolerance $\tau$) was empirically determined to be optimal at a value of 0.6 Daltons. It is interesting to notice that this value is higher than the precision of the instrument, estimated around 0.3 Dalton.

Finally, for each evaluated pair of spectra, the number of peaks retained for the first and the second spectra were respectively the 7 and the 30 most intense peaks\textsuperscript{9}. Since the score is symmetrically computed, the initial first spectrum becomes the second one during the second computation. The number of considered peaks strongly depends on the targeted application. For example, proteomics application would ask for much more peaks since spectra tend to be richer in this field.

8.2.3 Multiple instruments handling

As previously mentioned, two classes of instruments were used for the method development and the routine analysis work. The advantage of using multiple instruments resides in the complementarity they offer. Taking the example of the routine work of the toxicology laboratory at the HUG,

\textsuperscript{7}From Bruker Daltonics.
\textsuperscript{8}Note that these parameters are not linked to \textit{X-Rank}. They are acquisitions parameters used by the MS instrument.
\textsuperscript{9}This is calculated after noise filtering.
samples are analyzed on the AmaZon instrument, with dubious identifications being double-checked on the QTRAP.

From a technical point of view, using multiple instruments in a laboratory implied at least two constraints:

1. the identification platform has to be accessible by acquisition computer of each instruments,

2. specificities of each instruments, such as the use of different file standards, have to be properly handled.

To answer these needs, SmileMS was installed as a web server, at the hospital. Acquisitions and subsequent identifications carried out with the AmaZon instrument were handled by the integration of SmileMS server and Data Analysis / Compass OpenAccess from Bruker Daltonics\textsuperscript{10}. In the case of the QTRAP instrument, submissions were managed using a software client that programmatically connects to SmileMS. Though not as efficient as a complete integration, the use of this client avoid users to connect to SmileMS and to manually submit their experimental data.

Parallel analyses using SmileMS

Parallel analyses, using SmileMS as identification tool, were conducted with the QTRAP instrument, using the Weinmann/HUG library on one side, and with the AmaZone, using a Esquire HCT library\textsuperscript{11} on the other side. Table 8.1 presents identifications of a blood sample on the QTRAP (left), as well as the identification of the same sample on the AmaZon (right).

Most of the searched compounds were identified on both systems (Diazepam, Citalopram, Nordazepam, D5-Amphetamine, D3-Clomipramine, and Oxazepam). However, three identifications (Nicotine, Temazepam, and Lidocaine) could not be found with the AmaZon system. In addition, Tamoxifen is only found with the AmaZon system. Table 8.2 describes our interpretations of these differentially identified substances.

On the whole, this example demonstrates the ability to use the platform with different machines, showing consistent results between the two systems. It also highlights needs for possible improvements of the scoring, particularly in the case of spectra with a reduced number of peaks.

\textsuperscript{10}Details on the integration work are provided in section 7.4

\textsuperscript{11}Refer to section 8.4 for more details.
8.2. **THE HUG ENVIRONMENT**

<table>
<thead>
<tr>
<th>QTRAP identifications</th>
<th>AmaZon identifications</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Compound</strong></td>
<td><strong>X-Rank score</strong></td>
</tr>
<tr>
<td>Oxazepam</td>
<td>0.9</td>
</tr>
<tr>
<td>Nordazepam</td>
<td>0.9</td>
</tr>
<tr>
<td>Nicotine</td>
<td>0.9</td>
</tr>
<tr>
<td>Diazepam</td>
<td>0.89</td>
</tr>
<tr>
<td>D5-Amphetamine</td>
<td>0.88</td>
</tr>
<tr>
<td>D3-Clomipramine</td>
<td>0.87</td>
</tr>
<tr>
<td>Citalopram</td>
<td>0.83</td>
</tr>
<tr>
<td>Temazepam</td>
<td>0.64</td>
</tr>
<tr>
<td>Lidocaine</td>
<td>0.54</td>
</tr>
</tbody>
</table>

Table 8.1: *SmileMS* identifications of a blood sample using (left) a QTRAP 3200 from Applied Biosystems and a library from the same instrument type and (right) an Esquire HCT and a library from the same instrument type. Except for the Nicotine and Temazepam which were not present in the Esquire library, the Lidocaine, which is poorly recognized by the identification algorithm, and the presence of Tamoxifen, which is likely to be an artifact or a the effect of a contaminant, this example demonstrates the consistency of identification as well as the possible use of the platform with different machines.

8.2.4 **Poor spectra handling**

Another challenge at the HUG consisted in the handling of spectra containing very few peaks. Taking the example of the Weinmann library enriched by the team from the hospital, the total number of spectra is 1101, the minimum number of peak is 1, the maximum 612, and the average 66.63. Globally, 58 spectra (or 5.3% of the spectra) of this library contain less than 7 peaks\(^\text{12}\). These spectra are considered poor spectra. The example of the lidocaine presented in section 8.2.3 is an extreme example since the fragmentation spectrum contains only one peak\(^\text{13}\).

Poor spectra are often encountered in negative ionization mode but can also be noticed with certain compounds, in positive mode. Scores of matches involving these spectra are too low compared to richer spectra matches. This is due to the lack of a proper score normalization that takes spectral

\(^{12}\)In the hospital conditions, the algorithm considers the 7 most intense peaks of spectra.

\(^{13}\)Spectra with only one peak represent 0.54 % of spectra in the Weinmann/HUG library.
CHAPTER 8. CASE STUDY: THE UNIVERSITY HOSPITAL OF GENEVA (HUG)

<table>
<thead>
<tr>
<th>Substance</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nicotin and Temazepam</td>
<td>The absence of nicotin and temazepam is explained by the fact that none of these substances was present in the Esquire HCT spectral library. Since those substances are present in the Weinmann/HUG library, cross-instrument capabilities of the identification algorithm can potentially be used to enrich the Esquire HCT library. This library enrichment is detailed in section 8.4.</td>
</tr>
<tr>
<td>Lidocain</td>
<td>For this chemical compound, the library spectrum only contains one peak and is consequently not detected by the algorithm with a sufficient score. This example is characteristic of poor spectra identification and is further detailed in section 8.2.4 of this chapter.</td>
</tr>
<tr>
<td>Tamoxifen</td>
<td>The tamoxifen in the AmaZon identification is probably an artifact, since it is constantly found on run identified using this instrument. This incorrect identification can be suppressed by applying a 0.7 Dalton filter on the precursor mass.</td>
</tr>
</tbody>
</table>

Table 8.2: interpretation of chemical compounds differentially identified between the AmaZon and the QTRAP identifications presented in Table 8.1.

richness into account. A first approach to address this issue consists in the normalization of matching scores with the maximum score that a spectrum can obtain (i.e. the score that would be obtained for a perfect match, where each experimental peak matches a reference peak of a corresponding intensity rank). This approach is further detailed in the Phd thesis of Roman Mylonas [Myl10], section 3.3.

8.2.5 MRM versus GUS

An additional preoccupation was the choice of the screening strategy. Several MS instruments, such as QTRAPs, can operate in multiple ways. Among
8.2. THE HUG ENVIRONMENT

the most common possibilities, the ionization mode can vary (positive or negative) and the screening technique as well. Taking the example of the HUG, two options were assessed: GUS\textsuperscript{14} and MRM\textsuperscript{15} screening. In the context of the University Hospital of Geneva, GUS is employed on the AmaZon instrument, whereas MRM is used on the QTRAP.

Choosing between these two options impacts not only the experimental design, but also produced data and consequently proposed identifications. MRM selects compounds to be fragmented, generating a reconstituted chromatogram that looks cleaner than GUS derived ones. Figure 8.2 illustrates this discrepancy of complexity.

![Figure 8.2: Example of a 10 substances mixture. MRM chromatograms (down) tend to be rather unambiguous, while GUS ones (up) can be much more complex.]

On the other hand, GUS screening presents the advantage of allow-
CHAPTER 8. CASE STUDY: THE UNIVERSITY HOSPITAL OF GENEVA

From an identification point of view, GUS screening does not imply any performance modification, such as an increase of the false positive rate. This is easily explainable, since the algorithm is only concerned with spectra themselves, whether they are many or just a few. However, complexity level of a chromatogram is not taken into account at the algorithm level. Since chromatogram intensities are not directly exploited\textsuperscript{16}, there is a risk of introducing misleading identification. For example, low intensity spectra, showing a very clear fragmentation pattern produce high scores that place the match at a high confidence level in the generated results. In addition, non-relevant but potentially highly concentrated substances\textsuperscript{17} can hinder the reading of results by hiding smaller chromatographic peaks, for instance. These two examples are valid for both GUS and MRM, but tend to be more often problematic in the first technique, due to the more important number of signals.

Finally, the large amount of data generated by GUS experiments can also burden the identification process from a computing performance point of view. In that regard, SmileMS shows good scalability performance, keeping within one minute the identification of samples containing several thousands of spectra against libraries of the same size. This estimate accounts for non-stringent identifications. Performance dramatically increases when restraining the search space by applying filters.

8.3 Comparison with existing software

After having developed and adapted SmileMS to suit a medical environment, a next logical step was to benchmark its pertinence, compared to the software used in the toxicological laboratory of the HUG.

8.3.1 Comparison with Analyst

From an algorithmic point of view, a strict comparison with Analyst software is difficult because of the manual library search characteristic of this

\textsuperscript{16}The word “directly” is important, since the number of positive matches highlighted by the algorithm reflect the number of acquisition of a given substance and thus somehow the intensity of this substance on the chromatogram.

\textsuperscript{17}This case has often been reported for caffeine and nicotine.
application. In that respect, *SmileMS* offers a real advantage, by providing the analysis in an automated way.

The scope of the two applications also is different. Whereas *Analyst* is used for the identification and the acquisition, *SmileMS* focuses on the identification and library-related aspects.

In terms of identification efficiency comparison, *Cliquid* is preferred, thanks to its automated library search. A comparison example is provided in the next section.

### 8.3.2 Comparison with *Cliquid*

*Cliquid* software application acts as a layer above *Analyst* and allows running automated library search. Though *Cliquid* does not propose compound aggregation, it is still more adapted to comparative analysis than *Analyst*. Being based on the same algorithm, we here assume that *Cliquid* comparisons give a good estimate of *Analyst* performance.

In the following example, an MRM-acquired test sample was analyzed with both *SmileMS* and *Cliquid*. *SmileMS* was set not to use precursor or retention time filter. *Cliquid* was set to use a 0.3 precursor mass window. In addition, MRM data provides *Cliquid* with MRM transitions. These transitions are used by *Cliquid* in the identification of substances.

Table 8.3 and Table 8.4 show respective results for *SmileMS* and *Cliquid*. The first table shows *X-Rank* scores, while the second proposes three different scores; a fit\(^1\), a reverse fit\(^2\), and a purity\(^3\). In the case of the *Cliquid* analysis, each line corresponds to a searched MRM transition. Clenbuterol corresponds to an empty line because no matching spectrum has been found for the spectrum acquired with the transition corresponding to this substance. In both *SmileMS* and *Cliquid*, only the best candidate for each substance is shown.

While the great majority of chemical compounds (Diazepam, Codeine, Caffeine, Amiodarone, Morphine, Doxepin, Haloperidol, and Amphetamine) are found on both systems, three (Hydromorphone, 7-Aminoclonazepam, 18-Dot-product based score obtained when the experimental spectra is compared to the reference one.
19-Dot-product based score obtained when the reference spectra is compared to the experimental one.
20- *Analyst* being a close commercial software application, it is difficult to find exact definition of metrics. However, we estimate that the purity is a measure of distance between an identification and the following.
CHAPTER 8. CASE STUDY: THE UNIVERSITY HOSPITAL OF GENEVA

Table 8.3: results of SmileMS for a MRM acquired test sample. SmileMS was set not to use precursor or retention time filter. Only the best candidate for each substance is shown.

<table>
<thead>
<tr>
<th>SmileMS identifications</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diazepam</td>
<td>0.9</td>
</tr>
<tr>
<td>Codeine</td>
<td>0.85</td>
</tr>
<tr>
<td>Caffeine</td>
<td>0.81</td>
</tr>
<tr>
<td>Amiodarone</td>
<td>0.75</td>
</tr>
<tr>
<td>Morphine</td>
<td>0.72</td>
</tr>
<tr>
<td>Doxepin</td>
<td>0.69</td>
</tr>
<tr>
<td>Haloperidol</td>
<td>0.63</td>
</tr>
<tr>
<td>Amphetamine</td>
<td>0.57</td>
</tr>
</tbody>
</table>

Table 8.4: results of Cliquid for a MRM acquired test sample. Cliquid was set up to use a 0.3 precursor mass window. In addition, MRM data provides Cliquid with MRM transitions. These transitions are used by Cliquid in the identification of substances. Each line corresponds to a searched MRM transition. Only the best candidate for each substance is shown.

<table>
<thead>
<tr>
<th>Cliquid identifications</th>
<th>Fit (%)</th>
<th>Rev. Fit (%)</th>
<th>Purity (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caffeine</td>
<td>65.88</td>
<td>100.00</td>
<td>65.9</td>
</tr>
<tr>
<td>Morphine</td>
<td>92.46</td>
<td>97.41</td>
<td>90.1</td>
</tr>
<tr>
<td>Hydromorphone</td>
<td>85.69</td>
<td>74.12</td>
<td>67.8</td>
</tr>
<tr>
<td>7-Aminoclonazepam</td>
<td>42.86</td>
<td>76.22</td>
<td>32.7</td>
</tr>
<tr>
<td>Codeine</td>
<td>93.19</td>
<td>88.32</td>
<td>82.3</td>
</tr>
<tr>
<td>Amphetamine</td>
<td>93.98</td>
<td>100.00</td>
<td>94.0</td>
</tr>
<tr>
<td>Hydrocodone</td>
<td>77.18</td>
<td>68.82</td>
<td>57.5</td>
</tr>
<tr>
<td>Diazepam</td>
<td>95.70</td>
<td>94.59</td>
<td>91.2</td>
</tr>
<tr>
<td>Clenbuterol</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Haloperidol</td>
<td>95.49</td>
<td>100.00</td>
<td>95.5</td>
</tr>
<tr>
<td>Doxepin</td>
<td>83.85</td>
<td>72.12</td>
<td>61.0</td>
</tr>
<tr>
<td>Amiodarone</td>
<td>100.00</td>
<td>100.00</td>
<td>100.0</td>
</tr>
</tbody>
</table>

Table 8.3: results of SmileMS for a MRM acquired test sample. SmileMS was set not to use precursor or retention time filter. Only the best candidate for each substance is shown.

Table 8.4: results of Cliquid for a MRM acquired test sample. Cliquid was set up to use a 0.3 precursor mass window. In addition, MRM data provides Cliquid with MRM transitions. These transitions are used by Cliquid in the identification of substances. Each line corresponds to a searched MRM transition. Only the best candidate for each substance is shown.

and Hydrocodone) are found only by Cliquid, and one (Clenbuterol) if found neither by Cliquid, nor by SmileMS.

The three matches identified as positive identifications by Cliquid, but not found by SmileMS, are interesting cases. Figure 8.3 show the three
matches, as proposed in the *Cliquid* result export. A common behavior can be identified in these three cases; only one high-mass and high-intensity peak (in red on Figure 8.3) matches between the experimental and the library spectra\(^{21}\). These poor matches cast doubt on the identification of these three substances, based on \(MS_2\) matches.

This behavior highlights a weakness of dot-product derived algorithms, which tend to give an exaggerated importance on highly intense peaks. Moreover, this family of algorithms usually includes empirical rules, such as the over-weighting of high mass fragments. Such scoring enhancement can make sense, but must be well mastered to avoid deterioration of performance leading to cases such as those presented in Figure 8.3.

Another interesting aspect is the fact that *SmileMS* did not use precursor mass filters, whereas *Cliquid* did. Despite this disadvantage, *SmileMS*’ result do not display lower performance. However, by not using precursor filters, *SmileMS* could detect substances with adducts or metabolites, whereas *Cliquid* would be useless for these chemical compounds.

### 8.4 Libraries

#### 8.4.1 Building of library

As presented in Figure 2.1 from section 2.1.1, spectra coming from 3D traps, such as the AmaZon machine, present different fragmentation patterns than spectra acquired on QqQ-linear ion traps, such as the QTRAP instrument. In order to obtain optimal results using the AmaZon, it was decided to create a new toxicology-oriented spectral library, acquired on AmaZon and Esquire HCT instruments.

To build this library, chemical substance were injected as a pure sample in methanol. As a result, he best spectra for each of these compound were selected using *Data Analysis* software. *.library* files were then exported from *Data Analysis* and imported to *SmileMS*.

In order to allow a correct aggregation of results, using chemical annotations, compound names were added to the imported spectra. Retention times, chemical formulae, and theoretical precursor masses were also corrected when needed.

\(^{21}\)In the case of Hydroconone, two other minor peaks seem to match, at around 199 and 243 \(m/z\)
Figure 8.3: spectral matches proposed by Cliquid. Top-left: Hydromorphone, top-right: 7-aminoclonazepam, and bottom: Hydroconone.

The current version of this spectral library contains 146 chemical com-
8.4. LIBRARIES

pounds relevant for a clinical toxicology laboratory\textsuperscript{22}. From a spectrum point of view, 416 $MS_2$ and $MS_3$ spectra were selected and stored in the library.

8.4.2 Library enrichment

Library enrichment capabilities are used for two types of goals: either to build new libraries from scratch, or to improve the content of existing ones. At the HUG, the main use of this functionality concerned the second case.

To illustrate library enrichment, we refer to the aforementioned example of a blood sample analysis (see section 8.2.3). Let us recall the missing temazepam in the AmaZon identification. This situation arose from the absence of reference spectrum for this chemical compound in the Esquire HCT library, suggesting the need for including this entry. Detecting a proper AmaZon acquired temazepam spectrum was done while analyzing the AmaZon experimental data against the QTRAP-library\textsuperscript{23}. Temazepam could be identified with a score of 0.49 (Figure 8.4). The corresponding experimental AmaZon spectrum was then added to the Esquire CHT library.

A second blood sample where the QTRAP identification showed positive propranolol and temazepam (with scores of 0.84 and 0.83 respectively) was analyzed using the AmaZon and the modified Esquire HCT library. Both compounds were successfully detected with scores of 0.91, demonstrating an effective library modification.

This enrichment of library shows how cross-platform performance of SmileMS can be used in a context of library building.

\textsuperscript{22}Marc Fathi from the University Hospital of Geneva estimated that it covers more than 90\% of his needs.

\textsuperscript{23}A cross-instrument identification.
Figure 8.4: the missing compound temazepam was added to the Esquire library using a cross-instrument identification. The experimental Temazepam spectrum acquired on the Esquire positively matched the Temazepam spectrum from the QTRAP library (score of 0.49). After insertion of the experimental spectrum into the Esquire library, Temazepam was identified with high scores (0.91).

Chapter summary

In this chapter, we introduced:

1. the environment, provided by the University Hospital of Geneva, in which SmileMS is used as a routine emergency department software,

2. the training of X-Rank in this environment, as well as its performance alone and compared to Cliquid software,

3. the importance of using multiple instruments, in terms of complementarity and the software handling of this multiplicity,

4. the distinction between MRM and GUS analyses, in an identification context,

5. the building and enrichment of libraries, as well as how this can improve the handling of multiple MS instruments.
Chapter 9

Discussion and conclusion

"The worthwhile problems are the ones you can really solve or help solve, the ones you can really contribute something to. No problem is too small or too trivial if we can really do something about it."

Richard Phillips Feynman

Contents

9.1 Data exploration platform ........................................ 174
  9.1.1 Scoring model development .............................. 174
  9.1.2 Library standardization and sharing ..................... 176
  9.1.3 In silico chemical compound fragmentation .......... 177

9.2 Collaboration with the HUG .................................. 177
  9.2.1 MRM versus GUS .................................. 177
  9.2.2 Cross-platform identification ........................... 178
  9.2.3 Integration with MS manufacturer software ........... 179

9.3 Beyond clinical toxicology ................................... 179
  9.3.1 Current limitations .................................. 180
  9.3.2 LC-MS pre-filtering ................................. 180
This chapter presents conclusions and perspectives regarding SmileMS and its place within three different contexts; the exploration of data, our collaboration with the HUG, and finally applications other than clinical toxicology.

9.1 Data exploration platform

SmileMs was conceived as an exploration environment for research and development in the field of small molecule identification. Thanks to a modularized design, it is easy to develop new modules that use available objects and mechanisms. Throughout the development of the platform, we used these fundamental mechanisms to investigate and then integrate new functionalities.

9.1.1 Scoring model development

The development of X-Rank is a first example of such exploration. X-Rank was initially developed to achieve cross-platform identification in the field of LC-MSMS. Though it shows good performance when used in a standard way, it is still not discriminant enough for heterogeneous identification (i.e. comparisons achieved using experimental data acquired under conditions different from reference ones), in a medical environment.

Identifying heterogeneous data is interesting in the sense that this question approaches theories of signal processing. If we put the small molecule identification aside, this subject can be summarized as the distinction of pertinent signal in a noisy environment. In this approach, one tries to focus on parts of the signal that are conserved, despite a general dissimilarity. This kind of strategy is employed in numerous applications, such as automated music recognition. In this kind of automated process, an identification algorithm tries to identify musical signatures (conserved signal) among environmental noise. To come back to the field of small molecules, one objective is to get rid of disturbing signals and enhance interesting ones. In this context, the use of filters and spectra clustering could be tested.

In spite of the non-applicability to medical analysis, the X-Rank cross-platform performance is interesting in other contexts. In the context of spectral libraries, X-Rank can be used to build libraries from different instruments. X-Rank can also handle identifications implying spectra acquired on the same family of instruments, but under different conditions.
This first overview of X-Rank performance highlights a good overall scoring model that still suffers from weaknesses and thus needs further investigations. Enhancements presented in section 3.3 are a subset of the approaches we tested\textsuperscript{1}. Although these enhancements improved performance for well-defined areas, we learned that combining metrics is not a trivial process.

In that regard, WEKA is resourceful for exploring and probing new combinations of scores through the use of qualifiers. The question of combining scores goes far beyond the identification of small molecules. This issue is found in almost every application relying on multiple tools to analyze data. Though the step of combining metrics has not yet been achieved in this case, the peptide analysis workflow SwissPIT [Qua09] is a good example of such use of more than one scoring engine. In this case, the principle is to chain several peptide identification platform and to concatenate their results. A improvement of this platform would consist in building metascores that would take advantage of all score strength, without degrading their individual performance. In our presentation of scoring enhancements, we approach this metascore building by the use of classifiers such as neural network or supported vector machine.

Another important advantage of using qualifiers is the answer returned to the user. While scoring functions return a score which needs to be interpreted by the user, classifiers qualify a spectral match as true or false. Despite the potential danger of such an answer, it better suits users expectations in a medical environment. As a result, a perspective for X-Rank is to get closer to binary answers, keeping its result quality.

Besides the MSMS scoring itself, the pertinence of identification can be increased by taking other information (such as the fragmentation patterns, known metabolites, the precursor mass, or the retention time) into account. For example, the two very close chemical compounds presented in section 3 appear to differ by more than one minute, in terms of retention time, with the method used at the HUG. This example illustrates a case that can easily be addressed by a retention time filter.

\textsuperscript{1}Others of these approaches are detailed in the thesis of Roman Mylonas [Myl10], chapter 3.
9.1.2 Library standardization and sharing

Another area of exploration is the question of spectral library building and sharing. Even the very definition of spectral libraries seems not trivial. From $MS_1$ to $MS_n$ libraries, from those only made of spectra to highly annotated ones, libraries are unique and usually built to fulfill a dedicated range of applications.

In our approach, we tried to identify the different types of information and a way to organize them. The goal of this approach is the standardization of libraries, enabling at least two advances: their use throughout multiple environments and their sharing. The standardization of libraries is mainly based on the proper spectra assignment of chemical compound identifiers. Our hybrid system, between PubChem identifier and an internal one, answers current needs, but is difficult to maintain, due to the manual annotation users have to achieve.

Today, no software proposes library sharing functionalities, probably due to the technical issues of format conversion, chemical compound annotation and update mechanisms. Whereas the format and the annotation issues can be tackled by the standardization of libraries, update mechanisms exist in SmileMS and could be further developed to enable library sharing. Library sharing would be beneficial when pure substances are difficult to obtain and to avoid the cost of building new libraries.

Beyond these two benefits, the community aspect of library sharing could be interesting in rising the quality of libraries. In a different domain, the free encyclopedia Wikipedia is an impressive example of such a mechanism. Though the question of Wikipedia’s reliability is controversial, many people agree to say that the peer-reviewing process is beneficial to improve the quality of articles. Tom Cross, in an article about this kind of open initiatives [Cro06], wrote that “bad information is less likely to survive a collaborative editing process”, suggesting that a selection of high quality articles is naturally achieved. This modern version of Darwin’s Law of natural selection could perhaps be applicable to spectral libraries in a context of sharing. If it is the case, as mentioned by Tom Cross in the same article, the peer editing of information must be well presented to allow an easy evaluation of data quality. This is important to ensure the trust of people with regards to the information proposed by the platform.
9.1.3 *In silico* chemical compound fragmentation

Our last area of exploration consisted in the study of chemical fragmentation. In chapter 5, we introduced the central notion of *fragmentation graph*. This graph considers fragments of a molecule as a system rather than as disconnected parts. Beyond the role of this graph in assessing dubious identifications, it seems adapted to modeling of fragmentation process, as fragmentation energy varies.

Throughout our investigations of chemical fragmentation, one of the main issues was the annotation of spectra peaks with corresponding molecular structures. To ease this process, we developed a tool that helps visualizing molecules and their fragments. In a future version, this tool should be coupled to spectral libraries and transformed into a public peak annotation platform. Data generated on this platform would be retained to create training sets. Such sets would then be used to train an automated annotation algorithm.

Finally, a better understanding of chemical fragmentation patterns remains of high interest, since they can play a crucial role in the identification process. Particularly, the identification of substances absent from the reference library could be eased by proposing series of theoretically computed propositions. In that respect, chemical compound fragmentation would substitute the library search identification when this strategy reaches a dead end. During our investigations, we discovered that fragmentation mechanisms were directed by complex rules, only partially understood by a few experts. A successful advance in this field would probably involve such an expert to challenge and guide our hypotheses.

9.2 Collaboration with the HUG

Clinical toxicology was the first application of *SmileMS*, since the platform was initially developed in close interaction with the HUG. This interaction assessed the good performance of the platform in clinical toxicology, but also raised elements that should be improved. The three following sections present areas of improvement.

9.2.1 *MRM* versus *GUS*

In chapter 8, we introduced the HUG identification workflow, for which *MRM* proved to be more efficient than *GUS*, on the QTRAP instrument.
This approach is possible because the HUG laboratory is not working in a discovery set-up, where searched substances are not known \textit{a priori}. However, \textit{MRM} limits the number of substances that can be searched in one analysis, providing clinical laboratories with an argument to opt for a \textit{GUS} approach, as they tend to screen an always increasing number of substances.

In a \textit{GUS} context, \textit{SmileMS} generates an important number identifications which are pertinent from a matching point of view, but partially not interesting for the analyst. This is due to the fact that, in this technique, every molecule is fragmented and consequently generates a signal, whereas only selected molecules do so in the \textit{MRM} approach. This highlights a potential improvement in the direction of letting users predefine identification targets at the \textit{SmileMS} level. This targeted identification (as opposed to targeted screening) could generate clear results, with a removable filter allowing to consult the entire result list if needed.

9.2.2 Cross-platform identification

At the HUG, the use of a triple quadrupole and a 3D ion trap raised the question of identifying data of the first instrument against data of the other instrument. Though we showed the relatively good performance of \textit{X-Rank} in heterogeneous identifications, there is room for improvement.

First, since \textit{X-Rank} is based on statistical observations, it would be interesting to observe statistical distributions of cross-platform matchings. If this leads to distributions close to the ones generated for standard identifications, it would be possible to train \textit{X-Rank} with cross-platform data. On the other hand, if distributions are largely dissimilar, a new scoring should be conceived and benchmarked against \textit{X-Rank}.

Another approach consists in merging spectra of different MS levels (\textit{i.e.} \textit{MS}_2, \textit{MS}_3, ...) or acquired under different fragmentation energies. This approach is based on the assumption that features produced by one instrument, under certain conditions, can be found on another instrument, under different conditions. For example, a spectrum with two peaks generated by the 3D trap could be reproduced on the QTRAP, but at two different collision energies (\textit{i.e.} each of the two collision energies would produce one of the two peaks). In this situation, the optimal match between ion trap and QTRAP data would be obtained by merging the two QTRAP spectra.

To support these investigations, we need to dispose of large quantities of well annotated spectra, acquired on different instruments. In this respect, a spectra sharing platform would probably provide enough data to test new
hypotheses.

9.2.3 Integration with MS manufacturer software

Finally, in chapter 7, we described the integration with Bruker Daltonics software. At the HUG, this first integration enables to use *SmileMS* via the normal acquisition workflow of the 3D ion trap. On the other hand, QTRAP analyses lack such kind of facilitating process. An integration with this second instrument software should be done to ease its routine use.

More globally, a future challenge will consist in the integration of a maximum of MS instrument software. Integrations are interesting from a user point of view, facilitating the identification process, but also from a performance point of view. Indeed, they enable *SmileMS* to access low level information that can be used to improve identification accuracy. Typically, on the ion trap instrument, the fragmentation energy is not reported in standard data export, but can be accessed thanks to our integration work.

9.3 Beyond clinical toxicology

Although this thesis focuses on medical applications, *SmileMS* applies to several other fields.

A first example is the use of the platform for forensic investigations at the forensic center in Lausanne\(^2\). This second application is interesting in the sense that it implies different requirements, compared to the use of *SmileMS* at the HUG. For example, the speed of analysis is not a major factor, but the sensitivity of the identification must be very high. Indeed, whereas the HUG laboratory is often concerned with intoxications caused by high concentrations of a substance, identifications conducted at the forensic center must allow to detect very low concentrated drugs. A forensic center is also concerned with legal requirements that can impose certifications, specific development and documentation of the software.

Other applications such as food control, anti-doping, or pesticides analysis were also approached. Just as for the forensic center in Lausanne, each of these fields is based on LC-MSMS technology, but raises specific questions. Anti-doping is a good example of the need to support GUS approaches. This field is separated into two categories: routine testing and method development. While the routine testing activity is based on well

\(^2\)This application is presented in the thesis of Roman Mylonas.
calibrated methodologies that list the substances to screen, method development aims at going faster than drug users. To achieve that goal, it is important to screen for a very broad range of substances, without \textit{a priori} restrictions.

9.3.1 Current limitations

On the other hand, \textit{SmileMS} is currently not adapted to numerous situations. The two following paragraphs describe examples of analysis workflows that are currently not supported by \textit{SmileMS}. In that respect, they represent areas of improvement for the platform.

\textbf{a) MS\textsubscript{1} identification}

\textit{MS\textsubscript{1}} identification still concerns a broad range of applications. This technique is even growing with the apparition of always more accurate instruments. In geneva, the center for veterinary investigations relies on \textit{MS\textsubscript{1}} identification, for example. \textit{SmileMS} is currently not adapted to this kind of application, because of the different algorithmic approach it requires, and because it implies a different representation of data. For example, \textit{MS\textsubscript{1}} identification is centrally based on isotopic pattern recognition, whereas our scoring approach completely neglects this information.

Nonetheless, the use of \textit{MS\textsubscript{1}} information is interesting in at least two drastically opposite situations: when an identification is trivial and when a \textit{MS\textsubscript{2}} identification seems dubious. In the first case, a first \textit{MS\textsubscript{1}} screening could be used to analyze spectra that does not present any identification challenge. This strategy would avoid a resourceful \textit{MS\textsubscript{2}} identification. In the second case, \textit{MS\textsubscript{1}} could be used as a complement to \textit{MS\textsubscript{2}} identification. Due to similar molecular sub-structures, several molecules tend to produce similar fragmentation patterns. Consequently, the distinction of these molecules at the \textit{MS\textsubscript{2}} level is difficult. In this context, studying the \textit{MS\textsubscript{1}} isotopic signature could help discriminating ambiguous identifications.

9.3.2 LC-MS pre-filtering

Applications such as metabolomics deal with large amount of experimental data, though only a few spectra are interesting candidates. In order to efficiently reduce the number of spectra reaching an identification stage, a LC-MS filter is applied. During this step only retention time and precursor
mass information are considered. Spectra that pass this first filter are then identified and potentially lead to further investigations. This kind of identification workflow differs too much from the workflow used in SmileMS. As a result, SmileMS could not be currently applied to metabolomics study, except maybe if its role was restrained to identification aspects only.

On the other hand, metabolomics teaches us that data sets can grow enormously, engendering an unbearable number of spectral comparisons. The LC-MS pre-filtering strategy aims at reducing this number of comparisons by cutting into the number of experimental spectra analyzed. In preparation of the growth of analyzed data, another approach could be tested: the reduction of the search space. As opposed to the selection of experimental spectra, the reduction of search space aims at reducing the number of reference spectra against which an experimental spectrum is confronted. This practice has been more often employed in proteomics, where reference libraries, when used, tend to be gigantic. One of the possible strategies would consist in clustering spectra and thus obtain a classification tree of these spectra. The comparison engine would then visit branches of this tree, discarding entire groups of spectra in the case where a branch is judged non-pertinent from its root. This first approach could potentially avoid numerous worthless and costly comparisons.
Chapter 10

Publications, presentations and posters


• Yann Mauron, Roman Mylonas, Alexandre Masselot, Pierre-Alain Binz, Marc Fathi, Veronique Viette, Denis F Hochstrasser, Frederique Lisacek. A New Robust Library Search Algorithm for LC-MSMS of Small Molecules and its Spectral Library Sharing Mechanis. 11th International Congress of TDM & Clinical Toxicology, Canada, October 3-8, 2009 (Poster)

• Roman Mylonas, Yann Mauron, Alexandre Masselot, Olivier Philippe, Pierre-Alain Binz, Veronique Viette, Marc Fathi, Denis F Hochstrasser, Frederique Lisacek, Sebastian Goetz, Birgit Schneider, Jens Vagts, Carsten Baessmann. A New Approach for Acute Clinical Toxicology Based on Ion Trap MSMS Library Search. ASMS 2009, Philadelphia, May 31 - June 4, 2009 (Poster)


• Roman Mylonas, Yann Mauron, Alexandre Masselot, Pierre-Alain Binz, Denis F Hochstrasser, Frederique Lisacek. SmileMS; A new Mass Spectrometry Based Identification Platform for Small Molecules. 25th Montreux Symposium on LC/MS, Montreux (CH), November 12-14, 2008 (Poster)


Bibliography


[Brv09] Brvar M, Fokter N, Bunc M, and Mozina M. The frequency of adverse drug reaction related admissions according to method of
detection, admission urgency and medical department specialty. 


BIBLIOGRAPHY


Index

A

Acquisition, 35
Adobe Flex, 132
Adverse reaction, 21
Agile programming, 115
Analyser, 28, 30
Antidepressant, 21
Antipsychotic, 21
Atmospheric Pressure Chemical Ionization (APCI), 29

B

Benzodiazepine, 21
Bioinformatics, 16, 17
Biology, 16
Bioorganic chemistry, 50
BIRD, 32
Business logic, 122, 134

C

CAS, 83
CDK, 106
Chemical compound annotation, 83
Chemical fragmentation, 41, 50, 91
CID, 32, 83
Clinical environment, 34
Clinical toxicology, 20–22
Complexity, 17
Computer analysis, 31

D

Data layer, 124, 136
Data processing, 144
Decision tree, 75
Detector, 28
Disease, 20
Distance Measures, 39
Dot Product, 39
Drug design, 50

E

ECD, 32
Electrospray Ionization (ESI), 29
ETD, 32
Exploratory model, 114

F

Fragment Identificator, 43
Fragmentation, 29
Fragmentation algorithm, 99
Fragmentation energy, 93
Fragmentation graph, 92, 93, 96, 98
Fragmentation software, 43, 97
FTICR, 33

G
Galenic formulation, 50
Gamma distribution, 57
GC-MS, 23, 24, 46
Geneva University Hospitals, 22
Genomics, 17
GUS, 34, 47, 53, 164, 165
GWT, 132

H
Hibernate, 138
Hierarchical clustering, 69

I
Identification algorithm, 38, 46, 55, 161
Identification platform, 44, 53, 109, 143
Immunocassay, 23, 24
In-source fragmentation, 30
Interface, 123
Internal energy, 29, 42
Intoxication, 21
Ion source, 28
Ion trap, 33
Ionization, 29
IRMPD, 32
Iterative development, 113

J
JPL, 136

L
Layers of information, 82

LC-MSMS, 23, 25, 26, 34, 46, 56, 159
LC-UV, 23
Library building, 84, 169
Library Management, 171
Library management, 84, 150, 169
Library search, 37, 38
Library sharing, 84, 85

m/z, 17
Mass Frontier, 43, 51
Mass spectrometer, 28
Mass spectrometry, 17
Mass spectrometry applications, 18
Matrix effects, 26
Molecular biology, 16
MRM, 34, 53, 164, 165
MS Fragmenter, 43, 51
MS Interpreter, 43, 51
MS1, 29
MSn, 29
Multi-tier architecture, 119
mzML, 36
mzXML, 36

N
Neural network, 75
NIST, 37, 51
Noise filtering, 145

omics, 16

P
PBM, 39
Poor spectra, 163
Proteomics, 17
Prototyping, 113
PSI, 52

Q
Quadrupole, 33

R
R, 69
Ramp, 39
Rearrangement, 29, 42
Receiver Operating Characteristic (ROC), 63
REMEDi, 23, 158, 159
Reuse model, 114

S
Scientific exploration, 110
Scoring model, 46, 55, 131, 161
Screening, 22
Sector magnet, 33
Share peak count, 39
SID, 32
Single-stage mass spectrometry, 30
Software, 34
Software engineering, 44, 51, 109, 112
Software process model, 112
Spectral Library, 40, 79
Spectral library, 49, 80, 125
Spiral model, 114
Spring framework, 133, 134
Spring Security, 128
Support vector machine, 77

T
TOF, 33
Toxicology, 21

U
User interface, 121, 132, 148

W
Waterfall model, 113
Web service, 121, 133
Weka, 74

X
X-Rank, 48, 55, 61, 69, 160
**Glossary**

<table>
<thead>
<tr>
<th>Term</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACL</td>
<td>129</td>
</tr>
<tr>
<td>An Access Control List (ACL) is a list of permissions attached to an object. An ACL specifies which users are granted access to objects, as well as what operations are allowed to be performed on given objects.</td>
<td></td>
</tr>
<tr>
<td>Adobe Flex</td>
<td>132</td>
</tr>
<tr>
<td>Adobe Flex is a software development kit released by Adobe Systems for the development and deployment of rich Internet applications.</td>
<td></td>
</tr>
<tr>
<td>Adverse reaction</td>
<td>21</td>
</tr>
<tr>
<td>In pharmacology, any unexpected or dangerous reaction to a drug.</td>
<td></td>
</tr>
<tr>
<td>Bioorganic chemistry</td>
<td>50</td>
</tr>
<tr>
<td>Bioorganic chemistry represents a merger of organic chemistry and biochemistry. [Leo94]</td>
<td></td>
</tr>
<tr>
<td>Bitset</td>
<td>123</td>
</tr>
<tr>
<td>A Bitset is an array data structure which compactly stores individual bits (boolean values)</td>
<td></td>
</tr>
<tr>
<td>Data Access Object</td>
<td>129</td>
</tr>
<tr>
<td>The Data Access Object (DAO) implements the access mechanism required to work with the data source.</td>
<td></td>
</tr>
<tr>
<td>Dendrogram</td>
<td>69</td>
</tr>
<tr>
<td>A dendrogram is a tree diagram frequently used to illustrate the arrangement of the clusters produced by hierarchical clustering.</td>
<td></td>
</tr>
<tr>
<td>Drug design</td>
<td>50</td>
</tr>
<tr>
<td>Process of finding new medications based on the knowledge of the biological target. [Sch05]</td>
<td></td>
</tr>
<tr>
<td>Electrospray Ionization (ESI)</td>
<td>29</td>
</tr>
<tr>
<td>Electrospray Ionization is a type of ionization. In this technique, eluents from the column are nebulized in a chamber that contains a strong</td>
<td></td>
</tr>
</tbody>
</table>
electrostatic field and a drying gaz. This results in dissociated analyte molecules to evaporate and to concentrate charges. (see [Cec01] and [Agi01] for further details)

FTICR ................................................................. 30
FTICR or Fourier transform ion cyclotron resonance mass spectrometry is a type of mass spectrometer based on the cyclotron frequency of the ions in a fixed magnetic field.

Galenic formulation ............................................. 50
Galenic formulation deals with the principles of preparing and compounding medicines in order to optimize their absorption.

GUS ................................................................. 34
A general unknown screening (GUS) is a drug screening for which no a priori, in terms of substance desired to identify, has been decided. GUS is often replaced by targeted screening, such as MRM, nowadays. However, it is important to mention that this replacement is motivated by a lack of efficient solution to handle GUS.

Interface ......................................................... 123
An interface is an abstraction that an entity provides of itself to the outside.

IT ................................................................. 30
An ion trap is a combination of electric or magnetic fields, that captures ions in a vacuum system or tube.

Leave-one-out (LOO) ........................................... 64
In this thesis, LOO refers to identification conducted with a data set identified against itself. Scores are computed for each pair of spectra, except when the two compared spectra are identical.

library search .................................................. 37
In mass spectrometry, a library search describes the approach where experimental spectra are compared to a library of spectra. From those comparisons the identification of the experimental spectra is deduced.

Linear dynamic range ....................................... 33
Linear dynamic range is the range over which ion signal is linear with analyte concentration [McL01].

Mass accuracy ............................................... 33
The mass accuracy is the ratio of the mass-to-charge measurement er-
ror (i.e., the difference between the measured M and the true M) divided by the true mass-to-charge and is usually stated in terms of parts per million [McL01].

Mass range .................................................................................................................. 33

The mass range is the range of mass-to-charge ratios amenable to analysis by a given analyzer [McL01]. (Only the upper limit is given)

Mass resolving power ................................................................................................. 33

The term mass resolving power, according to the recommendations of the Measurements and Standards Committee of the American Society for Mass Spectrometry, is defined as $\frac{M}{\Delta M_x}$, where $\Delta M_x$ is the difference in mass-to-charge between two adjacent peaks in a mass spectrum that are of equal size and shape with a specified amount of overlap, the subscript $x$ denotes the overlap criterion, and $M$ is the average mass-to-charge ratio associated with the two peaks. For further details, see [McL01].

Matrix effects .................................................................................................................. 26

Matrix effects are the alteration of ionization efficiency by the presence of co-eluting substances.

MRM ................................................................................................................................ 34

MRM is a MS technique that allows to a priori select substances of interest. This technique is specifically adapted to triple quadrupoles. First (Q1) quadrupole ($MS_1$ level) selectively allows only the targeted parent ions to pass into second (Q2) quadrupole for collision-induced dissociation (CID) fragmentation. Third (Q3) quadrupole (MS2 level) selectively allows only a specific pre-identified fragment ion to pass to detector. [Pis07]

PSI .................................................................................................................................... 52

The HUPO Proteomics Standards Initiative (PSI) defines community standards for data representation in proteomics to facilitate data comparison, exchange and verification. Standards used for proteomics are widely used for small molecules.

QqLIT ................................................................................................................................. 30

A QqLIT mass spectrometer is a triple-quadrupole mass spectrometer with a linear ion trap mass spectrometer replacing the third quadrupole.

QqQ .................................................................................................................................... 30

A triple quadrupole mass spectrometer is a tandem mass spectrometer
consisting of three quadrupole mass spectrometers. The first (Q1) and third (Q3) quadrupoles serve as mass filters, whereas the middle (Q2) quadrupole serves as a collision cell. A precursor ion that is selected in Q1, is fragmented in Q2 and the subsequent fragments are passed through to Q3 where they may be filtered or scanned.

**QqTOF** ................................................................. 30
A QqTOF mass spectrometer is a triple-quadrupole mass spectrometer with a time-of-flight mass spectrometer replacing the third quadrupole.

**Quadrupole mass filter** ............................................ 31
A quadrupole mass filter consists of four parallel metal rods which allow to filter sample ions, based on their mass-to-charge ratio (m/z).

**R** .............................................................. 69
R is a free software environment for statistical computing and graphics.

**Receiver Operating Characteristic (ROC)** ......................... 63
In signal detection theory, a receiver operating characteristic (ROC), or simply ROC curve, is a graphical plot of the sensitivity vs. \( (1 - \text{specificity}) \) for a binary classifier system as its discrimination threshold is varied[Par04].

**RETOF** ................................................................. 30
RETOF or Reflectron TOF is a mass spectrometer uses a constant electrostatic field to reflect the ion beam toward the detector. The more energetic ions penetrate deeper into the reflectron, and take a slightly longer path to the detector.

**Sector magnet** .......................................................... 31
A sector magnet is a mass spectrometer that uses a static magnetic sector as a mass analyzer.

**Sectors Hybrid (BqQ)** .................................................. 30
A sector hybrid (BqQ) is the combination of a sector instrument with a collision quadrupole and quadrupole mass analyzer.

**SMILE** ................................................................. 106
The SMILE or simplified molecular input line entry specification is a specification for unambiguously describing the structure of chemical molecules using short ASCII strings.

**Speed** ................................................................. 33
Speed refers to the time frame of the experiment and ultimately is used
to determine the number of spectra per unit time that can be generated [McL01].

Spring Security ................................................................. 129
Spring Security is a Java/Java EE framework that provides advanced authentication, authorization and other security features for enterprise applications.

Stakeholder ................................................................. 116
Stakeholders are persons or organizations (legal entities such as companies, standards bodies) which have a valid interest in the system.

Tandem TOF ................................................................. 30
Tandem TOF is a tandem mass spectrometry method where two time-of-flight mass spectrometers are used consecutively.

TOF ................................................................. 31
Time-of-flight mass spectrometry is a method of mass spectrometry in which ions are accelerated by an electric field of known strength. The velocity of the ion depends on the mass-to-charge ratio. The time that it subsequently takes for the particle to reach a detector at a known distance is measured.

Weka ................................................................. 75
Weka is open source software that contains a collection of machine learning algorithms for data mining tasks. It can be directly applied to a dataset or called from Java code. Weka presents the possibility of testing various approaches, with the same software.