New approaches for processing and annotations of high-throughput metabolomic data obtained by mass spectrometry

Alexis Delabrière

université PARIS-SACLAY

NNT : 2018SACLS359

Thèse de doctorat

New approaches for processing and annotations of high-throughput metabolomic data obtained by mass spectrometry

Thèse de doctorat de l'Université Paris-Saclay
préparée à l'Université Paris-Sud

Ecole doctorale n°580 Sciences et technologies de l'information et de la communication (STIC)
Spécialité de doctorat: Informatique

Thèse présentée et soutenue à Gif-sur-Yvette, le 16/10/2018, par

ALEXIS DELABRIERE

Composition du Jury :

Florence d'Alché-Buc Professeure des Universités, Telecom ParisTech, Pris Saclay (Groupe Signal, Statistiques et Apprentissage Automatique)	Présidente
Steffen Neumann Directeur de Recherche, IPB-Halle (Groupe Bioinformatique et Spectrométrie de Masse)	Rapporteur
Fabien Jourdan Directeur de Recherche, INRA (Unité Toxalim)	Rapporteur
Michel Liquière Maître de Conférences, Université de Montpellier (Laboratoire d'Informatique, de Robotique, et de Microélectronique)	Examinateur
Alain Denise Professeur des Universités, Université Paris-Sud, Paris-Saclay (Laboratoire de Recherche en Informatique et Institut de Biologie Intégrative de la Cellule)	Directeur de thèse
Etienne Thévenot Ingénieur Chercheur, CEA (Laboratoire Analyse des Données et Intelligence des Systèmes)	Encadrant
François Fenaille Ingénieur Chercheur, CEA (Laboratoire d'Etude du Métabolisme des Médicaments)	Invité

Acknowledgements

I would like to express my sincere gratitude to Mr. Etienne Thévenot for his crucial involvement at every steps of this project, from his definition to the writing of this manuscript. I especially thank him for his humanity, his enthusiasm, and for the finding of solutions at the difficult moments of this thesis.

I also express my gratitude to Mr. Alain Denise for the direction of this thesis and for his contributions to this work, both scientific and human, and finally for his enthusiasm and his availability at all stages of the thesis.

I thank my rapporteurs Mr Steffen Neumann and Mr Fabien Jourdan for agreeing to evaluate this thesis, as well as the examiners Mrs. Florence d'Alché-Buc and Mr. Michel Liquière.

I thank the CEA and especially the High Commissioner for funding this thesis, and providing me with very comfortable working conditions.

This thesis has also benefited greatly from the collaboration with the CEA Laboratoire d'Etudes du Métabolisme des médicaments. I especially thank Mr. François Fenaille, Mrs. Anne-Laure Damont, Mrs. Marie-Françoise Olivier, Mr. Christophe Junot, Mrs Thais Hautbergue, Mr Ulli Hoheynester, Mr. Benoît Colsch and Mr. Jean-Claude Tabet. and Mrs. Anne Warnet for the many scientific contributions they offered and for their enthusiasm for my work, which often exceeded mine.

I thank Mrs. Natacha Lenuzza for her scripts that helped me when I discovered this subject, and for the many interesting conversations that we had at all the stage of this work.

I thank all my laboratory the LADIS for the friendly atmosphere and for the many interesting discussions which occurred there.

I also want to thank the LADIS PhD students with whom I shared my highs and lows, especially Anne Morvan and Maxime Velay.

I would like to thank my mother for reading and correcting many pages of this manuscript.

And finally, I thank the other person who had to cancel his weekend and who played an essential role of psychological support, my wife.

Preamble

The rise of global molecular (i.e. *omics*) sciences has considerably expanded our knowledge of many biological systems. Compared to genomics, or even proteomics, metabolomics appeared more recently, about two decades ago. The metabolome is the set of all small molecules (metabolites) involved in biochemical reactions, and thus provides specific informations about the phenotype. Moreover it is the end product of cellular metabolism. As such, it is of great potential to monitor any physio-pathological variations in a living system. This proximity with the phenotype leads to a greater variability than other omics data. It is therefore a challenge for data analysis, and large cohorts are thus required to increase the statistical power of the studies. One of the reference analytical approaches in metabolomics, Liquid Chromatography coupled to Mass Spectrometry (LC-MS), however, is nevertheless low throughput. New protocols are therefore developed, such as Flow Injection Analysis coupled to High-Resolution Mass Spectrometry (FIA-HRMS) to bypass the chromatographic step. In parallel, the design of robust computational workflows to process such specific raw signals is pivotal. In the the first part of this PhD, we describe the design and implementation of the first workflow for the preprocessing of FIA-HRMS data.

A specificity of metabolomics is the chemical diversity of the molecules. Since only partial information is provided by the analytical techniques (e.g. the mass-to-charge ratio, m/z in the case of MS approaches), structural annotation is a challenge for the majority of the detected compounds. Additional experiments are therefore used, such as tandem mass spectrometry (MS/MS), to obtain new informations about the fragmentation of the molecule, which can be compared to tandem spectra from chemical standards. With the recent emergence of high-speed acquisition instruments, hundreds of MS/MS spectra from distinct molecules can be generated in a single acquisition. Automated processing and analysis of such collections of spectra has therefore become critical. On the one hand, predictive approaches based on *in silico* fragmentation modeling have been developed. On the other hand, mining strategies have been recently described to analyze the similarities between spectra, by using molecular networks or pattern mining. Whereas the modeling of the physical fragmentation process is central to *in silico* fragmentation methods, it is not used by similarity based approaches, thus making the interpretation of the detected similarities more challenging. We therefore developed in the second part of the PhD a new method to find structural similarities within MS/MS spectra collections, based on an innovative graph mining algorithm.

Structure of the manuscript

This thesis starts by an introductory part describing the key terms and concepts used in mass spectrometry based metabolomics (Section 1). Next, an overview of metabolomics is presented (Section 2), focusing on the preprocessing and annotation challenges which led to our project. The datasets used in this thesis are also described in chapter (Chapter 3).

The two next parts of the manuscript describe our contributions about 1) the processing of Flow Injection Analysis coupled to High-Resolution Mass Spectrometry (FIA-HRMS) data (proFIA software; Part II), and 2) the mining of structural patterns within MS/MS spectra collections (MineMS2 software; Part III). The two parts can be read independently. They both start with a bibliographic introduction giving a technical overview of the problem, followed by the design and implementation of our algorithms, their validation on two real datasets, and a discussion about the perspectives of the proposed methods.

The thesis ends with a global conclusion about the approaches and software developed in this PhD, and their scientific impact.

Contents

Part I

Introduction

Chapter 1

Nomenclatures

1.1 Acronyms

AFG: Acyclic Flow Graph, the type of graph mined in chapter 11.

BFS: Breadth-First Search, a classical graph traversal algorithm

CWT : Continuous Wavelet Transform, a widely used algorithm for peak detection, see section 4.2.3.

DFS: Depth-First Search , a classical graph traversal algorithm.

EIC: Extracted Ion Chromatogram a slice of Mass Spectrometry data in a limited m/z range and eventually in a limited time range, see section 1.3. item[ESI:] Electro Spray Ionization, a reference technique of ionization used in metabolomics, see section 5.1.2 for a more complete description.

FIA-HRMS: Flow Injection Analysis coupled to High Resolution Mass Spectrometry, the high-throughput technique studied in part II.

k-LMDF: , k Left Most Depth First, a canonical form of an AFGdefined in chapter 11.

LC-MS: Liquid Chromatography coupled to Mass Spectrometry, a reference technique in metabolomics.

ME: Matrix Effect. The combined effect of all components of the sample other than the analyte on the measurement of the quantity (Murray et al. 2013), it is described in more detail in section 5.1.2.

MS: Mass Spectrometry.

MS/MS: Tandem MS spectrum obtained by isolating a single molecule and fragmenting it.

TIC: Total Ion Chromatogram, a curve obtained by summing all the intensities present on each mass spectra, for each time point. See 1.3.

TOF: Time-of-Flight mass spectrometer, one of the most widely used mass spectrometer. A very quick description of his functioning is given in part

1.2 Glossary

Because of the difficulty to define specific data terms without many cross-references, the vocabulary relative to Mass Spectrometry data is described in a dedicated section 1.3 similarly the vocabulary of graph theory used in part III is described in section 9.5.

Analyte : A molecule of interest from which needs to be quantified.

Carrier flow: The flow of solvent carrying a sample to the mass spectrometer.

Concentration curve: The gradient of concentration observed in function of time at a fixed point of the flow of an FIA system.

Fragment Ion An ion resulting from a fragmentation event visible on an MS-MS spectrum. See section 1.3 for more detail.

Losses Graphs : A graph representation of an MS-MS spectrum defined from a set of MS-MS spectra in chapter 10.

Mass analyzer: A mass analyzer is the component of the mass spectrometer that takes ionized masses and separates them based on charge to mass ratios.

Matrix Effect: The combined effect of all components of the sample other than the analyte on the measurement of the quantity (Murray et al. 2013), it is described in more detail in section 5.1.2.]Acyclic Flow Graph, the type of graph mined in chapter 11.

Matched filtration: A pattern detection procedure where the pattern is convolved to the input sequence, and match are found as maxima on the convolved sequence, see section 4.2.3.

Neutral loss A neutral molecule ion resulting from a fragmentation event. See section 1.3 for more detail.

Orbitrap: An High Resolution mass spectrometer is an ion trap mass analyzer consisting of an outer barrel-like electrode and a coaxial inner spindle-like electrode that traps ions in an orbital motion around the spindle.

Precursor Ion The isolated ion leading to an MS-MS spectrum, see section 1.3 for more detail.

Sample peak: The common common temporal profiles of all the concentration curves in an ideal FIA system without any retention.

1.3 Specific MS and MS/MS nomenclature

In this section we define the key MS terms and concepts which will be used throughout the thesis. A short introduction can be found in R. Smith et al. 2014, as well as a more detailed glossary in (Murray et al. 2013). The reference chemical terminology is defined by the International Union of Pure and Applied Chemistry (IUPAC Gold Book).

1.3.1 Description of MS data

An example of the data generated by an FIA-HRMS acquisition is shown in Figure 1.1. The data are 3-dimensional: m/z, **time**, and **intensity**, with m/z being a mass-to-charge ratio.

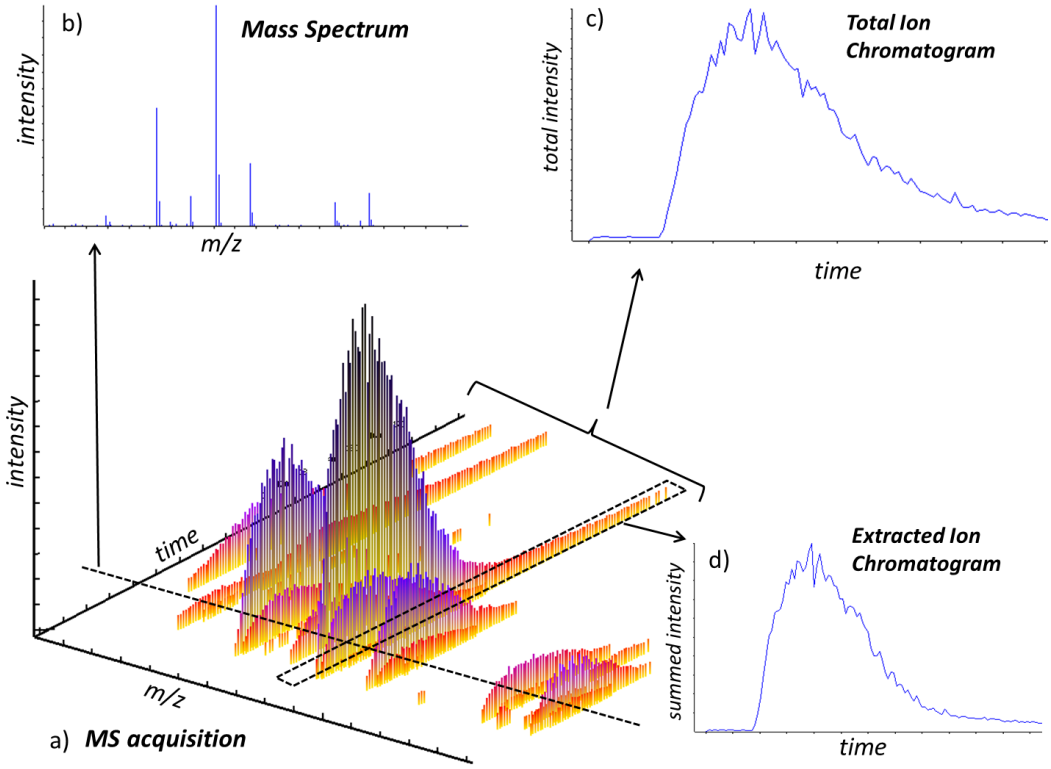
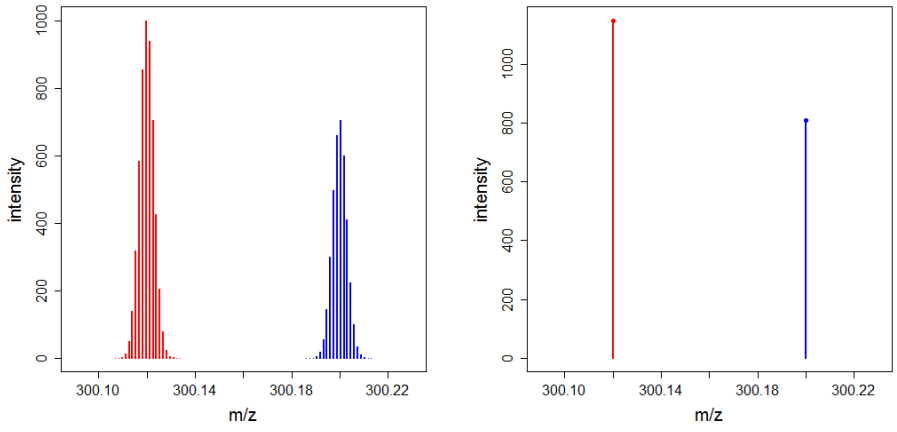


Figure 1.1: **Nomenclature used to describe MS data:** a) 3D representation of the data, b) scan at a specific time point: a mass spectrum, d) slice within a specific m/z window: an Extracted Ion Chromatogram (EIC), and c) slice covering the whole m/z acquisition range : the Total Ion Chromatogram (TIC), i.e., the signal obtained by summing all intensities for each scan

An MS run consists of multiple acquisitions (typically every second): for each scan,

the instruments provides a measure of the intensities detected across the whole m/z range, a mass spectrum, which is a vector of m/z values and corresponding **intensities** (Figure 1.1b). The whole data set corresponding to a complete MS run for one sample can be represented as a 3D plot (Figure 1.1a), where all scans are shown side by side in the time dimension. By definition, the time scale is identical for all data points. In contrast, points in the m/z dimension are usually not binned to achieve maximal measurement accuracy: as a result, slight random variations are observed from one scan/spectrum to the other. To capture all ions from the same feature along the time dimension, slices covering a small m/z window are used (**Extracted Ion Chromatogram (EIC)**; Figure 1.1d). It is a 2D curve with time as abscissa and intensity as ordinate. Depending on the m/z width of the slice, multiple points from the same spectrum may be selected: in that case, their intensities are summed. The EIC obtained by considering the full m/z range is called the **Total Ion Chromatogram (TIC)**, each intensity is the sum of the values of a whole mass spectrum (Figure 1.1c).

Profile vs centroid data



(a) (b)

Figure 1.2: **Profile and centroid m/z modes**. On these simulated data, a) m/z distributions can be observed before centroidization, whereas on b) each feature is represented by a unique m/z value in the centroid mode.

Mass spectrometers often sample distributions of m/z values corresponding to same ion. This distributions is clearly visible on the mass spectra (Figure 1.2a, data exhibiting these distributions are called "profile mode data" or "**profile** data". It is often more convenient from a storage and processing point of view to reduce each distribution to a single m/z value and its corresponding intensity (which may be the area of the peak, or its maximum). These data are called "centroided", or "in **centroid**

mode". Centroidization is discussed with more detail in section 4.2.1.

Low vs high resolution

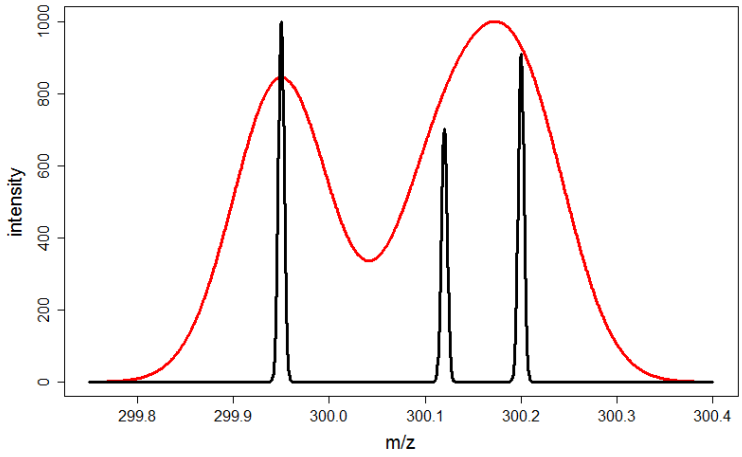


Figure 1.3: **Low and High resolution mass spectrometry**. Two signals have been simulated corresponding to a resolution of 4'000 (red) and 30'000 (black). The two peaks at a about 300.1 and 300.2 They cannot be separated at the sub-nominal, low resolution.

Multiple definitions of **resolution** have been proposed by IUPAC. Here we will use the ratio between the m/z value and the full width at half maximum (FWHM). It is important to keep in mind that the resolution of a mass spectrometer is defined at a specific m/z (since it decreases with large m/z values). Low resolution instruments were first used in metabolomics, in particular for high-throughput applications. Such mass spectrometers cannot separate ions below the nominal mass (i.e., **isobaric ions**, also called **isobars**; red signal in Figure 1.3). Today, most platforms are equipped with high-resolution mass spectrometers **HRMS** which enable to assign a unique chemical formula to m/z values up to a few hundreds Da (black signal in Figure 1.3). The data and algorithms presented in this thesis are therefore designed for HRMS data.

1.3.2 Description of MS/MS data

The m/z value alone is not sufficient to assign a chemical structure to a feature (e.g., all chemical isomers have the same m/z). Tandem mass spectrometry is therefore used to analyze the fragmentation pattern of the molecule. During the first MS acquisition, ions are selected within a certain m/z range (**Isolation Windows**). During the subsequent collision induced dissociation step, these (parent) ions are fragmented. The

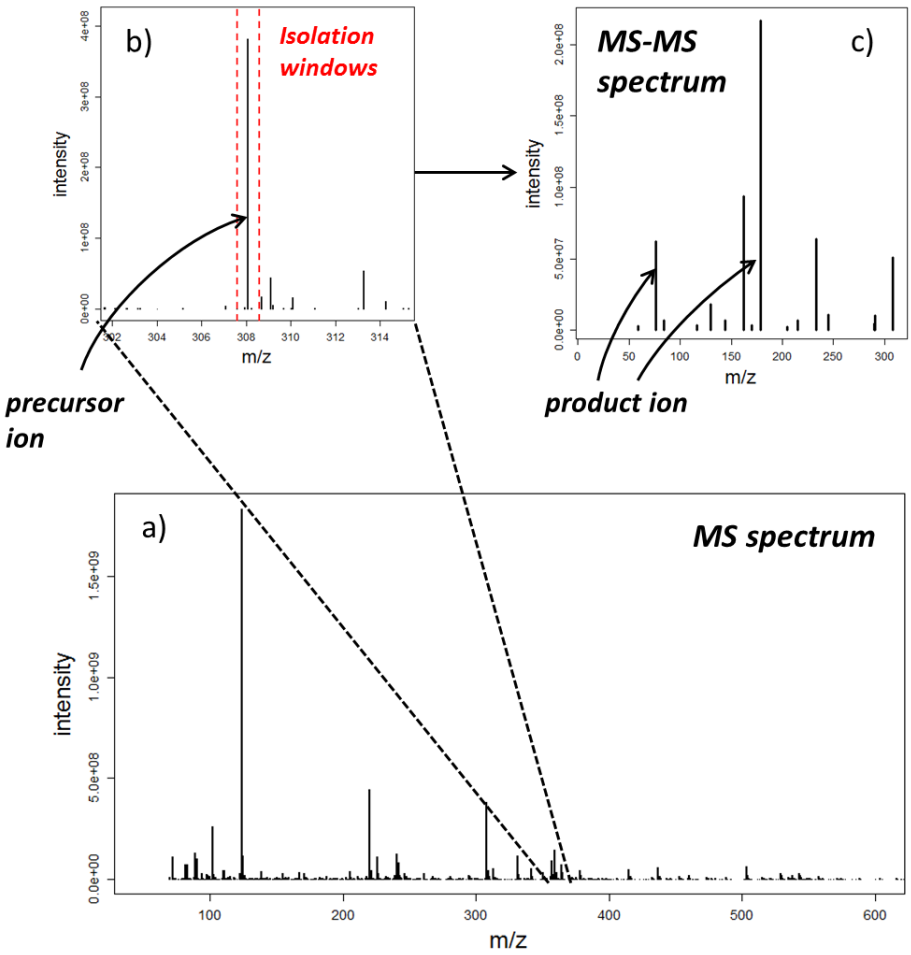


Figure 1.4: **Nomenclature for MS/MS data**. a) MS spectrum; b) zoom on the specific m/z isolation window used to produce the MS/MS spectrum shown in c)

fragments are analyzed during the second MS acquisition. MS/MS can be used to study the fragmentation of a few ions of interest (e.g., which have been highlighted by upstream statistical analysis of the MS spectra). Each fragmentation event generates at least two fragments, one **product ion** which will be detected in the **MS/MS (or MS^2)** spectrum because it keeps the charge, and one or several **neutral loss(es)** which would not be visible as they will be neutral (they can be inferred from the differences between the peaks from the parent and the product ions). With the improvement of the mass spectrometers in terms of sensitivity, mass accuracy, resolution and acquisition speed, new "non-targeted" (or "semi-targeted" strategies have emerged), such as **Data Dependent Acquisition (DDA)** and **Data Independent Acquisition (DIA)**(Fenaille et al. 2017). DDA and DIA were initially developed for proteomics but are now widely used in metabolomics. In DDA, the mass spectrometer selects the ions after the acquisition of the MS-spectrum, based on predefined criteria (e.g., the "top n" ions of maximal intensity or the presence of a characteristic ion). In DIA, all

the precursor ions within predefined (wide) isolation windows are fragmented.

Chapter 2

Bibliography

2.1 Metabolomics

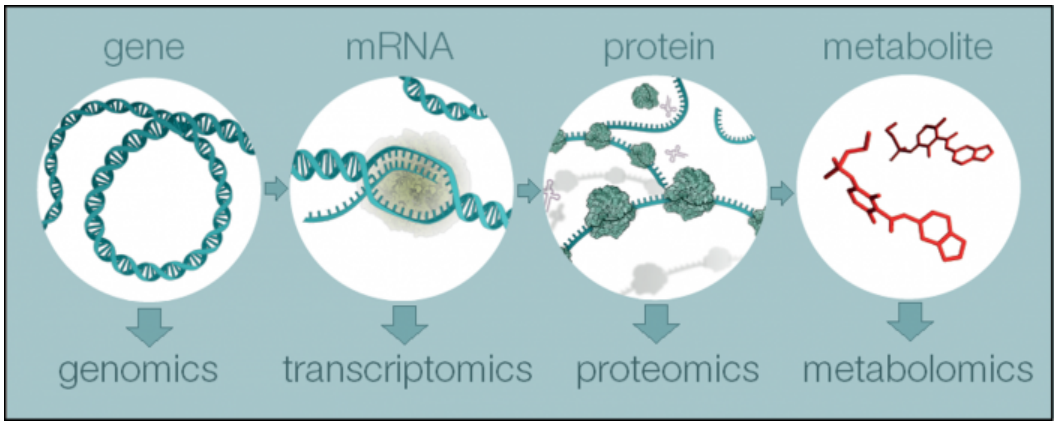


Figure 2.1: **Metabolomics among the main omics approaches (European Bioinformatics Institute website).**

Metabolomics (Oliver et al. 1998) is the global study of the small molecules (typically less than 1.5 kDa) present in a biological sample. Metabolites are the end products of regulatory processes in the organism (Figure 2.1), and are therefore of high interest to characterize the phenotype. Many strategies have been developed to study either the intra- or extracellular content, fluxes, or specific chemical families (Figure 2.2). Metabolomics has been applied to many fields, including health, nutrition, agroscience, and microbiology (Section 2.2). The set of all metabolites is referred as the **metabolome**: while an important number of metabolites have already been detected and characterized (the Human Metabolome DataBase, HMDB, contains more than 18,000 metabolites; David S Wishart et al. 2018), recent developments have shown

that a significant part of the metabolome remains unknown (Zamboni et al. 2015). The **chemical diversity** of the metabolome has led to the development of complementary observation techniques (Section 2.3).

2.2 Application of metabolomics

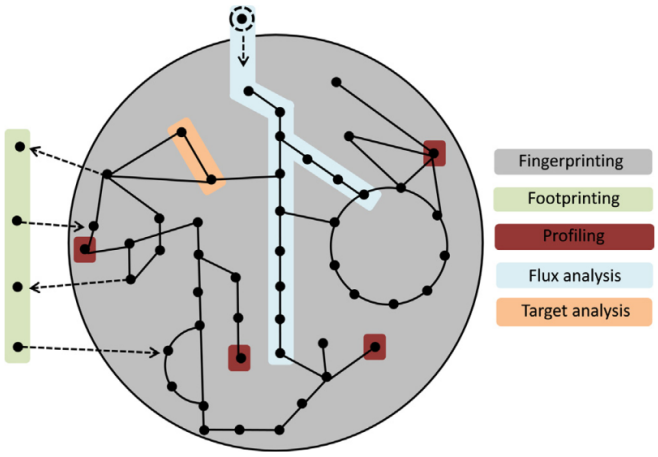


Figure 2.2: **Complementary strategies in metabolomics (from Emily G. Armitage and Barbas 2014)**. Fingerprinting involves the global screening for all detectable metabolites within the system under investigation. Footprinting (mainly referred to in *in vitro* cell systems) is the analysis of metabolites from the environment around the system under investigation and reveals information about metabolic exchange. Profiling is the screening of a particular class of chemicals, e.g. amino acids, for which standards are also analysed. Flux analysis is the tracing of one compound, usually isotope labeled carbon, through a particular pathway or set of pathways to determine the fate of the compound. Target analysis is the comparison of one or a few closely related target metabolites whose concentrations may change depending on the experimental conditions.

2.2.1 Biomarker discovery

Many metabolomics studies focus on the discovery of biomarkers representative of a specific physio-pathological condition. The aim of such **biomarker discovery** strategies may be either i) to define early and non-invasive molecular diagnostics, ii) to find new therapeutic targets, or iii) to monitor the response to a dedicated treatment (David S. Wishart 2016; Monteiro et al. 2013; Trivedi et al. 2017). The full validation of a biomarker is a long and complex process which consists of several steps (Nagana Gowda and Raftery 2013):

Discovery phase: An **experiment is designed**, and run, to **collect representative samples** from of distinct classes (e.g., case versus control). **Analytical chemistry** is used to detect either (Figure 2.2): as many compounds as possible (**metabolic fingerprinting**), or a preselected set of known metabolites (**metabolic profiling**). While the former strategy is semi-quantitative (i.e. relies on the comparison between the two conditions), the latter approach can be quantitative when standards are used. **Preprocessing** of the raw data from the instrument generates a sample by variable table of peak intensities. Finally, **statistical analysis** is performed to select the most discriminating features (**molecular signature**).

Identification: Unambiguous **chemical identification** of the selected candidates is mandatory for downstream clinical validation. This process involves both computational (e.g., matching with *in-house* or public **databases**, structure prediction using ***in vitro*** **fragmentation** and/or machine learning), and experimental tasks (e.g. additional **tandem MS** fragmentation experiments).

Validation: To limit the risk of false positives candidates, which may have been selected during the statistical analysis (e.g. type I error in univariate hypothesis testing or overfitting during multivariate modeling), the molecular signature has to be validated on a second, independent experiment (Broadhurst and Kell 2006; Castaldi et al. 2011).

Mechanism of action: Characterization of the biological role and the mechanisms of action are critical for a metabolite to be considered as a candidate biomarker for the clinic. This step involves many computational and experimental approaches such as metabolic network analysis, molecular biology, molecular imaging, etc. As a first step, metabolic networks may be used to provide information about the biochemical reactions and pathways involving the detected metabolites Caspi et al. 2018. Network analysis may also suggest new biomarkers of interest (Frainay and Jourdan 2017). The putative cascade of reactions may be studied dynamically (e.g. by fluxomics) by using isotope-labeled experiments Buescher et al. 2015. More generally, the investigation of the underlying mechanisms of disease has been the focus of recent technological advances (C. H. Johnson et al. 2016).

An example of such a comprehensive pipeline in metabolomics is provided by the study of cardiovascular disease by Z. Wang et al. 2011. Within an initial human cohort from 50 cases and 50 controls, 40 out of the 2,000 detected compound were found to be statistically significant. A total of 18 of these molecules were validated on an independent cohort. Three of the metabolites were unambiguously identified by NMR and MSn

experiments: choline, trimethylamine N-oxide (TMAO) and betaine. Supplementary experiments in mice highlighted the contribution of these metabolites to atherosclerosis. A similar strategy was applied to diabetes: the contribution of branched chain amino acids to insuline resistance was evidenced by targeted metabolomics followed by validation in mice (Newgard et al. 2009). Metabolic changes are also known to be critical in tumors (Hanahan and Weinberg 2011), and several biomarkers for distinct cancer types have been reported (Newgard 2017; M. Yang et al. 2013). Comprehensive reviews about metabolomics approaches for the discovery of disease biomarkers are available in David S. Wishart 2016 and Trivedi et al. 2017.

Metabolomic applications to Health also include the personalization of pharmacological treatments to the individual patient (patho)physiology (pharmacometabolomics; David S. Wishart 2016; Kaddurah-Daouk et al. 2008), and the study of the impact of nutrition and of the gut microbiome (Scalbert et al. 2014).

2.2.2 Insights into systems biology

Metabolomics has been used extensively to understand the metabolism as a network of biochemical reactions and discover new pathways (G.-F. Zhang et al. 2011; Frainay and Jourdan 2017). Furthermore, metabolomics provides unique informations about gene function Fukushima et al. 2014: recently, a high-throughput screening of *Escherichi coli* strains over-expressing individual proteins was developed: analysis of the concentration variations within a known mix of metabolites after incubation with each of the strains resulted in the discovery of more than 200 potential novel enzymes (Sévin et al. 2016). A review of the applications to enzyme discovery and annotation is available in Prosser et al. 2014.

The diversity of metabolite structures and functions has led the development of a wide range of analytical technologies (Rolin 2013) which are introduced in the next section.

2.3 Main analytical techniques

Analytical techniques have been historically divided into Nuclear Magnetic Resonance (NMR) or **Mass Spectrometry (MS)** based approaches (A. Zhang et al. 2012). A list of their main pros and cons according to is given in . While NMR is fast and reproducible, it is less sensitive than MS, especially when MS is coupled to chromatography

Technology	Advantages	Disadvantages	Refs
NMR spectroscopy	• Quantitative • Non-destructive • Fast (2–3 min per sample) • Requires no derivatization • Requires no separation • Detects most organic classes • Allows identification of novel chemicals • Most spectral features are identifiable • Robust, mature technology • Can be used for metabolite imaging (fMRI or MRS) • Can be fully automated • Compatible with liquids and solids • Long instrument lifetime (over 20 years)	• Not sensitive (LOD = 5 µM) • High start-up cost (>US$1 million) • Large instrument footprint • Cannot detect or identify salts and inorganic ions • Cannot detect non-protonated compounds • Requires larger sample volumes (0.1–0.5 mL)	17,18,35
GC-MS	• Robust, mature technology • Modest start-up cost (~$150,000) • Quantitative (with calibration) • Modest sample volume (0.1–0.2 mL) • Good sensitivity (LOD = 0.5 µM) • Large body of software and databases for metabolite identification • Detects most organic and some inorganic molecules • Excellent separation reproducibility • Many spectral features are identifiable • Can be mostly automated • Compatible with gases and liquids	• Destructive (sample not recoverable) • Requires sample derivatization • Requires separation • Slow (20–40 min per sample) • Cannot be used in imaging • Not compatible with solids • Novel compound identification is difficult	18–20
LC-MS	• Superb sensitivity (LOD = 0.5 nM) • Very flexible technology • Detects most organic and some inorganic molecules • Small sample volumes (10–100 µL) • Can be used in metabolite imaging (MALDI or DESI) • Can be done without separation (direct injection) • Has the potential to detect the largest portion of metabolome • Can be mostly automated • Compatible with solids and liquids	• Destructive (sample not recoverable) • Not very quantitative • Higher start-up cost (>$300,000) • Slow (15–40 min per sample) • Usually requires separation • Poor separation resolution and lower reproducibility versus GC-MS • Less-robust instrumentation than NMR or GC-MS • Not compatible with gases • Most spectral features are not yet identifiable • Novel compound identification is difficult • Short instrument lifetime (<9 years)	19,20,33,38

Table 2.1: **Main analytical techniques used in metabolomics (from (David S. Wishart 2016))**

(see Table 2.1 from David S. Wishart 2016). MS is therefore a technique of choice for biomarker discovery, and this thesis describes innovative methods for the analysis of MS data. We briefly describe below the classical experimental setup for metabolomics analysis with a mass spectrometer coupled to a chromatographic column.

2.3.1 Hyphenated Mass spectrometry setup

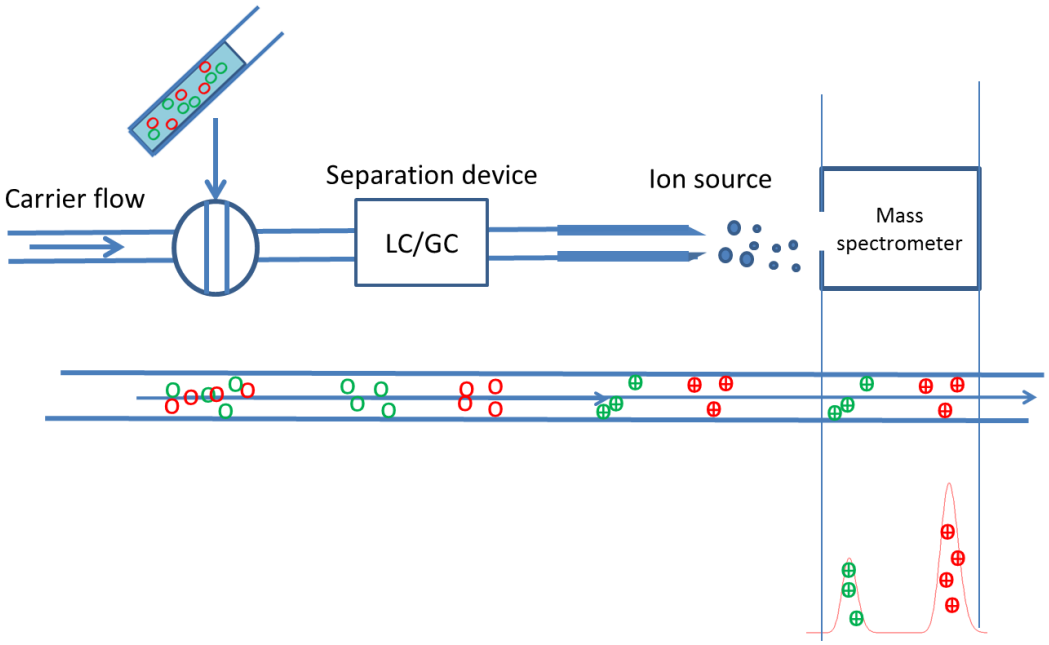


Figure 2.3: **Schematic view of an hyphenated MS setup.**

In classical acquisition by liquid chromatography (respectively gas chromatography) coupled to mass spectrometry, namely LC-MS (respectively GC-MS), the sample is injected into a carrier-flow which enters the chromatographic column which separates the compounds according to their physico-chemical properties (Figure 2.3). Before entering the mass spectrometer, a step of ionization (and, in the case of a liquid sample, desorption) is performed within the ionization source, to end up with ions in the gas phase which enter the mass spectrometer. Within the mass analyzer, ions are separated according to their mass-to-charge ratio m/z. MS technologies therefore generate complex 3 dimensional data (m/z, retention time, and intensity). As a consequence, efficient computational pipelines are required for the processing, statistical analysis, and annotation of metabolomics data (Junot et al. 2013).

2.4 Processing and analysis of MS-derived data

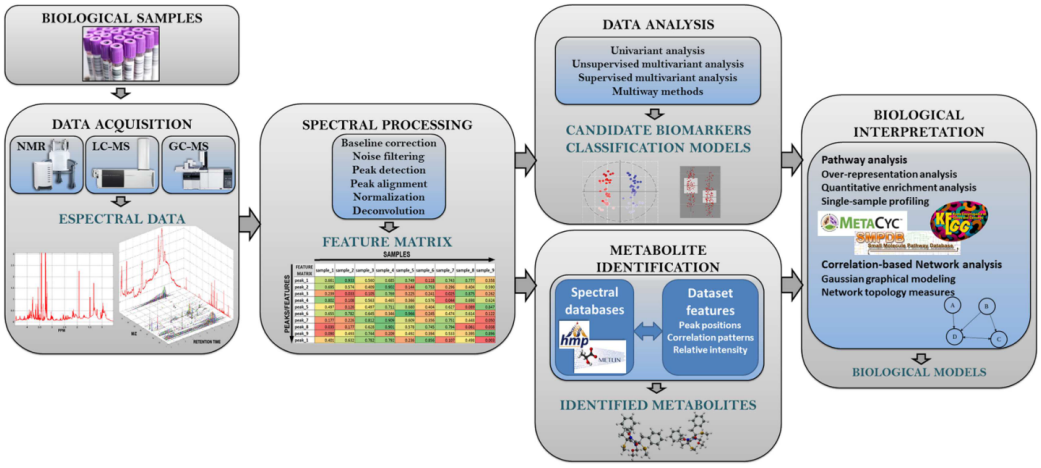


Figure 2.4: **Computational workflow for metabolomics data Alonso et al. 2015**

An overview of the computational workflow is described in Figure 2.4. First, **preprocessing** of the raw sample files from the MS instrument generates a sample by variable table of intensities. Second, **statistical analysis** is used to select the discriminating features according to the factor(s) of interest, and to build multivariate prediction models. Third, chemical and biological **annotation** of the selected variables is performed. Processing shares some similarities with proteomics, which also relies on MS instruments. Hypothesis testing and machine learning methods used for the statistical analysis of high-dimension datasets in transcriptomics and proteomics are also used in metabolomics. In contrast, metabolite annotation is specific, due to the chemical diversity of the compounds.

2.4.1 Preprocessing

Preprocessing consists of the detection and quantification of the features in the individual raw files, followed by the alignment between the samples to generate the sample by variable table of peak intensities (i.e. the **peak table**). As the first part of this thesis focuses on new computational approaches for preprocessing, this step will be more detailed in Section 4.1.

2.4.2 Statistical analysis

Statistical analysis of the peak table usually starts with the exploration of the data. Numerical and graphical summaries (e.g. with unsupervised approaches such as Principal Component Analysis) help detecting strong variations uncorrelated to the factors of interest such as signal drift, batch effect, outliers.

Missing values imputation

Missing values resulting from a failure during the preparation, acquisition, or preprocessing may be up to 10-40% of the intensities (Godzien et al. 2015; Emily Grace Armitage et al. 2015). Although several statistical methods can handle a certain amount of missing values (e.g. the NIPALS algorithm for Partial Least Square multivariate regression and classification (H. Wold 1966)), imputation methods are required for other methods. A post-processing approach was proposed in the XCMS reference software for LC-MS data preprocessing (C. Smith et al. 2006), whereby the region of the missing peaks in the raw data is integrated by using the m/z and retention time limits derived from the other samples. This strategy has multiple drawbacks, as the signals may deviate in both dimensions between the samples. Even though the integration region may be increased, this may eventually result in the inclusion of non-related signals. Alternatively, statistical methods applied to the peak table have been described: in particular, approaches based on k-NN (Hrydziuszko and M. Viant 2012) and random forest, were shown to outperform simpler strategies (such as the replacement by the mean or the median) consistently (Wei et al. 2018). To take into account the limit of detection, new methods based on left-censored distributions have been proposed, either adapted from a truncated version of the k-NN (Shah et al. 2017), or from a modified quantile regression (M. Lee 2010).

Unwanted variation correction

Biases affecting measurements may occur at any step: sample collection, storage, preparation, analysis. Such unwanted variations (i.e. non correlated to the factors of interest) need to be corrected by using quality control samples and statistical normalization procedures (Livera et al. 2015). A well-known bias during MS acquisition is the signal drift and batch-effect: when analyzing a large number of samples with MS (e.g. in the case of cohorts), a signal attenuation may be observed due to the contamination of the source (intensity drift). The MS instrument is therefore periodically cleaned,

Type of method	Method family	Method adopted
Unsupervised pattern recognition methods	Methods based on data projection to other dimensions	Principal Component Analysis (PCA)
		Nonnegative PCA
	Methods based on correlation analysis	Canonical Correlation Analysis (CCA)
		Sparse Canonical Correlation Analysis (SCCA)
	Cluster analysis methods	Hierarchical Clustering
Supervised classification methods	Methods based on PCA	Soft-Independent Model of Class-Analogy (SIMCA)
		Principal Component Analysis-Discriminant Analysis (PCA-DA)
		Ranking-PCA
	Bayesian methods	Linear Discriminant Analysis (LDA)
		Diagonal LDA (DLDA)
	Methods based on projection to latent variables	Partial Least Squares Discriminant Analysis (PLS-DA)
		Orthogonal Partial Least Squares (OPLS)
	Classification trees	Classification and Regression Tree (CART)
		Random forests (RF)
	Machine learning	Support Vector Machines (SVM)

Table 2.2: **Examples of unsupervised and supervised statistical methods for proteomics and metabolomics data analysis (from Robotti et al. 2014)**

resulting in offset differences of intensity between the batches. Both biases may be corrected by using quality control samples (QC or pool) consisting of a mixture of all samples (Warwick B Dunn et al. 2011). As each feature is expected to be detected in all QC measurements, the median (Livera et al. 2015) or mean (Kloet et al. 2009) of these QC intensities may be used for inter-batch correction (S.-Y. Wang et al. 2013). The most popular method for the correction of the intra-batch intensity drift is the locally weighted scatter plot smoothing (LOESS) regression (Cleveland 1979; Warwick B Dunn et al. 2011). A review of alternative approaches is described in Kultima et al. 2009. When QCs are not available, strategies applying the regression to the samples, and/or using replicates in different batches, have been proposed (Rusilowicz et al. 2015). At the end of this step, features which still have high intensity variations in the QCs after the correction are usually discarded (e.g. when the relative standard deviation of their QCs is above 30%.).

Statistical methods for biomarker discovery

Many statistical methods have been applied to metabolomics data for biomarker discovery (Figure 2.2). On the one hand, univariate hypothesis testing with a correction for multiple tests, is used. On the other hand, multivariate statistics, which take into account the structured correlation between the variables, are applied to the data matrix (e.g., review from Robotti et al. 2014 about multivariate modeling in proteomics, which also holds for metabolomics). In particular, latent variable based approaches have been shown to perform well on spectral data: Principal Component Analysis (PCA) and Partial Least Squares regression (PLS; S. Wold et al. 2001) are popular methods for unsupervised and supervised multivariate modeling, respectively (see Worley and Powers 2013 for a review). Furthermore, the Orthogonal PLS method

(OPLS, Trygg and S. Wold 2002) enables to separately model the variations *orthogonal* to the response before building the PLS model (Pinto et al. 2013). Additional machine learning approaches have been used more recently in metabolomics, such as Support Vector Machine (Meyer et al. 2003), Random Forest (Breiman 2001) and Classification and Regression Trees (Frank and Lanteri 1989).

To be useful in the clinics, the list of biomarkers must be restricted to a few molecules. Selection of the most important features for the performance of the predictive models is therefore of critical importance.

At the end of the statistic step, the chemical structures of the selected metabolites need to be identified.

2.4.3 Identification

For each feature, the annotation which is extracted from the raw data during the preprocessing step is: the m/z value, the retention time (when chromatography has been used), the peak intensity, and, in some cases, the presence and intensity of isotopic or adduct peaks. Such information, however, is not sufficient to characterize the chemical structure. Identification therefore remains a major challenge for high-throughput biomarker discovery by metabolomics (M. R. Viant et al. 2017).

In fact, the Metabolomics Standards Initiative (MSI) has proposed four levels for metabolite identification, from an unknown compound merely detected as a peak on a spectrum (level 4) to the identified structure (level 1; (Sumner et al. 2007; Warwick B. Dunn et al. 2012)): the level 1 requires the matching of two or more orthogonal properties (e.g., retention time, m/z, MS/MS spectrum) between the candidate compound and the pure chemical standard. In practice, however, the requirement of a standard is difficult to fulfill since an MS acquisition potentially contains thousands of mass signals, and chemical standards are not available for most them. As a result, level 2 (putatively annotated) is often preferred: it requires the matching of one or two properties to spectra acquired with potentially distinct analytical conditions (e.g., spectra from external databases). One of these matching criteria is often the MS/MS spectrum, which gives an information about the structure of the molecule (i.e., its fragmentation pattern).

2.4.4 Tandem mass spectrometry (MS/MS) for structural elucidation

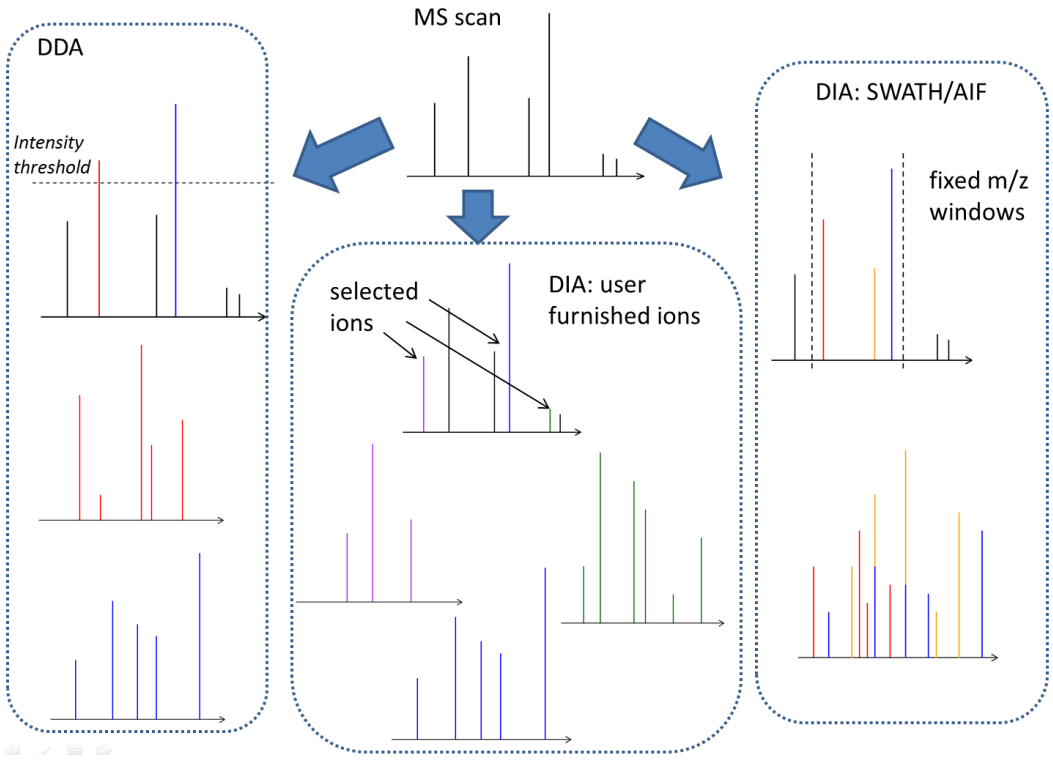


Figure 2.5: **Schematic view of Data Dependent (DDA) and Data Independent (DIA) acquisitions.** See the text for the details.

MS/MS spectra are obtained by the fragmentation of the (precursor) ions. Narrow m/z windows (isolation windows) are used to select the ion of interest between the first MS analyzer and the collision cell (although the size of the window is typically less than 1 Da, it is however not always sufficient to prevent the co-selection of isobaric ions). The nomenclature about tandem mass spectrometry (MS/MS, or MS2) that we will use hereafter is described in Section 1.3. MS/MS spectra are critical for the discovery of new metabolites (M. R. Viant et al. 2017) as they provide insights about the structure of the molecule. Moreover, the recent technological evolution of mass spectrometers (in particular the increase of the acquisition speed) has resulted in new applications (Fenaille et al. 2017): while MS/MS spectra were classically acquired on a restricted set of selected molecules (e.g. after a first pass of MS experiments), two new modes of acquisitions, Data Independent Acquisition (DIA) and Data Dependent Acquisition (DDA), have emerged in the recent years (Figure 2.5).

In DDA, the instrument automatically selects the precursor ions meeting a criterion defined by the user (e.g. an intensity threshold or a specific isotopic pattern; Heng-

Keang et al. 2008; Figure 2.5 left). This approach is popular because the set of generated MS/MS spectra covers a maximum of the detected molecules in the sample (Fenaille et al. 2017).

In DIA, the set of fragmented ions is independent from the data. Two types of DIA approaches have been described. In the first strategy, a specific set of metabolites (possibly identified in a previous run, or from a known biological pathway) is selected for fragmentation a priori (Figure 2.5, middle). This type of DIA is the most common one. It is used for the further characterization of putative biomarkers, or in studies targeting a specific set of molecules (e.g. from a given pathway). It is also the method of choice to analyze a known standard. Alternatively, in the second DIA approach, wide m/z windows covering the full m/z axis are defined, and the instrument fragments all the ions detected within these windows. Despite a few publications addressing the deconvolution of the generated spectra (Tsugawa, Cajka, et al. 2015; Nikolskiy et al. 2013), this method remains scarcely used in metabolomics because of the added complexity in the annotation process (Fenaille et al. 2017).

Both DDA and DIA approaches generate the same kind of data: a collection of MS/MS spectra associated to the m/z values of the precursor ions. With the recent advent of instruments with high acquisition speed, the amount of such MS/MS data has increased (hundreds of MS/MS acquisitions can now be performed within a single MS run; Benton et al. 2015), and the routine annotation of a large proportion of the metabolome seems achievable (M. R. Viant et al. 2017). To address this challenge, the development of innovative tools for MS/MS data analysis is pivotal (Section 9.4).

2.5 Current challenge of metabolomics

As stated above, the metabolome has a huge potential for biomarker discovery as metabolites are the end products of biological processes. This results, however, in a stronger influence of the environment on the metabolomic data compared to other omics. As a consequence, large cohort sizes are required to achieve sufficient statistical power. Nevertheless, the main metabolomics technology, LC-MS, is highly time-consuming (runs often exceed 30 minutes). New acquisition strategies are therefore needed (Zamboni et al. 2015). Direct introduction into the mass spectrometer (thus bypassing the chromatographic step) is an attractive approach for high-throughput metabolomics. As the time dimension is lost, high mass resolution is needed to discriminate the ions. Direct infusion (i.e. with a syringe) into very high-resolution and expensive instruments (FT-ICR) have been described (Andrew D Southam et al.

2017). Alternatively, Flow Injection Analysis (FIA) approaches using the same setting as LC-MS (except the chromatographic column) have been described initially with low resolution mass spectrometers(Beckmann, Parker, et al. 2008). With the emergence of high-resolution mass spectrometers (HRMS), the interest for FIA-HRMS in high-throughput metabolomics was renewed (Madalinski et al. 2008). Since there was no software tool for the processing of such data, the first part of the PhD was focused on the development and implementation of a preprocessing workflow for FIA-HRMS data (Part II).

Another major bottleneck in metabolomics is the chemical identification of the detected compounds. Tandem mass spectrometry is a powerful approach to obtain structural informations about the precursor (M. R. Viant et al. 2017). Direct matching with MS/MS spectra is, however, limited by the amount of available standards and the content of MS/MS databases. As we will see in Section 9.4, computational approaches for structure prediction have been developed in the recent years, based on *in silico* fragmentation. However, even top performing methods such as CSI:FingerID (Shen et al. 2014; Böcker and Dührkop 2016) still fail to identify known molecules in more than 20 % of the cases, and this rate of error is expected to be higher for unknown compounds. Alternatively, strategies mining the structural information within collections of MS/MS spectra are emerging. Similarities can then be used to propagate annotation of known molecules. Such approaches take advantage of the increasing number of MS/MS spectra which can be generated in a single acquisition. Furthermore, they do not require any information from spectral or molecular databases. The second part of the PhD aims at developing a new method to extract the similarities from a set of spectra using a graph representation (Part III).

Chapter 3

Datasets

Among the datasets which have been used for the validation of the algorithms throughout the PhD work, we describe two representative ones for each part of the thesis: **serFusion** and **serExactive** for the FIA-HRMS data processing (Part II), and **LemmDB** and **PenicilliumDIA** for the MS/MS structural mining (Part III). All data have been generated on HRMS Orbitrap instruments (Thermo Fisher Scientific) at the Drug Metabolism Research Laboratory (CEA LEMM, Saclay) and the Toxalim Research Center on Food Toxicology (INRA Toxalim, Toulouse).

3.1 FIA-HRMS datasets

The two datasets were generated to optimize the analytical conditions for the detection of compounds in serum by FIA-HRMS (Table 3.1). A commercial serum sample was used (Biopredic). In the **serFusion** dataset, the sample was spiked with increasing concentrations of a mixture from 40 compounds. These metabolites were selected from the chemical library of the LEMM partner. In the **serExactive** dataset, several dilutions of the serum sample alone were studied. The samples from the **serFusion** and **serExactive** were analyzed on the Fusion (resolution = 100,000 at $m/z = 200$) and Exactive (resolution = 50,000 at the same m/z) Orbitrap high-resolution mass spectrometers, respectively.

3.1.1 Biological material and sample preparation

Concentrations of the serum sample and the spiking mixture are detailed in Table 3.1. The list of the 40 spiked compounds is available in the supplementary information from the publication (Delabrière et al. 2017). Metabolites were recovered after methanol precipitation of proteins as described in Boudah et al. 2014.

3.2 MS/MS datasets

3.2.1 The PenicilliumDIA dataset

To study the secondary metabolism of *Penicillium Verrucosum*, a DIA approach was designed (Hautbergue et al. 2017). The majority of the detected compounds are unknown. We selected all spectra obtained with the CID mode at a collision energy of 40 eV, as this was the most represented type of collision among the dataset. The **PenicilliumDIA** dataset thus contains MS/MS spectra from 45 compounds (out of the 98 molecules described in the article). The peaks were detected using the MZmine local minimum search algorithm, which was the method providing the largest number of detected features with this dataset. MS/MS spectra were then extracted using the MS2process software presented in Section 9.9.

3.2.2 The LemmDB dataset

The **LemmDB** dataset corresponds to the set of standards (metabolites and drugs) used for identification by our LEMM partner (Roux et al. 2012). All 1021 compounds were analyzed in FIA-MS/MS on an Exactive mass spectrometer (Thermo Fisher Scientific), in HCD mode, at the following dissociation energies: $10, 20, 40, 80$ eV. For some compounds, no correct MS/MS spectra was obtained: a second set of energies was then used, depending on the structure of the compound.

As only the raw files were available, a preprocessing workflow was developed to match the selected MS/MS precursors and the MS detected features, and generate MS/MS spectra of high quality. This workflow relies on our proFIA (Part II) and MS2process (Part III; Section 9.9) packages. For the compounds with at least one spectrum containing more than 1 peak, a summed spectrum was computed. The

		serFusion	**serExactive**
Biological sample and spiked mixture	Serum dilution(s)	1/50	1/3, 1/6, 1/12, 1/24, 1/48, 1/96, 1/192, and 1/384
	Spiked mixture	0, 10, 33, 100, 333 and 1000 ng/mL	none
FIA-HRMS acquisition parameters	Instrument	Orbitrap Fusion	Orbitrap Exactive
	Spray voltage	3.5kV	4kV
	Ionization mode	positive	
	Capillary temperature	320	325
	Sheath gas flow	15	
	Auxiliary gas flow	8	12
	m/z range	85-1000	95-1000
	Mass resolution	500K at m/z 200	100K at m/z 200
	AGC target	10^5	3×10^6
	Ion Injection time(ms)	100	250
	Runtime duration	5	
	Mobile phase composition	75% isopropanol in water + 0.1 % formic acid	
	Flow rate (μL/min)	50	

Table 3.1: Experimental setup of the mass spectrometers used in the acquisitions

LemmDB dataset thus consists of 663 MS/MS spectra from the positive ionization mode. For the 368 remaining compounds, the MS/MS signal was either absent (no ionization, at least in the positive mode), or of low quality.

Part II

A preprocessing workflow for FIA-HRMS: proFIA

Chapter 4

Introduction

4.1 FIA-HRMS preprocessing algorithms and software tools

The goal of this part is to give an algorithm to pass from raw FIA-HRMS data to a data matrix *variables* × *samples* suitable for the statistical analysis. To do so we will start by reviewing the existing FIA-HRMS workflows. We will then see how the time dimension, while being reduced in FIA-HRMS data, still give some important information, which will be used in chapter 5 to model an EIC in FIA-HRMS. We will then give a review of the existing principle in peak-picking LC-MS data, which will be used in the proFIA workflow 6.

FIA-MS has been used in metabolomics for more than twenty years (Smedsgaard and Frisvad 1995). Initial FIA-MS experiments were used to obtain a global "fingerprint" of a sample (J. Allen et al. 2003; Beckmann, D. P. Enot, et al. 2007; Lloyd et al. 2011). The most frequent steps to obtain these fingerprints were:

- An FIA-MS acquisition was performed for each sample.
- The data were binned in the m/z dimension with a predetermined bin size, often 1.
- The intensities in each bins were summed leading to a single integer for each m/z bin.

Each raw acquisition is thus processed into a vector of intensities of fixed dimen-

sion (the number of bins): preprocessing is minimal, since no peak-picking nor peak alignment between samples is required. Subsequent statistical analysis based on such fingerprints (D. Enot et al. 2008), including machine learning (I. M. Scott et al. 2010), is straightforward. However the initial binning results in major drawbacks in terms of information loss. First, multiple isobaric compounds are grouped within the same feature. So, even if the statistical analysis of the sample is successful, the subsequent identification of the underlying compounds is complex. To perform such identification, authors have therefore often relied on prior information about the metabolites contained in the sample (Beckmann, D. P. Enot, et al. 2007; Ward et al. 2010), or on a supplementary acquisition using a higher resolution mass spectrometer (Favé et al. 2011). Second, removal of the baseline requires an additional acquisition from the solvent only.

With the development of high-resolution mass spectrometers (HRMS), it is now possible to assign a single molecular formula based on the mass measurement up to a few hundreds of Da (Kind and Fiehn 2006; Kind and Fiehn 2010). In a first attempt to pre-process such FIA-HRMS data, Yang and colleagues proposed a strategy starting with a nominal mass binning preprocessing step, followed by a subsequent step back into the high-resolution data to annotate the detected features(L. Yang et al. 2009). However, this type of approach has a major drawback: if the discriminating signal is weak compared to other signals in the same bins, this signal of interest will be masked. Alternative processing methods include a few proprietary software (Madalinski et al. 2008). The ability access to the source code, however, is mandatory to understand the underlying algorithms and develop complementary or alternative approaches. Recently, the following workflow was proposed by Fuhrer et al. 2011 and successfully reused in Sévin et al. 2016:

- All spectra from an acquisition file are summed to obtain a single total ion spectrum (TIS) in profile mode.
- A wavelet transform was used on the TIS to detect the m/z centroids.
- m/z centroids are matched between the different acquisitions by binning.
- Annotation was performed by matching the m/z features to a list of candidate molecular ions, adducts and isotopes generated from the KEGG database.

The success of the proposed approach, however, relies on the extensive metabolism annotation of the target organism, *Escherichia coli* to validate the m/z features. Such

a strategy would therefore fail with less characterized organisms, and make the detection of new metabolites impossible. Moreover the summing of all time point for each m/z prevent the discrimination between solvent and sample and therefore results in the incorporation of chemical noise. Finally, the Matlab scripts are not publicly available, and no detail is given about the quantification procedure.

Whereas no preprocessing pipeline for the FIA-HRMS data has emerged yet, a robust workflow for the the preprocessing of Direct Infusion (DIMS) acquisitions has been described recently (Andrew D Southam et al. 2017). The authors used the spectra-stitching procedure, which consists in acquiring mass spectra of multiple small overlapping m/z windows to increase sensitivity (Andrew D. Southam et al. 2007). However such a workflow cannot be applied to standard FIA-MS protocols, as it requires multiple technical replicates and acquisitions. Furthermore, in FIA, the additional time dimension and the presence a sample peak is not compatible with the last filter for DIMS data, which filter out the signal which vary too much along the time axis is incompatible with FIA data which present a peak.

In conclusion, no software package was available to process FIA-HRMS data. In addition, the described algorithms either relied on the comprehensive identification of the metabolites (Fuhrer et al. 2011), or on complex constraints on the acquisitions sequence (D. Enot et al. 2008; Andrew D Southam et al. 2017). As a result, there was an unmet need for bioinformatics methods and tools to pre-process FIA-HRMS data.

One of the initial expectations about FIA-HRMS data was that well-defined peaks would be observed on the EICs from sample compounds, (Figure 4.1a), and flat baselines would appear on signals originating from the solvent. Surprisingly, however, exploration of the raw data also highlighted unexpected peak shapes such as "reversed" peaks (Figure 4.1c) or double peaks (Figure 4.1b). Such peak shapes had already be found to be caused by matrix effect (Nanita 2013) in the case of well chosen but these dynamics have not been described at the scale of a full sample including hundreds of compounds. They could not be processed correctly by the workflow which sums the intensities over the whole EIC (D. Enot et al. 2008; Fuhrer et al. 2011). Therefore building an accurate a model of EICs became a priority (chapter 5).

Recently, S.Nanita (Nanita 2013) proved that information about the ME can be extracted from the EIC. These results suggested that the sample zone and ME components of the EIC could be modeled separately. As the previously described workflows do not consider EICs and compress the time dimension, they cannot extract this information. Therefore we chose to use the EICs, which allowed the extraction of information on matrix effect for the first time in an untargeted acquisition. The problem

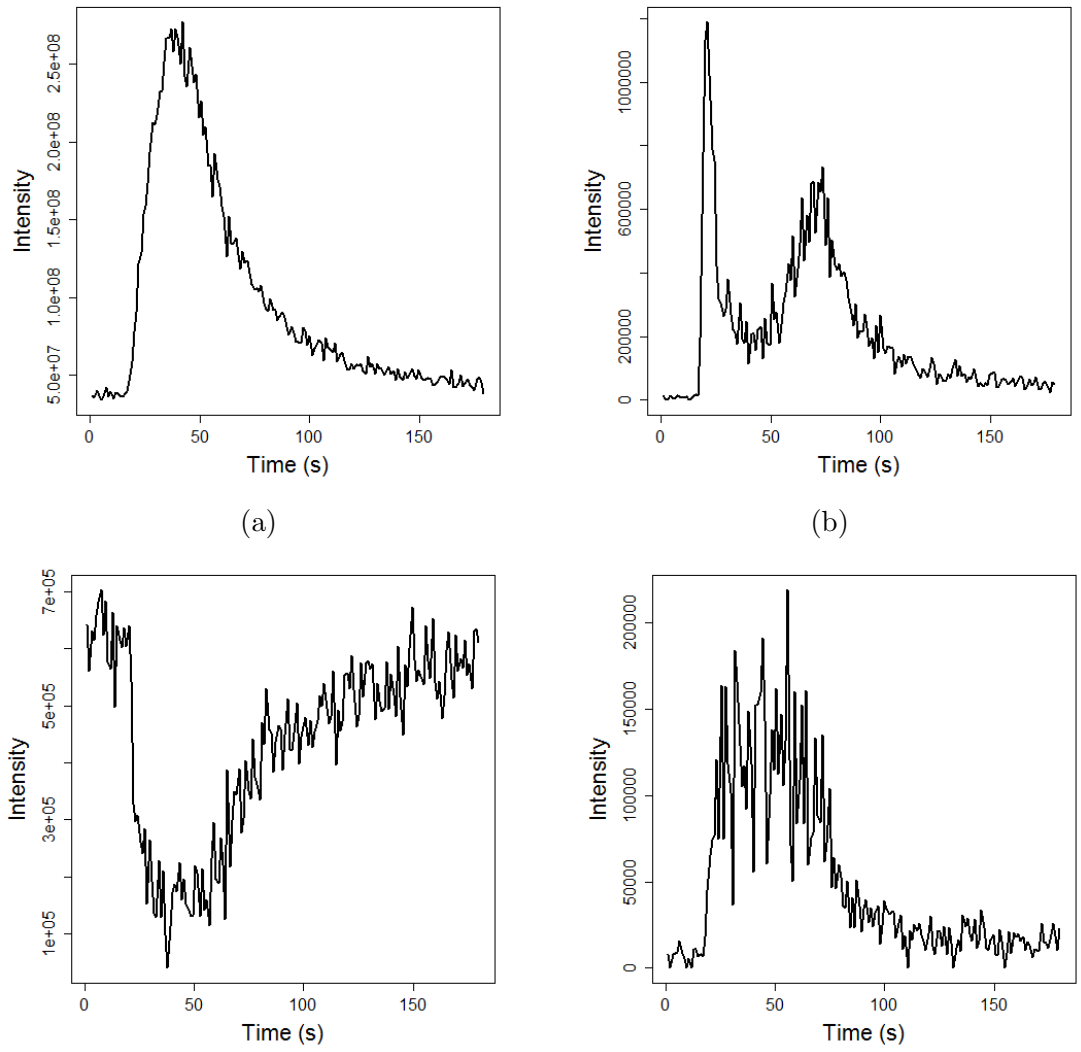
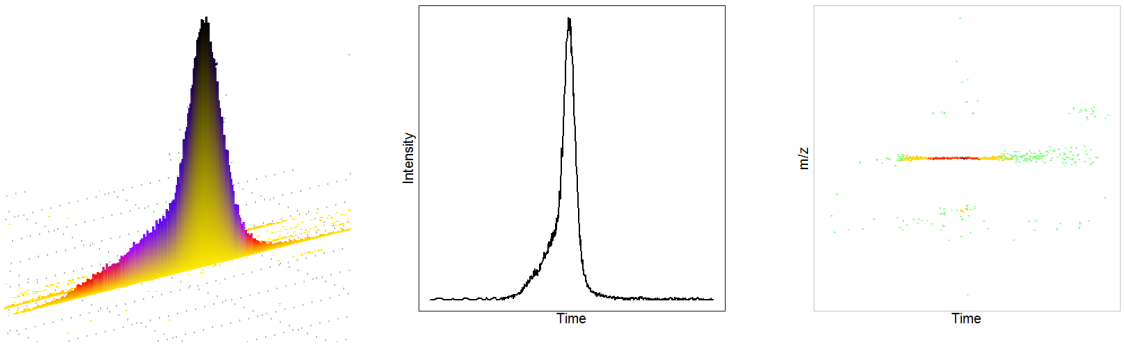


Figure 4.1: Influence of physical phenomena on the EIC signal (FIA-HRMS acquisition with an Orbitrap Fusion instrument): a) EIC with a clear sample peak, b) sample peak affected by matrix effect, c) solvent affected by matrix effect, d) combination of sample peak, solvent, matrix effect, and strong noise



(a) (b) (c)

Figure 4.2: A feature extracted from an LC-MS acquisition. a) shows the 3D peak, b) shows a projection along the m/z dimension and c) shows a projection into the time by m/z plane.

of preprocessing FIA-HRMS data can therefore be related to EIC processing of LC-MS data.

4.2 Peak detection methods in LC-MS

The preprocessing software which were selected for the review are freely available, with a clear documentation of the algorithms 4.1. A recent comparison of 3 of the most used softwares, mzMine, XCMS and MS-DIAL is available in Li et al. 2018. We discarded approaches which rely on additional data, such as the machine-learning method from MAVEN which requires pre-existing annotated data (Melamud et al. 2010), or the apLCMS strategy based on known compounds in the dataset (Yu and Jones 2014): while these methods have proven to be efficient, there was no such dataset available for FIA-HRMS data.

The first step of preprocessing algorithm is usually to reduce the complex 3-dimensional data to a lower number of features, often by summarizing a signal from an analyte by a single m/z, rt, and *intensity* (e.g., the area or the maximum of the peak) values. The second step is the removal of chemical noise, i.e., 1) the baseline generated by the molecules from the mobile phase and 2) the signals for which an accurate measurement of intensity is not possible.

While there is no official definition for the term "features" in LC-MS experiments, features are usually considered as points with similar m/z measurements in successive scans (figure 4.2c), with a peak clearly visible by projection of the data on the time by intensity plane 4.2b. If the data is in profile mode, a peak is also visible on the

Name	Article	Processing type	Data type	Implementation	Recently updated
XCMS-matchedFilter	C. Smith et al. 2006	sequential	Profiles	XCMS R package	Yes
XCMS-centWave	Tautenhahn et al. 2008	sequential	Centroids	XCMS R package	Yes
XCMS-MassifQuant	Conley et al. 2014	2D detection	Centroids	XCMS R package	Yes
MzMine-Baseline Cutoff	Katajamaa and Oresic 2005; Pluskal et al. 2010	sequential	Profile/Centroids	Java GUI	Yes
MzMine-Noise amplitude	Katajamaa and Oresic 2005; Pluskal et al. 2010	sequential	Profile/Centroids	Java GUI	Yes
MzMine-local minimum search	Katajamaa and Oresic 2005; Pluskal et al. 2010	sequential	Profile/Centroids	Java GUI	Yes
MzMine-savitzky-golay	Katajamaa and Oresic 2005; Pluskal et al. 2010	sequential	Profile/Centroids	Java GUI	Yes
MzMine-Wavelets	Du et al. 2006; Wee et al. 2008	sequential	Profile/Centroids	Java GUI	Yes
MzMine-GridMass	Treviño et al. 2014	2D detection	Profile/Centroids	Java GUI	Yes
OpenMS-FeatureFinderMetabo	Kenar et al. 2014	sequential	Profile/Centroids	C++ GUI	Yes
MS-DIAL	Tsugawa, Cajka, et al. 2015	sequential	Profile/Centroids	C++ GUI	Yes
apLCMS	Yu, Park, et al. 2009	sequential	Profile/Centroids	apLCMS R package	No
MAVEN	Melamud et al. 2010	sequential	Profile/Centroids	C++ GUI	No
MassTrace2	Tengstrand et al. 2014	sequential	Centroids	MATLAB GUI	No
Met-COFEA	W. Zhang et al. 2014	sequential	Centroids	C++ GUI	No

Table 4.1: List of reference software for LC-MS data preprocessing

m/z by intensity plane (see the *Description of MS data* section). Because of these characteristics, peak detection algorithms usually rely on a sequential approach (in 13 on 15 software from table 4.1), which include the following steps :

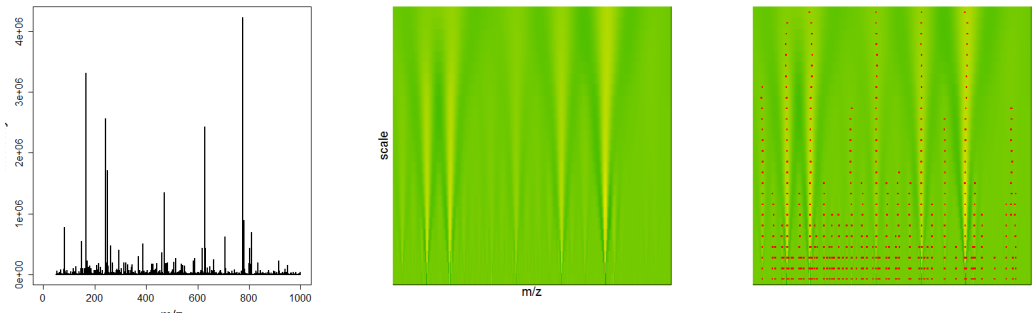
1. (Optional) Find peaks in the m/z dimension on each scan
2. Find points with similar m/z in consecutive scans and store them
3. Find peaks in the time dimension in the previously stored m/z traces

Although many approaches have been proposed for each step, they share a few underlying common principles. For example, two main strategies have been used for step 2: **mass traces** and **binning** methods. Regarding the third step, approaches based on either **peak modeling** or **extrema** detection have been proposed. The pros and cons of each method will be detailed. Finally, alternative strategies which have also proved successful for peak detection in LC-MS will be reviewed, including methods based on Kalman filter, or processing the m/z and rt dimensions simultaneously.

4.2.1 Peak detection in m/z dimension

The detection of the peaks in the m/z dimension, often referred to as *centroidization*, is probably the step for which the least different algorithms have been proposed. It is because this step is quite specific to the mass spectrometer technology. Fourier Transform based mass spectrometers may generate specific artifacts resulting from data transformation (Mathur and O'Connor 2009) which are not present in TOF data. Centroidization is therefore often provided by the constructor software, and sometimes even performed during the acquisition process. Two algorithms are available in MS-DIAL (Tsugawa, Cajka, et al. 2015) and MzMine (Katajamaa and Oresic 2005). The first one is based on a simple detection of local maxima, followed by a descent of the slopes on both sides of the peaks. The descent stops after the intensity decreases under a fixed threshold or when the intensity starts to increase. This basic approach is often the one implemented in constructor software.

The second type of algorithm for the detection of mass peaks is based on Continuous Wavelet Transform (CWT). It was initially developed for proteomics data (Lange et al. 2006) and was later modified to include additional noise estimates (Du et al. 2006; French et al. 2014) .



(a) (b) (c)

Figure 4.3: Continuous wavelet transform Du et al. 2006. a) shows a simulated spectra, b) shows the wavelets coefficients at each scale and c) shows the detected ridges corresponding to peak maxima

Continuous Wavelet transform (Daubechies 1992) are based on the convolution of a rescaled model wavelet and an input sequence, in the case of mass spectrum the input sequence is the raw mass spectrum and the model peak is often the Mexican Hat Wavelet. The convolved sequences are grouped into a matrix (Figure 4.3b), where rows correspond to different wavelet scales and columns to m/z values from the spectrum. P. Du et al. (Du et al. 2006) then proposed to identify the peaks by finding ridges (Figure 4.3c). French et al. 2014 developed a simpler algorithm by detecting maxima separated by more than 0.1 m/z. These wavelet methods rely on a local estimation of the noise, e.g., a percentile of the wavelet coefficients at the smaller scales. A more complete description of the underlying properties of wavelet methods in the case of LC-MS methods is available in section 4.2.3.

Centroidization results in spectra where each peak has been compressed to a single point in the m/z dimension (its centroid), often found by averaging all the masses in the detected peak and an intensity equal to his area or to the maximum of the peak.

4.2.2 Inter-scans m/z matching

This step aims to match peaks with similar m/z between different spectra (or scans). Three types of approaches have been described. The first one is mass binning on the m/z axis, as in the initial version of XCMS (C. Smith et al. 2006) for low resolution data, and in the MS-DIAL software for high resolution(Tsugawa, Cajka, et al. 2015; Tsugawa, Kanazawa, et al. 2014). To avoid that a feature would fall astride two bins, overlapping bins are used. A major drawback of binning is the loss of resolution in the m/z dimension. With the advent of high-resolution mass spectrometers, new approaches therefore emerged, which are based on the detection of mass traces

(Figure 4.4), such as in the centWave, MzMine, MAVEN, MET-COFEA and OpenMS software (Tautenhahn et al. 2008; Katajamaa and Oresic 2005; Melamud et al. 2010; W. Zhang et al. 2014; Kenar et al. 2014).

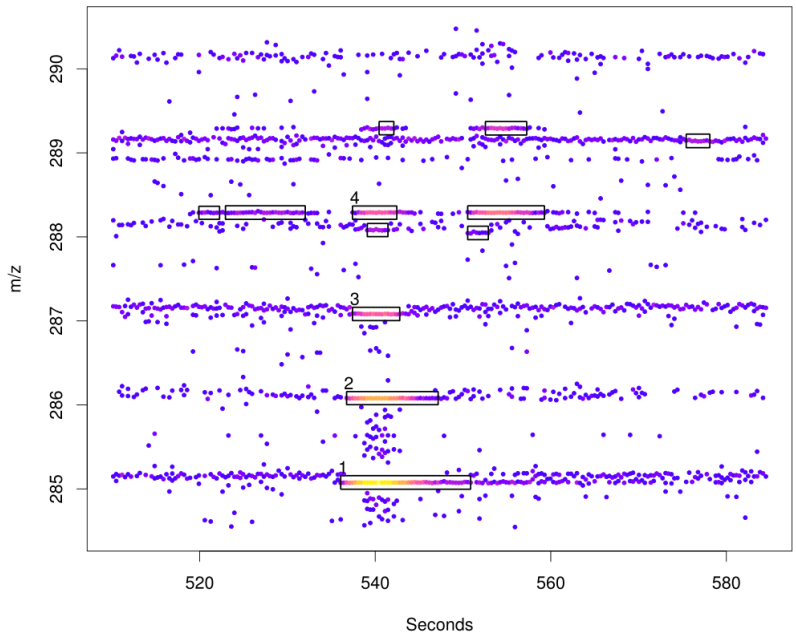


Figure 4.4: **An example of the detection of regions of interest (ROI) by the centWave algorithm from XCMS (Tautenhahn et al. 2008).**

In these algorithms, data points are grouped through consecutive scans: on the one hand, centWave, TracMass2, MET-COFEA and MAVEN process scans sequentially in increasing time order. On the other hand, OpenMS and MZmine (module Chromatogram builder) build m/z traces by extending the most intense point in both directions.

Another difference between the software is the m/z trace stopping criterion. All algorithms extend the mass traces by selecting points from the consecutive scan which have close m/z values. The m/z vicinity may be determined with a fixed threshold in ppm or Dalton (e.g., in centWave, MET-COFEA, and MAVEN). Alternatively, a refined strategy may be used: for instance, OpenMS uses an online variance and mean estimator for the mass trace to be extended. In case of multiple candidates, MET-COFEA takes into account both the intensity and the mass difference with the previous centroid to select the next point. In contrast, TracMass2 selects the only candidate with minimal m/z difference.

An additional stopping criterion is the continuity of the mass trace in the time dimension: OpenMS and MET-COFEA allow a fixed number of gaps, but not centWave

nor MAVEN.

While these principles are common to all algorithms, some additional steps may be included, such as the *prefilter* parameter in centWave which discards features below an intensity threshold, the m/z trace merging algorithm in MET-COFTEA, which try to correct mistakes by merging close mass traces.

In contrast to the above principles, an alternative approach has been implemented in apLCMS (Yu, Park, et al. 2009). First, m/z points are first grouped globally using a kernel density estimation. Second, within each m/z group, the segments in the time dimension capturing at least a fixed proportion of intensities are selected iteratively by decreasing length.

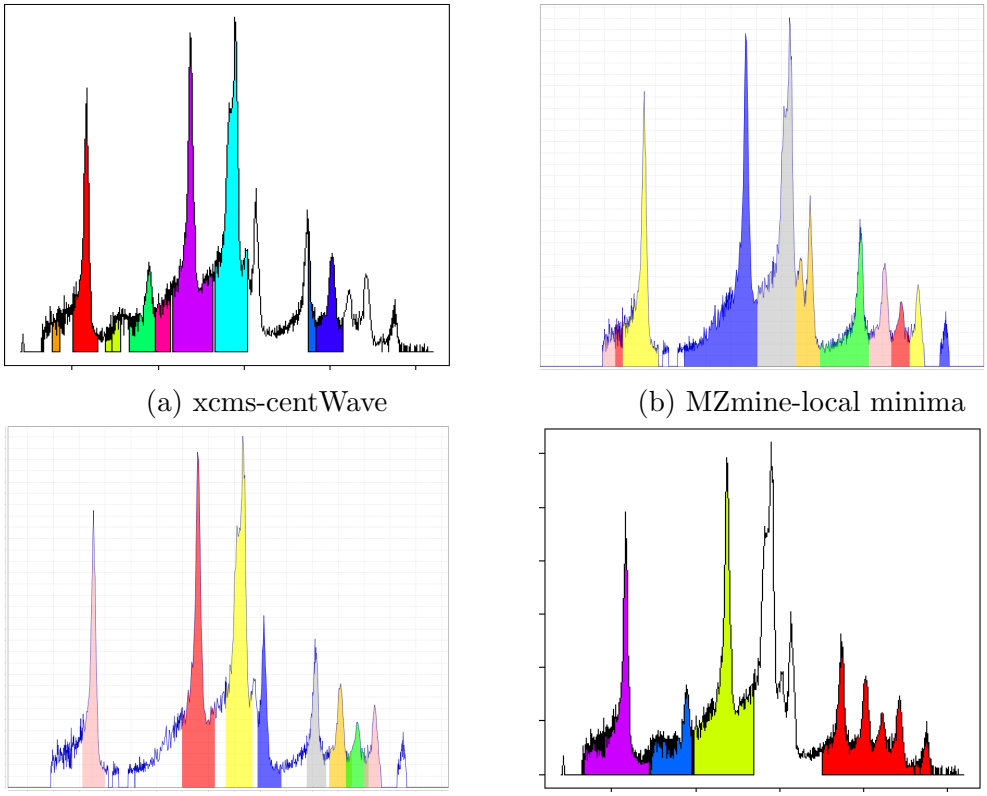
The output of this step is a set of m/z-rt cut of the data, each cut containing all the data points of a mass trace, or in the m/z dimension only if a binning strategy have been used. Although there may be a high number of such traces, the storage required for such lists is limited compared to the initial raw data (centWave ROI detection reduces an LC-MS dataset from 55.5 Mb to 17.7Mb). The next step aims at quantifying and separating the peak(s) contained in these m/z traces.

4.2.3 Peak picking on the EIC

At this stage, an EIC is extracted for each m/z trace or bin from the previous step. This EIC may include multiple co-eluting compounds, in addition to some chemical noise. To separate co-eluting compounds, a step of peak detection is therefore necessary. A comparison of the peak detection by reference processing software on such a complex EIC is shown in Figure 4.5. Multiple approaches have been proposed for peak detection. Currently, the most popular ones may be classified as **model** or **extrema** based methods. The former rely on the matching of a model to the signal using a convolution operator. In contrast, the latter first smooth the data before performing extrema detection directly on the smoothed sequence or on its derivative.

Model based method

Model based methods use the matching of a peak model to each points from the extracted m/z traces by convolution. They are currently implemented in many of the reference softwares, including both centWave and matchedFilter algorithms in XCMS, and both wavelets algorithms in MZmine and MET-COFEA. In this section we will



(a) xcms-centWave (b) MZmine-local minima

(c) MZmine-Savitsky-Golay (d) xcms-MassifQuant

Figure 4.5: **Peak picking on a complex EIC extracted from PenicilliumDIA**. Each color corresponds to a detected feature. The main strategies described in the text are illustrated: a) wavelet transform in the centWave algorithm (XCMS), b) simple local minima (MZmine), c) Savitsky-Golay smoothing with filter on the derivative values (MZmine), and d) Kalman filtration implemented in the MassifQuant algorithm (XCMS).

present the application of such techniques to EICs, which is similar to their application to the m/z dimension as described above in section 4.2.1.

The only matched pattern used in current LC-MS software is, to the best of our knowledge, the "Mexican hat" function, which is the second derivative of the Gaussian function (denoted P hereafter). This model is usually rescaled using a parameter provided by the user, as the scale of the model needs to match the width of the chromatographic peaks. This function has many interesting properties, such as a 0 area and symmetry, which are used by the algorithm. It is matched against the input signal using the convolution operator:

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau$$

Usually, the scans are supposed to be equally spaced in the time dimension (which is often the case in LC-MS experiments). This allows to compute a simple discrete convolution. We will denote the intensity of the n points of the EIC as $\mathcal{I}[n]$. The convolution then becomes:

$$(P * \mathcal{I})[n] = \sum_{m=-k}^{m=k} P[m]\mathcal{I}[n-m]$$

The k indicates that the model function is sampled on a relevant interval. Matched filtration has been extensively studied for chromatographic peaks (Bogaert et al. 1993; Danielsson et al. 2002; Andreev et al. 2003). One of the major results from these works is the proof of the noise reduction achieved with Gaussian (Andreev et al. 2003) or Mexican-hat functions (Danielsson et al. 2002). Another interesting finding is the application of matched filtration to different kinds of noisy data: (Bogaert et al. 1993) concluded that the inclusion of a complex noise model does not lead to significant improvement of the peak detection. This is of importance given the heteroscedasticity of the noise in LC-MS (see Figure 4.1). Matched filtration was also proved to be robust to errors on the filter width in (Bogaert et al. 1993), which is of crucial important as the peak width may vary in a single experiment.

The value of the symmetric and zero-area properties of the Gaussian is illustrated in Du et al. 2006: if we see the LC-MS chromatogram as an addition of a constant baseline ($\mathcal{C}$), a slow evolving baseline ($\mathcal{B}$), and the peak ($\mathcal{P}$), the convolution may be expressed, by using the distributivity property:

$$(P * \mathcal{I})[n] = \sum_{m=-k}^{m=k} P[m]\mathcal{C}[n-m] + \sum_{m=-k}^{m=k} P[m]\mathcal{B}[n-m] + \sum_{m=-k}^{m=k} P[m]\mathcal{P}[n-m]$$

Because P is symmetric and $\mathcal{C}$ is constant the first sum is 0. In addition, if we consider that $\mathcal{B}$ varies very slowly, the second term is close to 0. Therefore the convolution reflects only the matching between the peak model P and the real peak $\mathcal{P}$, with a limited contribution from the baseline. Matched filtration has also been shown robust across the width of the Gaussian (Danielsson et al. 2002).

Nevertheless, matched filtration faces some limitations in the case of co-eluting compounds with varying intensities, and is less robust to changing peak width in the time domain. As a result, a second group of methods based on the Continuous Wavelet Transform (CWT) emerged more recently (Daubechies 1992).

The use of CWT for MS data processing has been first reported in the m/z dimension for proteomics application (Lange et al. 2006; Du et al. 2006; Wee et al. 2008). The CWT may be described as the convolution of a signal with a family of filters which are rescaled versions of a function known as the mother-wavelet (denoted ϕ). The wavelet coefficients are more formally written as:

$$c(a,b) = \int_{\mathbb{R}} \mathcal{I}(t)\phi_{a,b}(t)dt$$

with $\phi_{a,b}(t) = \phi\left(\frac{t-b}{a}\right)$ where a is the scale and b is the translation. In metabolomics, ϕ is often the Mexican Hat wavelet. As ϕ is a symmetric, we have that $\phi_{a,b}(t) = \phi\left(\frac{b-t}{a}\right)$ and therefore by using the substitution property of integration, we can see that $c(a,b) = (\mathcal{I} * \phi_a)(b)$, with $\phi_a(t) = \phi\left(\frac{t}{a}\right)$. Therefore the CWT on a fixed scale is the matched filtration using the ϕ_a rescaled wavelet as the filter, and the pros of matched filtration apply to wavelet transform.

Du et al. 2006 showed that wavelet coefficients allow a more robust peak detection since each maximum should be detected at multiple scales: in fact, ridges of the wavelets coefficients are expected at peak locations (Figures 4.3b and 4.3c). This procedure allows a more robust and accurate detection, and is used in centWave (Figure 4.5a), MET-COEA, and MZmine wavelets implementations.

The scales range to be used in the algorithm are provided by the user, using his knowledge of the chromatographic system. The lower limits of the scale is especially important as it may result to the incorporation of high frequency noise if it is not stringent enough.

An additional filter is used to ensure that the peak is distinct from the baseline.

A signal over noise is often calculated, where the signal may be taken at the maximal value on intensity or on the wavelet coefficient. The noise is either computed locally, or by taking the mean of all or a portion of all the intensities of the EICs (Compared to the high number of points on an EIC, the points belonging to a chromatographic peaks are generally rare). This explains the poor performances of these methods in complex EICs (Figure 4.5), as there is not enough signal-free regions to estimate the noise correctly. These approaches are implemented in all the algorithms, and the interested reader is referred to the corresponding articles listed in table 4.1.

Maxima-based method

Such methods are based on the detection of peaks on the smoothed signal. The smoothing is an essential step as chromatograms may be very noisy. Comprehensive software toolkits such as MzMine or MS-DIAL provide multiple smoothing methods. An important challenge for these methods is that they should not distort the peak too much. Multiple smoothing methods have been used, including the well known Savitsky-Golay filter (Savtisky and Golay 1964), which is equivalent to the fitting of a polynomial by the least square procedure within a fixed time window. It has been implemented in the MS-DIAL and MZmine software (Figure 4.5c). Other methods which gained popularity recently are the Locally WEighted Scatterplot Smoothing (LOWESS) (Cleveland 1979), based on the weighted fitting of a polynomial (often of degree two or three) locally (i.e., in small windows). LOWESS is implemented in the OpenMS algorithms. Finally, the Kernel Smoother has also been used (often using a gaussian kernel), e.g., in MAVEN, apLCMS, and TracMass2.

The simplest procedure is to detect local maxima on the smoothed sequence and to perform slope descent on each side, the first reached minimum defining the limits of the peaks (Figure 4.5b). This is the procedure used by the OpenMS and the MAVEN software, and by apLCMS to obtain a first peak estimate. Peaks may also be detected by using the derivative (Figure 4.5c), such as in MZmine (Savitsky-Golay module) or in MS-DIAL. After this step, the software often refine their estimates by going back to the non-smoothed data.

The filters used to remove noisy peaks are different between all softwares, but often include some local noise estimation. MZmine uses a minimum relative peak height and a ratio of peak maximum to edge length, and checks that the peak is intense enough in a small window. MS-DIAL filters are based on the amplitude change of the derivatives and of the intensities. Three noise thresholds are used in MS-DIAL, one

on the difference of the smoothed sequence, one of the first-derivative sequence, and one on the second derivative sequence, the derivative sequences being calculated using 5-point Taylor approximation. The noise on each sequence is calculated as the mean of the points below 5% of the maximum on each sequence. The left edge of a peak is detected when both the first derivative and the intensity on the smoothed sequence exceed the thresholds. The top of the peak is recognized when the first-derivative change of signs and the second derivative pass the associated threshold. Right edge is detected similarly to left edge.

These methods often perform better than model-based methods on complex EICs, as it is shown in Figure 4.5. However they are generally less sensitive than wavelets to low intensity signals as smoothing becomes less reliable.

4.2.4 Alternative methods

This section presents three algorithms, TracMass, gridMass, and apLCMS, which are based on different approaches.

Kalman tracking based methods

Historically, Kalman filters have been used in multiple area, including in object tracking along multiple measurements (Cuevas et al. 2005). Kalman derived the equations to track an object moving with a constant velocity given random fluctuation between measurements (Kalman 1960): more precisely, a first set of equations provides the prediction for the next time point, as well as a confidence interval, and a second set updates the estimation. The reader interested in the Kalman theory and its application to MS processing is referred to Kalman 1960 and Aberg et al. 2009, respectively. Kalman tracking was initially applied to LC-MS in the TracMass software (Aberg et al. 2009) and reimplemented in the MassifQuant algorithm integrated into XCMS (Conley et al. 2014) (Figure 4.5d). The latter was shown to outperform centWave in terms of quantification, but was more sensitive to noisy data (Conley et al. 2014). It is interesting to note, however, that the second version TracMass2 did not use a Kalman filter but a more classical m/z trace detection approach, the authors stating that the results were similar with a lower level of complexity (Tengstrand et al. 2014).

gridMass algorithm

gridMass (Treviño et al. 2014) is the only algorithm for LC-MS processing which processes the data in the m/z and time dimensions simultaneously. The gridMass algorithm work by generating a grid of equally spaced probes on the m/z-rt plane. Probes are then sequentially moved to the highest intensity point in a small rectangle around it. A feature is then detected when multiple probes converge to this maximum, the initial position of the probes determining the limits of the peak in both dimensions. The initial set of features is then cleaned to remove artifacts caused by the baseline. This principles however does not use the inherent sparsity of the data in both dimensions which reduce the processing time.

apLCMS software

apLCMS was first described in Yu, Park, et al. 2009 and further extended in Yu and H. Peng 2010. While apLCMS processing starts with a sequential peak detection using a "maxima" based method, the workflow also contains a deconvolution step which is not present in the other software: a bi-Gaussian mixture model if fitted to accurately model asymmetric peaks, by using a modified Expectation-Maximization algorithm (Yu and H. Peng 2010).

4.2.5 Grouping peaks

The presented algorithms give a good overview of the principles from the most popular peak-picking methods in LC-MS. Some of these methods were used to build the proFIA algorithm, as described in chapter 6. After this peak detection in individual sample files, the peaks need to be matched across the different acquisitions. Because the methods for peak alignment described in the literature often rely on both m/z and rt matching, they could not be directly transposed to FIA-MS data. Moreover they often rely on the fact that the distribution of the m/z values on an individual spectrum is sparse (because of the chromatography step upstream from the MS instrument). Therefore the matching in the m/z dimension is usually performed by using a simple window (e.g., in MAVEN and XCMS). A more advanced algorithm in apLCMS uses kernel smoothing followed by peak detection. More complex strategies include the RANSAC aligner in MZmine (Pluskal et al. 2010), which uses a model of m/z and rt deviations built on multiple subsamples, or the approach from MET-COFEA which includes a hierarchical clustering based on both a proximity metric and peak shape

correlation. In contrast to LC-MS, we will see in chapter 6 that the m/z axis in FIA-MS is very dense and that a dedicated method for peak matching is required.

In conclusion, due to the lack of a software which would rely on the specific properties of FIA signal, the development of a new workflow was necessary. Due to the complexity of the observed peak shapes, a dedicated model was required, as detailed in chapter 5. This model was then used to design a workflow for the preprocessing of FIA-HRMS, proFIA (chapter 6). Finally the performance of proFIA was evaluated on several datasets from distinct HRMS instruments and applications (chapter 7).

Chapter 5

A computable model for an Extracted Ion Chromatogram in FIA-MS

The goal of this section is to define a general model for the Extracted Ion Chromatogram (EIC) profiles generated by Flow Injection Analysis coupled to High-Resolution Mass Spectrometry (FIA-HRMS) technologies, as independent as possible of the specific instrumental setup used. In particular, for such a model to allow efficient and robust preprocessing, the following criteria must be fulfilled:

1. As few physical quantities from the FIA-HRMS system as possible should be required by the model
2. The model should be computable and interpretable
3. The model should explain, at least qualitatively, the various types of EIC profiles, such as those shown on Figure 4.1.

To develop the EIC model, we begin in section 5.1 by extracting the major physical phenomena affecting an EIC in FIA-HRMS, based on a review of the literature and on the observation of real data sets from distinct instruments. Section 5.2 then aims to choose a computable model for each of the selected phenomena. Finally, section 5.3 presents our complete EIC model for preprocessing, and shows how the proposed model successfully explains the signals observed on Figure 4.1.

5.1 Extraction of physical phenomena affecting an EIC in FIA-MS

Flow-Injection Analysis (FIA) was defined in 1988 by Ruzicka and Hansen 1988 as "information-gathering from a concentration gradient formed from an injected, well-defined zone of fluid, dispersed into a continuous unsegmented stream of carrier". The reader interested in a more detailed introduction is invited to take a look at the website from J. Ruzicka. The part of the system performing the injection of the sample and including the carrier flow will be called the FIA system in this thesis. It determines the shape of the signal in the time domain (chromatogram): the chromatogram of a specific ion (i.e., within a specific m/z range) is classically referred to as the "Extracted Ion Chromatogram" (EIC). Some examples of EICs are shown on Figure 4.1. In the case of FIA-MS, the information-gathering part is provided by the mass spectrometer. Mass spectrometer instruments include three main components, the ion-source, the mass analyzer and the detector. Each of them has an influence on the resulting signal. In this section we will give an overview of the main physical phenomena from the FIA system, the ion-source, and the analyzer, which impact on the EIC profile. However we will not discuss the underlying physics theory, as this is out of the scope of this PhD.

5.1.1 Physical phenomena originating from the Flow Injection Analysis (FIA) system

Because all the physical phenomena affecting an EIC in FIA-MS are not well-known (Kolev 2008), and the full modeling of an FIA-system is out of the scope of this thesis, we will consider the influence of the FIA system on the EIC shape as depicted in Figure 5.1. It is a reasonable approximation of a simple FIA system used for the coupling with a mass spectrometer (John Draper and Beckmann 2013).

In standard analysis, solvents are chosen to avoid reaction with the sample: therefore, we will not take these interactions into account. According to Kolev 2008, the phenomena responsible for the mass transfer of the sample zone in the carrier flow (and hence the distortion of the sample zone) are convection and diffusion. Convection occurs in the axial direction, and is caused by the difference of flow speed between the middle and the walls of the tube (Figure 5.1a). However convection is partially compensated by diffusion, a radial transfer of mass between the flow at the center of

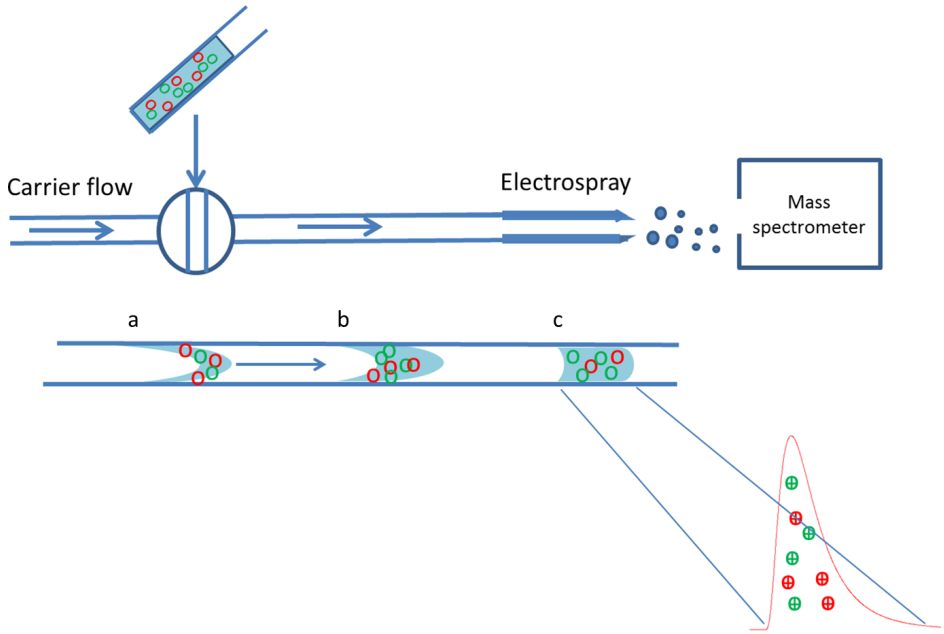


Figure 5.1: Schematic view of an FIA system. The injected sample forms a zone (a), which is subsequently dispersed within the carrier flow (b-c).

the tube and the tube wall, and reciprocally. The diffusion homogenizes the parabolic sample zone, resulting in a more symmetrical volume (Figure 5.1b-c). Both convection and diffusion vary according to the apparatus, and therefore their influence on the peak shape may differ between FIA systems.

This phenomenon results in a gradient of concentration which will pass at consecutive times through the electrospray to be ionized. This gradient of concentration will be referred in the next section as concentration curve.

5.1.2 Phenomena occurring in the Electrospray Ionisation Source (ESI)

The electrospray ionization source (ESI) is most used in metabolomics because of its ability to ionize polar and semi-polar molecules (Lei et al. 2011). It has been extensively used in FIA (J. Draper et al. 2013). A typical ESI source is described in Figure 5.2. The sample is injected through a charged capillary and surrounded by a counter electrode. This leads to the formation of liquid-phase ions (Kebarle and Verkerk 2009). The electrolytic liquid is sprayed in micro-meter sized droplets using an aerosol. The droplets then shrink in size while the solvent evaporates, and, after passing the Rayleigh limit of stability (F.R.S. 1882), they disaggregate because

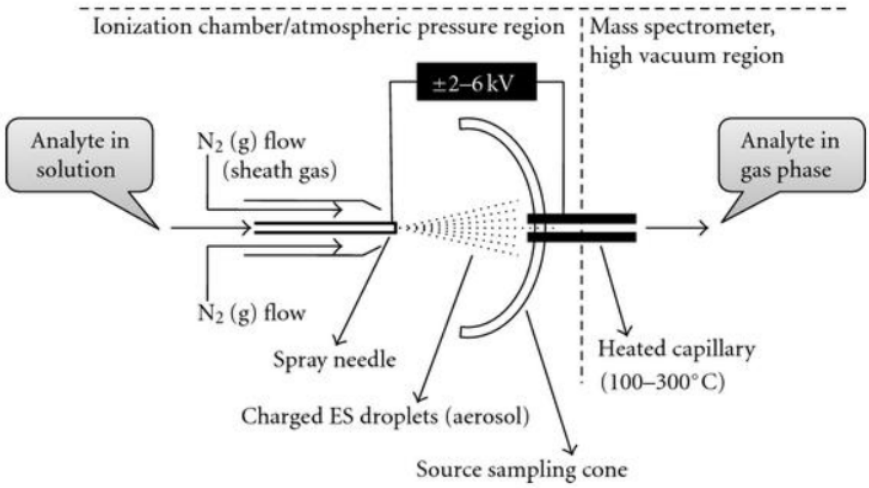


Figure 3: A schematic representation of the ESI-ion source.

Figure 5.2: **Overview of an electrospray**, extracted from Banerjee and Mazumdar 2012

of electric strength, until only ions remain in the gas. ESI usually produces single-charged ions. The sign of the charge depends on the voltage between the electrodes. ESI is considered as a "soft" ionization technique resulting in minimal fragmentation of the molecules.

The transmission from the liquid-phase to the gas-phase highly depends on the parameters of the ESI source, such as the current in the electrode, the distance between the tip and the capillary, and between the capillary and the skimmer (Page et al. 2007). Furthermore, the efficiency of the desorption/ionization process for an analyte is influenced by its physico-chemical environment, and in particular by the surrounding analytes.

The **matrix effect (ME)** has been defined as *"the combined effect of all components of the sample other than the analyte on the measurement of the quantity"* (Guilbault and Hjelm 1989). The exact causes of ME are not known, and are complex to investigate because electrospray includes gas-phase and liquid-phase reactions simultaneously. Trufelli et al. 2011 listed four possibles explanations:

- Competition between analytes for the charge and the access to the droplet surface (Kebarle and Verkerk 2009; Cech and Enke 2001).
- Presence of interfering molecules, which may increase the viscosity and surface tension of the droplets, thus affecting their formation and resulting in a loss of

charge reaching the mass analyzer (King et al. 2000).

- Non-volatile additive which may result in the formation of solid precipitates (King et al. 2000).
- Presence of ions in the matrix or the additives which will pair with ions naturally present in the sample (Holcapek et al. 2004).

Because in FIA all analytes enter the ESI source simultaneously (contrary to hyphenated techniques relying on chromatography), ME strongly affects the signal. The modeling of ME was therefore a critical step within our preprocessing workflow.

The ions produced in the ESI source then enter the mass spectrometer.

5.1.3 Phenomena occurring in a Mass Spectrometer (MS)

Orbitrap and Time-Of-Flight (TOF) are the two types of mass spectrometers which are most used for high-resolution metabolomics (despite its very high resolution, Fourier Transform Ion Cyclotron is less frequent because of its cost, J. H. Gross 2011). They are based on completely different principles, which make a general description of the physical principles of mass spectrometer detection impossible. We will give a short description of the principles of each type of mass analyzer, based on the book from J. Gröss (J. H. Gross 2011).

Time-of-flight mass analyzer

The first functional TOF instrument was presented in 1948 by A. E. Cameron and D. F. Eggers Jr. (Cameron and Jr. 1948). In TOF instruments, ions are accelerated in an electric field before entering a field-free region in which they drift at different speeds according to their kinetic energies. Their velocities follow a statistical distribution depending on their mass over charge (m/z). A major source of noise in this set-up is called the dead-time effect (Gedcke 2001; Titzmann et al. 2010). When too many ions of similar m/z hit the same detector simultaneously, some events are ignored because of detector saturation, resulting is a bias in the measured spectra. To reduce the statistical noise generated by such detection methods, the output spectrum is usually a combination of multiple acquisitions.

Orbitrap mass analyzer

Ion trapping in quadro-logarithmic field (orbitrap) to produce mass spectra was demonstrated in the late 90s by A. Makarov (Makarov 2005). In an orbitrap mass analyzer, ions are injected around an electrode surrounded by another electrode. An electrostatic voltage of several kilovolts, negative for positive ions, is applied to the central electrode, while the outer electrode is at ground potential. Ions are injected around the central electrodes and start to loop around the central electrode. Their looping frequencies are directly linked to the mass-on-charge ratio of the ions, and can be measured using the generated electrostatic field.

Both of these spectrometer, are source of noise for the mass spectrometers, which is caused by the process of measurement and by the stochastic nature of the physical process. This noise is notably caused by the Flicker noise present in all electronic currents and by the counting of ions in TOF mass spectrometer. We just denoted by observation that the noise included an heteroscedastic component, his existence have been notably described in Anderle et al. 2004 and Wentzell and Tarasuk 2014. Both modeled the noise variance on TOF data as a second order polynom.

In conclusion, we selected three main physical phenomena for our EIC model:

- The influence of convection and diffusion on the sample zone.
- The influence of matrix effect on the compound intensities (ESI source).
- The noise added by the the mass spectrometer.

We will now define computable expressions for each of these components.

5.2 Selection of a computable model for the extracted physical phenomena

Our aim in this section is to define a mathematical expression of the physical phenomena affecting an EIC, which will then be used to obtain a computable model of the intensities of the EIC as a function of time. This is of crucial importance because the goal of an FIA-HRMS acquisition is in the general case to obtain quantitative or qualitative information about the analytes in the sample, while at the same time

discarding the chemical background or noise present in the solvent (Trotzmuller et al. 2010). However without a clear model of the comportment of the ions originating from the sample, this discrimination is impossible. We consider an ideal FIA system in which there is no retention of the analytes. The validity of this hypothesis will be discussed in the next chapter. We also consider that there is no chemical reaction occurring between the sample and the carrier flow, and between the individual analytes within the carrier flow (such an hypothesis is common in chromatography, and is based on the low concentration of the analytes).

5.2.1 Modeling of the concentration curve

We stated that the sample zone has an initial parabolic shape because of convection, and then homogenizes because of diffusion. This evolution has been described as the convection diffusion by Levenspiel and Bischoff (Levenspiel and Bischoff 1964). However because of the complexity of this equation, no analytical solution has been provided yet, to the best of our knowledge (Kolev 2008).

A review by Kolev 1995 lists different types of models for FIA-MS data. These models are referred either as black-boxes, (when considering the system in terms of input-output only), or deterministic and probabilistic (i.e., based on the physical properties of the system). Deterministic models require an accurate knowledge of the experimental setup and of the physical phenomena dominating the system's dynamics. Such models are therefore not compatible with our constraints of robustness and generalization. Probabilistic models suffer from the same drawbacks, the required parameters often including the mean carrier flow velocity, diffusion coefficient, or tube diameter (Wentzell, Bowdridge, et al. 1992; Kucza 2013). We therefore selected a black-box approach.

Black-box models consider the system only in terms of inputs and outputs (Kolev 1995), without relying on the underlying physical process. While Kolev listed a lot of models used to cover the majority of FIA systems, most of them are not relevant for our problem, as our goal is not to predict the response of the system from previous data (Margaret and Dermot 1993), but instead to obtain a computable model which may be fitted to an EIC. We therefore focused on functions known as "empirical peak shape models", which have been widely used in chromatography(Felinger 1998) and flow injection approaches(Kolev 1995). The majority of these peak models are derived from the Gaussian function as this function traces back to the first linear model of chromatography by Martin and Synge (Martin and Synge 1941), the interested readers

is referred to Felinger 1998. However the Gaussian is symmetric: although concentration curve are supposed to be a gaussian function if the time of residence is long enough (Kolev 2008), this condition is not reached in practice with the majority of FIA experiments, as it would lengthen the acquisition. Therefore we focused on asymmetric empirical functions instead. We chose to discard piecewise-defined functions such as the bi-Gaussian (Eli et al. 1970) as they double the numbers of parameters, and rely on the assumption that the left and right slopes of the peak are independent.

Exponentially Modified Gaussian (EMG) The EMG function has been successfully used to describe concentration curves (B. F. Johnson et al. 1992; Brooks and Dorsey 1990). It is the convolution between a Gaussian with parameters μ and σ and an Exponential with decay rate τ.

$$c(t)_{\mu,\sigma,\tau} = G_{\mu,\sigma}(t) * E_{\tau}(t) = \frac{1}{\tau\sqrt{2\pi\sigma^2}} \int_{x=0}^{t} e^{--\frac{(x-\mu)^2}{2\sigma^2}} \times e^{-\frac{t-x}{\tau}} dx$$

With only three parameters, these functions can efficiently model a wide range of asymmetric peak shapes (especially in FIA systems). Moreover many techniques allow fast calculation (Berthod 1991) and determination of parameters and statistical moments from simple measurements on peak shape (Mark and Joe 1992; Brooks and Dorsey 1990; Foley 1987). These methods may therefore be used to give good starting points for curve fitting.

Gamma distribution: Another asymmetric curve successfully used in FIA is the Gamma distribution (Smit and Scheeren 1988). However it requires the addition of a shift parameter and has therefore as many parameters as the EMG. As it has been less studied and less used in the case of concentration curves a single time, we chose to model the concentration curve as an EMG function over the gamma distribution.

5.2.2 Modeling of matrix effect (ME)

The mathematical expression of matrix effect (ME) should remain simple and computable even for a thousand of molecules measured simultaneously in a complex biological sample.

While the presence of ME has been known for quite a long time (Kebarle and Tang 1993) and multiple putative sources of ME have been suggested, no comprehensive

ME model has been proposed yet because of the multiplicity and of the complexity of the phenomena involved. However two models of the molecular competition for the ionization have been described (Kebarle and Tang 1993; Enke 1997). In this section we show the main incompatibility of these model on real experimental data including hundreds of features.

Model proposed by C.G. Enke (Enke 1997)

C. G. Enke proposed a model where charge is partitioned between the surface and the inside of the droplets. While this model correctly describes different known properties of the response of ESI-MS systems (e.g., linearity of the response at low concentration, loss of linearity at high concentration), the observed signal for a molecule depends on the concentrations of all other analytes. Therefore this model cannot be used in practical situations where the concentrations of analytes are unknown.

Model proposed by P. Kebarle and L. Tang(Kebarle and Tang 1993)

Original model Kebarle and Tang made the assumption that ionization is limited by one main factor, the finite amount of charges which end up at the surface of the droplets. Each compound gets a limited amount of this intensity. They proposed the following formula for the intensity of an analyte A co-eluting with an analyte B (and an electrolyte E):

$$I_{A^+} = pf \frac{k_A C_A}{k_A C_A + k_B C_B + k_E C_E} I \tag{5.1}$$

where I is the total intensity provided by the electrospray; k_i is an analyte specific factor representing the ability of the respective ion species to become part of the charge on the droplet surface and its ability to subsequently escape and enter the gas phase; f is the fraction of charges on the droplets which are converted to gas-phase ion; p is the fraction of the ion detected in the mass spectrometer relative to their gas-phase ion.

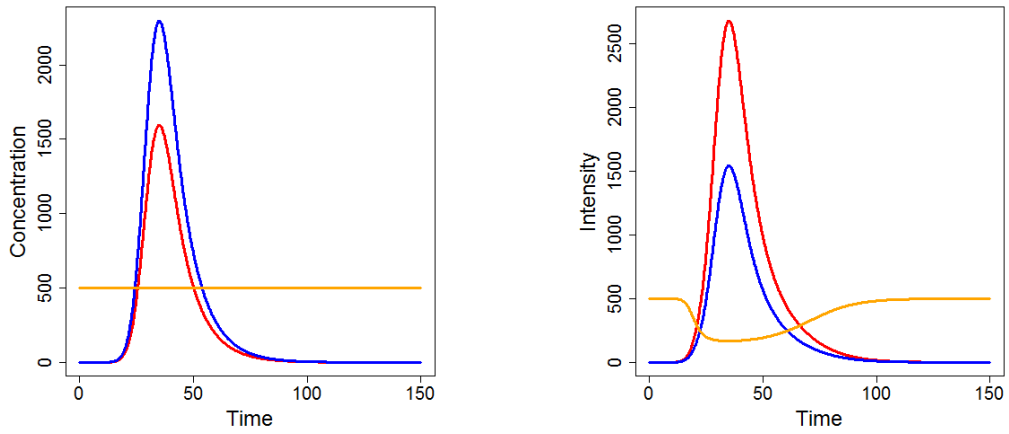
Extensions to multiple compounds It is possible to extend the model to all analytes in the solution, O:

$$I_{A^+} = pf \frac{k_A C_A}{\sum_{i \in O} k_i C_i} I \tag{5.2}$$

Extension to FIA Since there is no chromatographic separation in FIA, all compounds with the same mass (e.g., all isomers) contribute to the same m/z signal. For a given m/z, the observed intensity on a sufficiently small EIC is therefore $\sum_{j \in A_m} I_{j^+}$ (where A_m is the set of all analytes with the same m/z). The Formula (5.2) then becomes:

$$\sum_{j \in A_m} I_{j^+} = pf \frac{\sum_{j \in A_m} k_j C_j}{\sum_{i \in O} k_i C_i} \tag{5.3}$$

Note that in this model, the denominator remains unaffected by the summing. To make the notation easier, I_{M^+} will hereafter denote the sum of intensities observed at a certain m/z. An example of the application of this model to simulated FIA curves is shown in Figure 5.3.



(a) Concentration curve

(b) Intensity affected by suppression

Figure 5.3: Kebarle and Tang model applied to two analytes (blue, $k_A = 2$ and red $k_B = 5$), and a solvent compound (orange, $k_E = 1$)

Model effect on solvent We can prove that I is proportional to the Total Ion Current (TIC) of a scan. If we sum the Formula (5.2) for all analytes we get :

$$TIC = \sum_{j \in O} I_{j^+} = pf \frac{\sum_{i \in O} k_i C_i}{\sum_{i \in O} k_i C_i} I = pfI \tag{5.4}$$

so $I = \frac{TIC}{pf}$. In an FIA acquisition, the analytes either come from the solvent, or from the sample. Compounds from the solvent can be assumed to have a constant concentration during the acquisition. In contrast, compounds from the sample have a peak shape which depends on the flow rate, and on the tube and solvent properties

(Figure 5.3a).
These two different chromatographic profiles allow us to rewrite the Formula (5.2) :

$$I_{M^+} = pf \frac{\sum_{j \in A_m} k_j C_j}{\sum_{i \in O_{sol}} k_i C_i + \sum_{i \in O_{sam}} k_i C_i} I \tag{5.5}$$

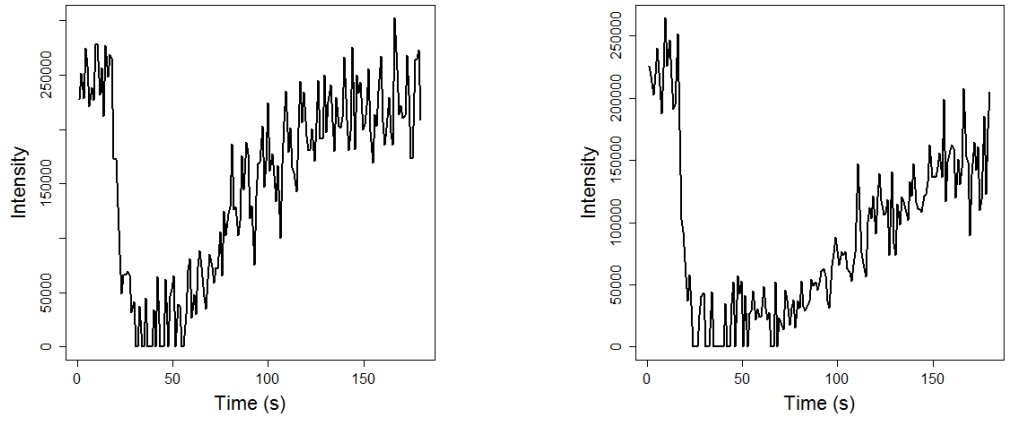
with O_{sam} for the analytes from the sample and O_{sol} for the molecules from the solvent. As there are multiple scans in an FIA acquisition, we can specify the time dimension, and replace I by $\frac{TIC}{pf}$:

$$I_{M^+}(t) = \frac{\sum_{j \in A_m} k_j C_j(t)}{\sum_{i \in O_{sol}} k_i C_i(t) + \sum_{i \in O_{sam}} k_i C_i(t)} TIC(t) \tag{5.6}$$

We can use the Equation (5.6) to highlight an interesting property of this model for the signal from the solvent. If we suppose that the observed m/z comes from solvent ions only (which we denote S), this means that their concentration curve is constant: $\sum_{j \in A_m} k_j C_j(t) = C_S$. Therefore Equation (5.6) becomes:

$$I_{S^+}(t) = \frac{C_s}{\sum_{i \in O_{sol}} k_i C_i(t) + \sum_{i \in O_{sam}} k_i C_i(t)} TIC(t) + \tag{5.7}$$

In this equation we can observe that none of the terms depend on S, except the C_S. This means that in the model proposed by Kebarle and Tang, EICs coming form solvent features are proportional. This property, however, was not observed in experimental data as shown in Figure 5.4.



(a) (b)

Figure 5.4: **EICS composed of signal originating from the solvent with different shapes (serFusion):** It is clearly visible that figure a) may not be a rescaled version of 5.4b) because 5.4a) come back close to his initial level, while 5.4b) does not.

In conclusion, as the existing models were either impossible to use without the knowledge of the real concentration profiles, or do not agree with experimental data, we switched to an empirical approach. Such an approach has been described by Nanita (Nanita 2013), who compared the EIC from a pure standard (i.e., without ME) to the EIC from the same standard spiked into a biological matrix. Using this technique, ME was expressed as a function of the normalized intensity (S_{norm}) using a first- or second-order exponential $ME(S_{norm}) = ae^{bS_{norm}}$ or $ME(S_{norm}) = \frac{ae^{bS_{norm}} + ce^{dS_{norm}}}{a + c}$. While Nanita showed that the second-order exponential better fits the data, we selected the first-order model as it reduces the number of parameters with only a small impact on the fit quality. Moreover to ensure that $ME(0) = 0$ we used the expression $a(e^{bS_{norm}} - 1)$

5.2.3 Modeling of the noise in HRMS

Because of the differences between TOF and Orbitrap detection processes, it is impossible to define a single noise model based on physical principles. We therefore consider the noise on each data point as independent from the other points, with a law depending on the intensity at the point, and on the observed intensities. An algorithm to obtain a raw estimation of noise variance as a function of the intensity will be detailed in the next chapter.

5.3 Proposed EIC model integrating these components

5.3.1 Definition

We can now propose a model for EICs for FIA HRMS. We first consider the case where an EIC contains a single sample analyte that we will denote A and some solvent molecules. As we assume that there is no retention in the pipework, we can consider that the concentration curves of all the analytes in the sample are identical (noted P), up to a multiplicative factor corresponding to their respective concentration (e.g., C_A for analyte A). P will be called **sample peak** in the rest of this thesis. However, even without matrix effect, the transmission of the analyte from the liquid-phase into the gas-phase is not 100% efficient, therefore a second multiplicative constant representing the efficiency of the transmission T_A is needed. These two constants may be grouped

into a single constant $k_A = C_A T_A$, and the EIC model of A becomes $k_A P$.

The analytes are then ionized in the electron spray, together with the molecules from the solvent. We assume that the amount of ions from the solvent is very small compared to the ions from the sample. Consequently, the matrix effect is caused by the molecules from the sample only. Therefore the matrix effect should be a function of $k_A P$. Since we selected a first-order exponential model for ME, we have $ME(k_A P) = a_A e^{b_A k_A P} = a_A M e^{b'_A P}$, with $b'_A = b_A k_A$. Therefore ME effect may be expressed as a function of P and two constant terms for the molecules, and without an explicit knowledge of the his concentration at the given time. We remind the reader that C_A is a constant. Moreover for commodity purpose we can define $a'_A = a_A / k_A$.

Furthermore, a baseline may be present if a molecule from the solvent has a similar m/z (as it this the case in Figure 4.1d). We refer to this baseline, which is specific to A, as B_A. Although the baseline is also affected by ME, we chose to ignore this effect as the baseline is generally small compared to the amplitude of the peaks, (if this is not the case, the analytical conditions need to be optimized). We can similarly define $B'_A = B_A / k_A$

The observed intensity of the EIC corresponding to A may therefore be written as:

$$I_A(t) = k_A P(t) - ME_A(P(t)) + B_A + \epsilon \tag{5.8}$$

or by fully developing the model and putting k_A as a factor:

$$I_A(t) = k_A(\underbrace{G_{\mu,\sigma} * E_\tau(t)}_{P} - \underbrace{a'_A(e^{b'_A \times (G_{\mu,\sigma} * E_\tau(t))} - 1)}_{ME} + B'_A) + \epsilon \tag{5.9}$$

With ϵ is the heteroscedastic noise. This factorized model is in practice simpler to use than the non factorized version, therefore for commodity purpose we will use it in the following section, and write a'_A, b'_A, B'_A as a_A, b_A, B_A for commodity purpose. A visualization of the contributions from each of the three terms is shown on Figure 5.5. The proposed model includes 7 parameters:

- 1 multiplicative constant representing the concentration and the ionization factor: k_A.
- 3 derived from the EMG function used to model the sample peak (P): μ, σ, τ.
- 2 derived from the first order exponential used to model matrix effect (ME_A): a_A, b_A.

- 1 related to the solvent baseline: B_A.

Out of these seven components, B_A can be extracted from the EIC (by focusing on the intensities at the beginning of the acquisition), and μ, σ, τ are identical across all the EICs.

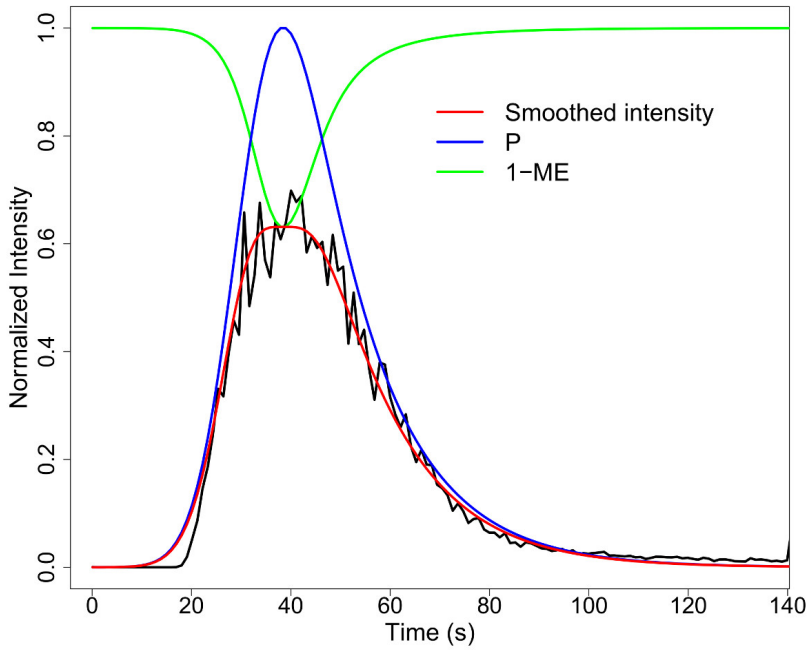
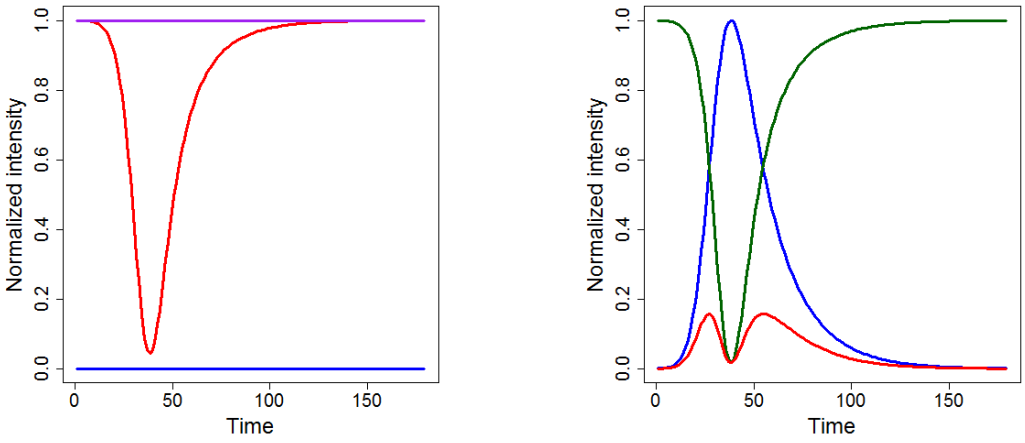


Figure 5.5: Fitting of model described in Equation (5.8) (showed in red) on a real (rescaled) EIC extracted from the **serFusion** dataset. The sample peak is shown in blue, the matrix effect in green (here $1 - ME$ is plotted for visualization purpose). The solvent baseline is equal to 0 on this EIC.

As expected, the proposed model explains the various types of observed EICs shown in Figure 4.1. On the EIC 4.1a, the major contributor to the intensity is the sample peak. However, even in that case of a "well-behaved" EIC, ME impacts on the maximum height of the sample peak, as suggested by the simulation from Figure 5.5. The "suppressed" EIC shown in Figure 4.1b results from a similar contribution of the sample peak and of the ME, as explained by the model on Figure 5.6a. In the absence of a model, such observed EICs would have been mistakenly interpreted as two distinct peaks. Finally, Figure 4.1c illustrates an EIC from solvent alone: the sample peak is absent, but the baseline is still affected by ME. Such a profile is also explained by our model (Figure 5.6b).

These simulations highlight the value of our model to explain the various types of EIC profiles observed on real data sets, depending on the ME and the baseline.



(a) (b)

Figure 5.6: Simulated EIC using our Model (5.8). Figure a) results from a ME of similar intensity as the sample peak (here we have set $a = 0.05$ and $b = 50$). Figure b) shows a signal of solvent affected by ME, with $a = 0.25$. In both cases we set $\mu = 30, \sigma = 8$, and $\tau = 0.05$ (i.e., the sample peak is identical for the two EICs).

Chapter 6

Construction of a processing workflow for FIA-HRMS data: proFIA

In this section we will detail our suite of algorithms created for the processing of FIA-HRMS data (Figure 6.1). The whole pipeline is available as the *proFIA* R package in the Bioconductor repository (DOI: 10.18129/B9.bioc.proFIA; Delabrière et al. 2017). A set of demonstration data is also available in the *plasFIA* extracted from the **serFusion** dataset. The structure and some of the algorithms were inspired by the reference softwares for LC-MS processing reviewed previously (Table 4.1).

The main innovation is the development of an EIC model integrating matrix effect, as presented in the previous chapter 5: because the expected peak shape highly depends on the experimental setup and is not clearly defined, the sample peak is first estimated on well-behaved EICs, and subsequently used for matched filtration. ME was also considered when selecting the m/z traces (m/z band detection). Finally, potential retention in the FIA pipe was taken into account, as such an effect was observed in the data sets.

The critical part of proFIA is the detection and the quantification of peaks within each sample. As in LC-HRMS, it includes the detection of m/z traces followed by the analysis of EICs. As we have seen in the previous chapter, in a theoretically perfect FIA-MS acquisition (i.e., without matrix effect, baseline solvent, or retention), all EICs should have the same shape, and should only differ by a factor corresponding to the analyte concentration and its ionization efficiency. The sample peak is common to all analytes from the sample. Estimation of the sample peak is essential, since it is used in many steps of the processing (including m/z band detection and EIC matched

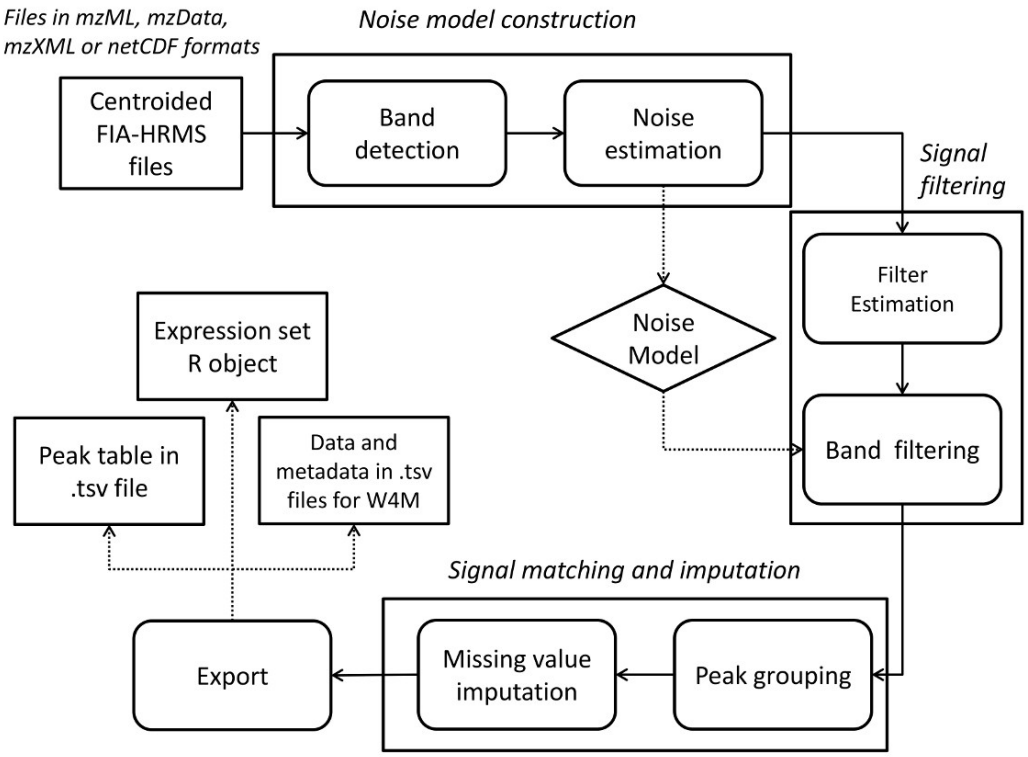


Figure 6.1: **proFIA workflow**. Input files are the centroided raw data in standard formats. First, within each sample file, mass bands are detected in the m/z by time plane (using the *ppm* and *dmz* parameters), and a noise model is built. The sample peak is modeled, and subsequently used within each m/z band to detect the temporal limits of the analytes signal, and evaluate its quality. Second, the previously detected features are grouped between samples by using a kernel density estimation in the m/z dimension. Finally, missing values can be imputed with either a k-Nearest Neighbors or a Random Forest based method. The output is the peak table containing the characteristics of each feature (m/z limits, mean correlation with the sample peak as a quality metric, and intensity in each sample)

filtering). It is not possible, however, to obtain a direct estimation, since most of the EICs (and hence the TIC) are affected by matrix effect. Therefore a selection of EICs which are not too affected by matrix effect is done.

6.1 Initial estimation of the sample peak limits

An initial raw estimation of the limit of the sample peak is first performed. This step aims to obtain a 3 points estimate: the first scan on which the sample injection is visible (l_{inj}), the maximum of the injection (m_{inj}), and finally the last scan of the injection

(u_{inj}). These three points are used in the first three steps of the algorithm (band detection and noise estimation, and sample peak estimation) to detect the signals affected by the solvent, and to discard them when determining the sample peak shape. Therefore to allow the maximum of flexibility and robustness regarding the sample peak shapes, a geometric algorithm was used:

1. The Total Ion Chromatogram (TIC) is rescaled in the $[0, 1]$ interval both in the time and intensity dimensions.
2. The solvent value is set to the first intensity of the TIC.
3. The starting time of the injection (lower limit of the injection window: l_{inj}) is set to the first scan corresponding to 3 consecutive increasing intensities superior to the solvent.
4. m_{inj} is set to the maximum of the smoothed TIC (on the sequence smoothed using a 5-points median filter).
5. All scans superior to $l_{inj} + m_{inj}$ are considered as candidates for the end of injection (upper limit: u_{inj}). For each candidate time u, let us denote M, U, and L the summits of the triangle joining the signal intensities at time points m_{inj}, u and at the last time point of the chromatogram. The limit of injection u_{inj} is computed as the time point maximizing $cos(UM, UL) - UM.UL$ (the second term avoids the selection of points with noisy intensities at the end of the chromatogram).

The output is a triplet $l_{inj}, m_{inj}, u_{inj}$ which will be used in the subsequent steps. m_{inj} and l_{inj} are good estimates of the true injection time and maximum intensity of the sample peak as long as the majority of the molecules are not affected by retention in the pipework. The limit of injection u_{inj} is a less reliable estimate as any retained molecule will strengthen the right-tail of the peak.

6.2 m/z band detection

The m/z band detection step in *proFIA* differs from the building of m/z traces in LC-MS in several ways. Because of ME, the intensity may decrease at the maximum of the sample peak (Figure 4.1b), resulting in missing values in some of the consecutive scans when the signal is under the limit of detection of the mass spectrometer. Therefore gaps

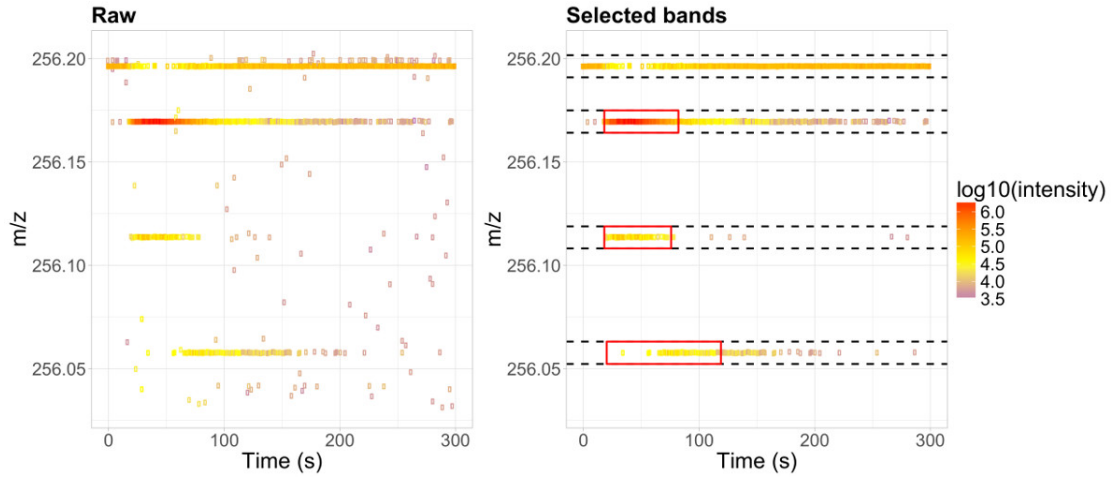


Figure 6.2: **Band and peak detection on serFusion**: Data before (left) and after (right) band detection and filtering. Dashed black lines indicate the detected bands (the m/z window is enlarged for visualization purpose); red rectangles correspond to the regions identified by the peak detection step as containing the analyte signal. Band at m/z 256.20 is discarded during the peak detection step

need to be allowed when building the traces in the time dimension. Another difference with LC-MS is the fact that m/z bands are detected without limits in the time domain (Figure 6.2, dashed lines). Such bands, however, are not detected by binning the m/z scale (which would result in a loss of accuracy), but by grouping centroids with close m/z values. Each band (denoted B) consists of a list of *centroids* characterized by their 3-dimensional coordinates $(mz, int, scan)$, and a scalar $meanMz$. The data points are processed sequentially by increasing time and increasing m/z. The full algorithm is detailed in 1.

The algorithm starts by initializing a new band for each point of the first scan. These bands are stored in a linked list sorted in increasing order of m/z. Data points are processed 1) by scan order and 2) by m/z value within each scan. For each new centroid p, the bands at the vicinity are selected from the full bands list (Line 6) using a tolerance in ppm and a minimum threshold dmz (Line 5), to handle the loss of accuracy at low masses. If no candidate is found, a new band is created directly(11).

In the case where at least a band is found (function InsertCentroid), we first select the band B which is closest to p by designing the following metric $Dist(B, p)$ (with b denoting the last added point in B):

$$Dist(B, p) = \frac{|p.mz - b.mz|}{ppm \times p.mz} + \frac{log(p.int) - log(b.int)}{2} \tag{6.1}$$

(note that this definition takes into account the local deviation as it considers b). Then

Algorithm 1 Band detection algorithm

1: **procedure** FINDBANDS($Scans, ppm, dmz$)
2: $BL \leftarrow$ Initialize bands with points in the first scan. ▷ List of bands
3: **for** s in $1, \ldots, |Scans|$ **do**
4: **for** $p = (mz, int)$ in $Scans[s]$ **do**
5: $tol \leftarrow max(dmz, ppm \times mz \times 10^{-6})$
6: $C_p \leftarrow$ Bands $B \in BL$ such that $|B.meanMz - mz| < tol$ ▷ Candidate bands
7: $inserted = False$
8: **if** $|C_p| \geq 1$ **then**
9: $(p, inserted) \leftarrow$ INSERTCENTROID(C_p, p)
10: **if** not $inserted$ **then**
11: Create a new band with p only
return BL
12: **end procedure**

1: **procedure** INSERTCENTROID(C_p, p)
2: $C_{p'} = C_p$
3: **while** $|C_p| \neq \emptyset$ **do**
4: $B_p = \text{argmin}_{c \in C_p} Dist(p, c)$
5: $b_{last} \leftarrow$ last point of B_p
6: **if** $b_{last}.scan = p.scan$ **then** ▷ B_p already contains a point for scan s
7: $B_p^{-1} \leftarrow B_p$ without b_{last}
8: **if** $Dist(p, B_p^{-1}) < Dist(b_{last}, B_p^{-1})$ **then**
9: Replace b_{last} by p in B_p
10: $p \leftarrow b_{last}$ **return** INSERTCENTROID$(C_{p'}, p)$
11: **else**
12: $C_p \leftarrow C_p / B_p$
13: **else**
14: Assign b_p to B_p **return** $(p, True)$
return $(p, False)$
15: **end procedure**

(Line 14), if there is no point b from this scan in B already, p is added to the band. Otherwise, p is compared to b using $Dist$, and the other point is inserted once again if p is a better candidates (Line 10). If on any points all the candidate bands are considered, the stopping criterion of the while is met and the point is inserted into a new band. This procedure ensures that the number of incorrect assignments is kept to a minimum, and avoid the incorporation of noisy points with close m/z in the mass traces.

The list of detected bands is subsequently cleaned by discarding bands without at least 3 successive points in the time dimension.

To avoid the risk of band splitting (i.e., two bands are created for the same m/z feature), and increase the robustness of the algorithm with respect to the ppm and dmz parameters, two bands are fused when they have less than $2 \times ppm \times mz$ difference and share less than 2 scans.

The final filtering step focuses on the quality of detected bands, to discard features which are too suppressed by ME. The proportion of points from each band B within the sample peak limits (l_{inj}, u_{inj}) determined in the previous section (6.1), $F = (\sum_{c \in B.centroids} \mathbb{1}(l_{inj} \leq c.scan \leq u_{inj}))/(u_{inj} - l_{inj})$, must be superior to the user defined $bandCoverage$ parameter, the default parameters in the proFIA software is 0.4. An example of such a discarded band is displayed in Figure 6.2 (top band on the right panel).

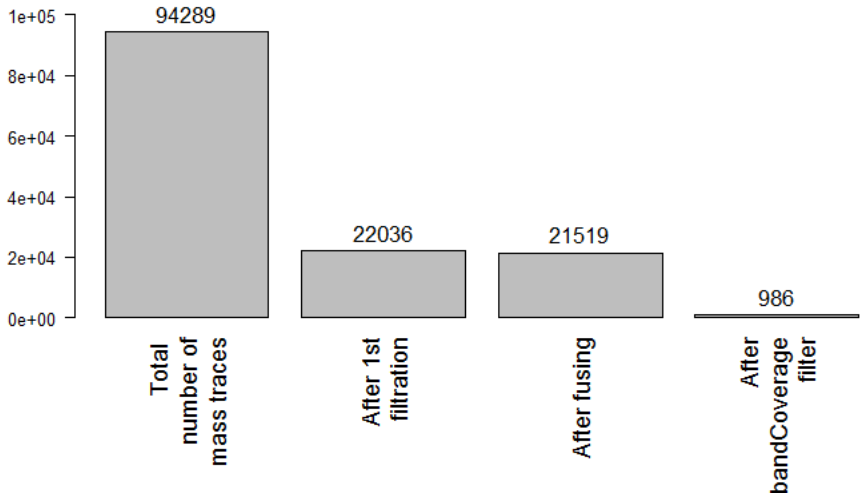
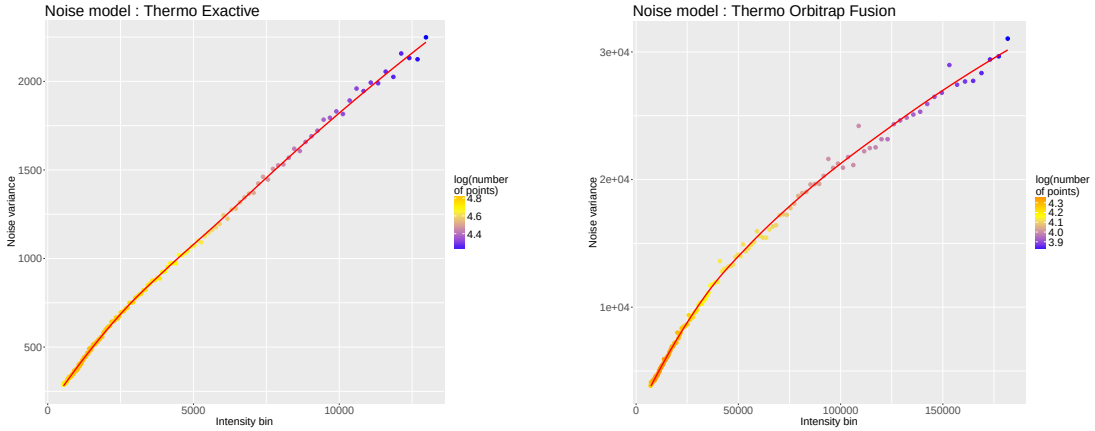


Figure 6.3: **Impact of the filtering steps on the total number of detected m/z features (i.e., bands) from an acquisition of the plasFIA package.**

This filtering steps may result in up to 100-fold reduction of the number of detected bands (Figure 6.3).

The bands are then used for the estimation of noise and the characterization of the sample peak model.



(a) (b)

Figure 6.4: **Noise variance estimates**. The ordinate values of the points correspond to the estimated variance in each intensity bin, and their color encodes the logarithm of the number of points in the bin; red curve: LOESS fit. a) Q-Exactive (**serExactive**), and b) Fusion Orbitrap instruments (**serFusion**).

6.3 Model of the noise variance

6.3.1 Noise variance estimation

Noise depends on the technology of the MS analyzers and detectors (e.g., Time-Of-Flight vs Fourier Transform). To develop a robust method, we therefore implemented a non parametric estimation of the noise variance as a function of intensity. This estimation will be used in the subsequent steps (e.g., to filter out solvent features in section 6.5.3 by discriminating analyte signal from solvent baseline).

The proposed algorithm is a variant from Wentzell and Tarasuk 2014. In the original approach, noise variance is estimated in each intensity bin from a logarithmic spaced series of bins covering the full intensity range of the whole acquisition. More specifically, for each EIC, the signal is filtered using a low-pass filter. The noise is estimated as the difference between the smoothed signal and the original data. While this strategy results in a very raw estimate of noise on individual measures, each bin contains tens of thousands estimates, and the mean of each bin therefore provides a robust estimate of the variance. In the original method, a curve is then fitted to the bin means (Figure 6.4).

Because proFIA is intended to process more than one sample at a time, we extended this algorithm to estimate the noise on multiple files simultaneously: the range of

intensities is set from a minimum provided by the user (who has some expertise on the noise level) to a maximum equal to twice the maximum intensity in the first processed file. The logarithmically spaced bin scale is built and is identical for all files. The binning process on each individual file f returns for each bin b a pair $(V_{b,f}, S_{b,f})$, where $V_{b,f}$ is the noise mean and $S_{b,f}$ is the number of points. The global variance estimate for each bin across the samples (V_b) is then computed as the weighted mean of the variances estimated in the individual files: $V_b = \frac{\sum_{f \in samples} V_{b,f} \times S_{b,f}}{\sum_{f \in samples} \times S_{b,f}}$.

6.3.2 Regression

A Local Polynomial Regression (LOESS) is then performed (LOESS was shown to outperform a simple polynomial regression). Because there is usually only a few high intensity points in each file, noise cannot be estimated in a robust manner in these bins: we therefore discard the set of high intensity bins with a cumulative proportion of points inferior to 5%. The minimum intensity (provided by the user) may be lower than the real value, leading to non-robust smoothing of intensity and over-estimation of the noise in the lowest bins (Figure 6.5) as described in Wentzell and Tarasuk 2014. Variance estimates for low intensities are thus smoothed (moving average of 7 points), and the intensity bin corresponding to the minimum of the smoothed variance is used as the lowest bin for the LOESS regression.

The result of this step is a model of the noise variance of the intensity, as well as an interval of definition $[minIntensity, maxIntensity]$.

6.4 Sample peak determination

After band detection, it is now possible to work in the time dimension to perform peak quantification on each Extracted Ion Chromatogram (EIC). We will see however that our EIC model (chapter 5) is too complex to be fitted directly to the data (6.4.1). In this section, we therefore use the model to estimate the sample peak P (common to all EICs) by using the best behaved EICs (strong intensity, little ME, no baseline). In the next section (6.5), P will be used to detect the time limits of the signal on each EIC by matched filtration, to subsequently allow an accurate quantification of the analyte.

In chapter 5, we proposed the following model for EIC:

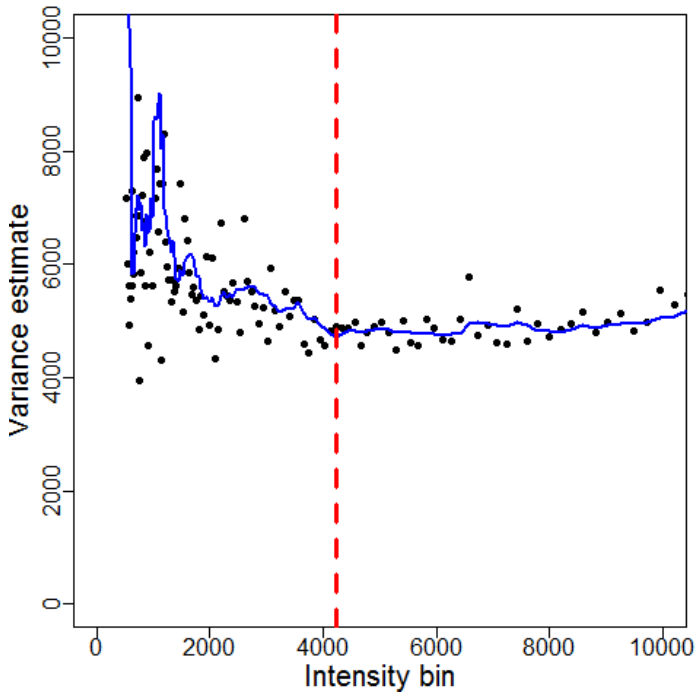


Figure 6.5: **Determination of the lowest intensity bin for the model:** A bin scale starting at a minimum intensity(500) too low for robust variance estimation. Blue: smoothed variance; Red: selected lowest bin for the LOESS regression (**serFusion**)

$$I_A(t) = k_A \times (\underbrace{G_{\mu,\sigma} * E_\tau(t)}_{P} - \underbrace{a_A(e^{b_A \times (G_{\mu,\sigma} * E_\tau(t))} - 1)}_{ME(P)} + B_A) + \epsilon \tag{6.2}$$

which may be written in a complete form as :

$$\begin{aligned} I_A(t) = & k_A \times \Bigg(\underbrace{\frac{1}{\tau\sqrt{2\pi\sigma^2}} \int_{x=0}^{t} e^{--\frac{(x-\mu)^2}{2\sigma^2}} \times e^{-\frac{t-x}{\tau}} dx}_{P} - \\ & \underbrace{a_A(e^{\left(b_A \times (\frac{1}{\tau\sqrt{2\pi\sigma^2}} \int_{x=0}^{t} e^{--\frac{(x-\mu)^2}{2\sigma^2}} \times e^{-\frac{t-x}{\tau}} dx)\right)} - 1)}_{ME(P)} + B_A \Bigg) + \epsilon \end{aligned} \tag{6.3}$$

The least square estimate objective function of this model is non-convex, a global minimum cannot be computed by a simple method. We therefore evaluated the efficiency of the regression on simulated data.

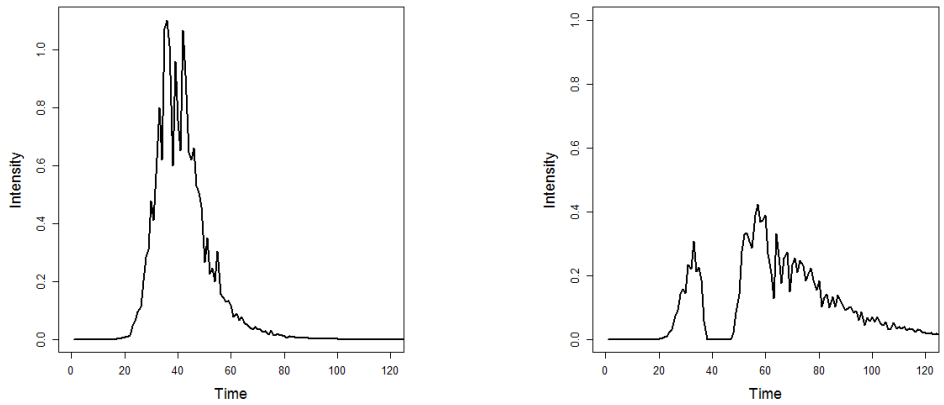
parameter	distribution
μ	$\mathcal{N}(40, 8)$
σ	$\mathcal{U}(3, 8)$
τ	$\mathcal{U}(0.01, 0.2)$
$a = 10^c$	$c \sim \mathcal{U}(-2, 0.5)$
$b = 10^d$	$d \sim \mathcal{U}(-2, 3)$

Table 6.1: Parameters used to simulate EICs

6.4.1 Regression on simulated data

We generated 20 distinct sample peaks P and, for each of them, 40 EICs (half with little ME, and the other half with strong ME; Table 6.1).

To simplify the regression problem, baseline was omitted from the simulation, and the EMG term was normalized to 1 before calculating the ME term. A multiplicative noise was added to each point: the noise was drawn from a Gaussian distribution with variance $0.2 \times i$, where i is the intensity of the point. Simulations successfully reflected the EICs from real FIA-HRMS data sets (compare Figure 6.6 with Figure 4.1). The amount of ME was evaluated as the ratio between the area of the ME term and the area of the sample peak P term.

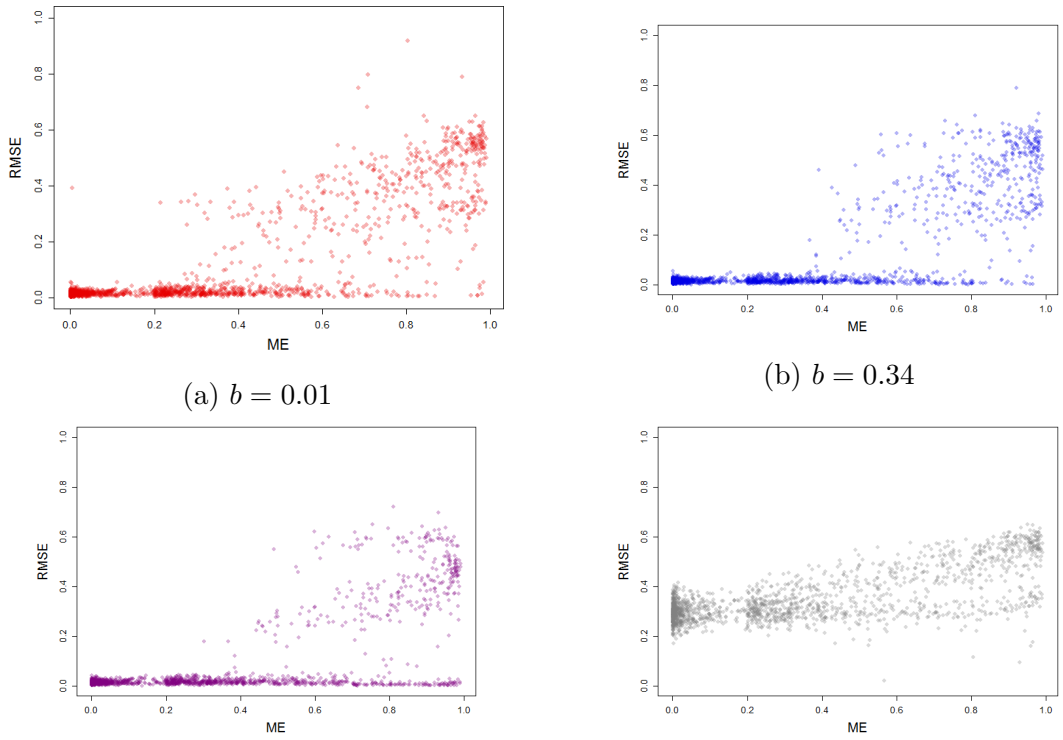


(a) (b)

Figure 6.6: **Simulated EICs**. Equation 6.2 was used to simulate a wide variety of EIC, including well-behaved EIC (a) and suppressed and tailed EIC (b).

The Levenberg-Marcquadt algorithm implemented in the *minpack.lm* R package (Moré 1978) was used to perform the regression, as it is one of the most popular algorithms for optimization, and has successfully been applied to the fit of EMG functions to m/z peaks (Lange et al. 2006). The initial parameters of the sample peak (μ, σ, τ) were estimated using measurements of peak height and peak asymmetry as described

in Brooks and Dorsey 1990. Due to the high fluctuations of ME, direct estimation of initial value for b was not possible. We used instead 4 sets of starting points corresponding to increasing levels of the b parameter, from 0.01 (low ME), 0.69 (ME masking half of the sample peak height at the peak apex), 1.09 (completely masking the peak height, similar to 6.6b), and up to $b = 5$ (extreme ME). a was kept to 0.5 in all settings. The Root Mean Squared Error (RMSE) of the actual vs modeled sample peak P was computed, and plotted against the ME (Figure 6.7). While P is well modeled when ME is low, RMSE is above 5% for more than 58% of the EICs in case of high ME (even when high b values are given as starting point).



(a) $b = 0.01$

(b) $b = 0.34$

(c) $b = 0.5$

(d) $b = 5$

Figure 6.7: **Quality of sample peak modeling as a function of the amount of ME in the EIC.** EICs with various amounts of ME were simulated using Equation 6.2 and parameter distributions from Table 6.1. The quality of the sample peak P modeling obtained with increasing initial b values was evaluated with the RMSE.

In conclusion, direct estimation of the sample peak P on EICs with strong ME is not accurate even on these simplified simulated data (no baseline, similar noise variance). We therefore selected for proFIA a two step strategy. First the sample peak was estimated by regression for the well behaved EICs only (i.e., the one less affected by ME and high intensity). Then this sample peak was used to perform matched filtration on all the EICs (to ensure robustness).

6.4.2 Selection of well behaved EICs

"Well-behaved" EICs (corresponding to the previously detected m/z bands) were selected as follows:

1. No baseline.
2. High intensity.
3. Small ME.
4. No retention time shift.

Condition 1 is fulfilled if the intensity of the points before the start of the sample peak (estimated as l_{inj} in the chapter 6.1) is 0. For condition 2, the maximum intensity value of each EIC is computed: only the EICs with a maximum within the top decile are selected. Criterion 4 is met if the beginning of the EIC peak (set to the first three points with a consecutive increment in intensity) falls within 1s of a refined l_{inj} estimate. For condition 3 to be fulfilled, the delay between the apex of the EIC and m_{inj} must be less than 2 s.

Among the EICs meeting the 4 conditions the 20 most intense are selected. The set of these "well-behaved" EICs is denoted as M. The sample peak window and apex time estimates $(l_{inj}, m_{inj}, u_{inj})$ are updated by considering the sum of these EICs as the TIC an reapplying the initial algorithm for the 3-points estimate of the sample peak (section 6.1). The initial parameters for the sample peak P are also updated by applying the moment methods (Brooks and Dorsey 1990) to this summed EIC.

6.4.3 Regression of the sample peak

Because the sample peak is supposed to be the same for all the well-behaved EICs, the regression is performed on all the M EICs simultaneously, to ensure the convergence to a local maximum. While the parameters of the EMG (μ, σ, τ) are common to all M EICs, the ME parameters a_j, b_j and the intensity coefficient k_j are specific to the analyte from EIC_j. Therefore we decided to minimize the combined error function using the same data, to obtain an estimation of μ, σ, τ as robust as possible. Our error function then becomes:

$$\sum_{j \in M} |I_j - k_j P_{\mu,\sigma,\tau} + ME_{a_j,b_j}|$$

This quantity is estimated using the Levenberg-Marcquadt algorithm. The initial values of μ, σ, τ were determined from the sum of the M EICs by using the moment methods. The initial parameters values for all (a_j, b_j) were $(0.5, 0.69)$ since this corresponds to a low ME. All the EICs of were normalized before the regression to ensure that the errors have the same scale. All k_j parameters were therefore set to 1. Noise was neglected on these high intensity EICs.

After this step an estimate of the sample peak P is available. P is used on the EIC from each m/z band to 1) perform matched filtration and detect the time limits of the peak for each analyte individually, and 2) evaluate the amount of ME.

6.5 Peak detection using modified matched filtration

An approach based on the sample peak model P determined previously was used for peak detection on each EIC, because peaks are often very noisy. Matched filtration was used over wavelets transform because in FIA a peak greatly differing from the sample peak determined earlier is probably ill-formed, there was therefore no need for a rescaling of the injection peak. Matched filtration (described in section 4) has several advantages for FIA, including robustness to high levels of noise, and low sensibility to rescaling of the sample peak (e.g., in the case of the low intensity EICs). However, the classical approach had to be modified to take into account ME and the absence of large signal free regions on the EIC. Our algorithm consists of two main parts: detection of the putative peak borders and discarding of signals which are too close to the baseline.

6.5.1 Peak limits estimations

The putative peak limits were determined on the EIC by using matched filtration of the sample peak P. Because of ME, the initial peak detection on the filtered sequence is refined by detecting a second maximum on the convolved sequence. The full algorithm is described in algorithm 2.

First, the algorithm detects a local maximum (m denoting the apex) of a convolved sequence (blue curve in Figure 6.8). A first guess of the initial peak limits is obtained by descending the slopes of the peak. The processing then depends on the position of m relative to the sample peak time characteristics $l_{inj}, m_{inj}, u_{inj}$. All the EICs

Algorithm 2 Determination of peak limits, DownSlopes just follows the slopes starting from a local maximum until a minimum is found.

function FINDPEAKLIMITS($I, P, S, l_{inj}, m_{inj}, u_{inj}$)
  $n \leftarrow$ length of I
  $n_p \leftarrow$ length of P
  $C \leftarrow I * P$
  $m \leftarrow \text{argmax}\, j \in 1 \ldots nC[j]$
  $l, u \leftarrow DownSlopes(C, m)$
  $R \leftarrow False$
  if $l_{inj} < m$ **then**
    if $m < m_{inj}$ **then** ▷ Left part of the peak
      $m' \leftarrow \text{argmax}\, j \in u \ldots u_{inj} C[j]$
      if m' is a local maximum **then**
        $l', u' \leftarrow DownSlopes(C, m')$
      $l, u \leftarrow MaxRange(u, l, u', l')$
    if $m_{inj} < m < u_{inj}$ **then** ▷ Right part of the peak
      $m' \leftarrow \text{argmax}\, j \in l_{inj} \ldots lC[j]$
      if m' is not a local maximum **then** ▷ Refinement (triangular filter)
        $C' \leftarrow I * S$
        $m' \leftarrow \text{argmax}\, j \in l_{inj} \ldots lC'[j]$
      if m' is a local maximum **then**
        $l', u' \leftarrow DownSlopes(C', m')$
        $l, u \leftarrow MaxRange(u, l, u', l')$
    if $u_{inj} < m$ and $(u - l) \geq n_p$ **then** ▷ Retention in the pipework
      $R \leftarrow True$
  elsereturn $(R, -1, -1)$
  $Refine(l, u)$ **return** (R, l, u)
end function

with $m < l_{inj}$ are discarded as it means that the best position of the peak occurs before the sample reaches the mass spectrometer (24). For the other EICs, m usually falls between the injection limits. In this case, the algorithm looks for a second local maximum on the other side of m_{inj} (line 9 and 14) to take into account the signals strongly affected by ME (Figure 6.8b). If ME is low, no second maximum is detected (Figure 6.8a). However in case of signal suppression, a second maximum may be detected. Because the first "peak" of strongly suppressed signals may be very sharp (and thus may not be detected as a local maximum), a convolution with a narrower filter (triangular wavelet denoted S) may be used to accurately estimate the left limit of the peak (line 17). The support is extended to include both peaks in lines 21 and 13.

Finally the peak limits are refined. The $Refine$ function looks for a local minimum to ensure that the detected peak is visually correct. If the refined limit l is inferior to l_{inj}, l is set to l_{inj}.

Now that the time limits (l_j, u_j) for every peak have been detected, the amount of solvent must be checked to ensure that it does not prevent reliable quantification (Figure 6.8c).

6.5.2 Solvent removal

EICs with a large part of their intensities potentially coming from the solvent must be discarded. Although many of these signals have been already removed at the previous matched filtration step, an additional filtering is applied by looking at the points of the EIC before l_{inj}. If at least 3 points have a non zero intensity, the EIC is considered as being affected by the solvent, and the solvent intensity noted i_{sol} is set to the mean of the non 0 EIC points before l_{inj}.

If no solvent is detected the peak is validated, as it has already passed the $bandCoverage$ filter in the band detection steps, meaning that there is enough point in the sample peak for reliable integration.

If the EIC includes some solvent, it is kept if the amount of signal is sufficient compared to the baseline. Due to the the limited duration of the acquisition, there are very few signal-free region in the EICs: as a result, a local estimation of the noise based on the quantile of the convolved sequence, as it often used with matched filtration or wavelets methods, is not possible. We therefore rely on the comparison of i_{sol} and the maximum intensity of the EIC m_{eic}: if $m_{eic} < 1.5 i_{sol}$ (the 1.5 threshold

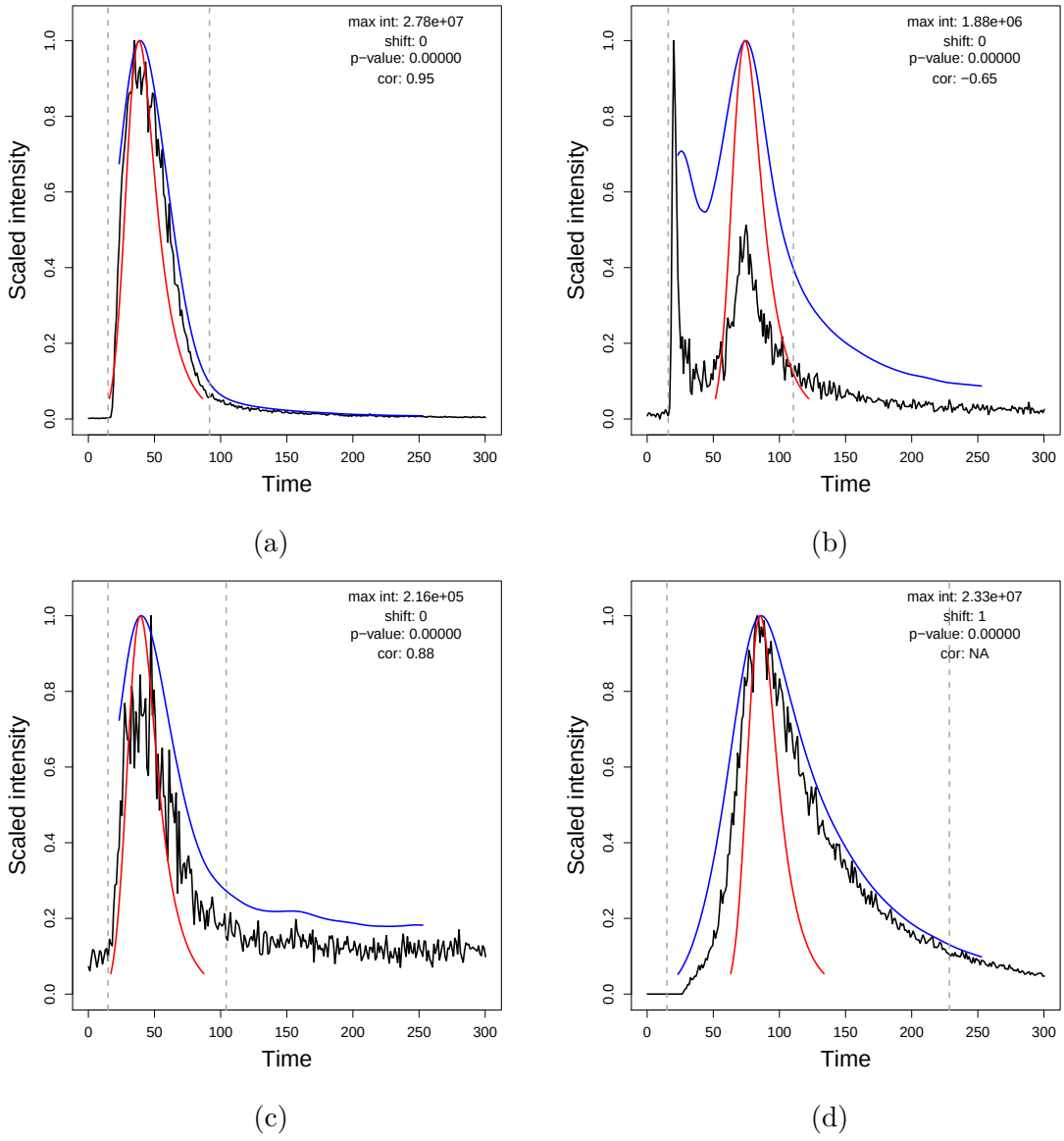


Figure 6.8: **proFIA filter on real EICs extracted from serFusion**. Black line: EIC, blue line: convolved sequence C, red line: matched peak P at the maximum of C, grey dotted vertical lines: peak limits for subsequent integration. Legend: quality metrics provided by proFIA (maximum intensity, indicator of retention shift (d), p-value obtained when testing the significance of signal compare to baseline (c) and correlation with the sample peak when the peak is not shifted).

was determined empirically), the EIC is discarded, as it means that the relative level of solvent is very high compared to the signal originating from the sample, such level of solvent invalidate our hypothesis on the negligibility of i_{sol}. If $m_{eic} > 1.5 i_{sol}$ a statistical test is used.

6.5.3 Statistical testing of sample contribution

The null hypothesis H_0 states that the signal consists of baseline only: $I = B + \epsilon$, where the noise ϵ is modeled at each time i by $\epsilon_i \sim N(0, V(B[i]))$ with V the estimation of the variance in function of the noise obtained in section 6.3. If we assume the independence of noise with time, the total noise between the signal limits also follows a normal distribution:

$$E = \sum_{i=l}^{u} \epsilon_i \sim N\left(0, \sum_{i=l}^{u} V(B[i])\right)$$

The statistic $\hat{E} = \sum_{i=l}^{u} \hat{\epsilon_i} = \sum_{i=l}^{u} (I[i] - \hat{B}[i])$ is then compared to this distribution (with $\hat{B}$ being the linear segment between the peak limits $I[l]$ and $I[u]$). The peak is kept if the unilateral test p-value is inferior to a given threshold (0.01 by default; Figure 6.10c). While this test makes the hypothesis that the baseline is linear, it is still valid when the baseline is affected by ME, as we show below.

6.5.4 Extension of the testing to include matrix effect

Our hypothesis H_0 remains unchanged (stating that the signal consists of baseline only). However, we now include ME in our equation describing intensity under H_0:

$$I = B - ME(P) + \epsilon$$

Our previous linear estimation of B is superior to the quantity $B - ME(P)$, as the contribution of ME is always negative. As a result, our previous estimate of the noise ϵ_i at each data point i, $\hat{\epsilon_i} = I - B$ is inferior or equal to the real value of $\epsilon_i = I - B$, and our test statistic $\hat{E} = \sum_{i=l}^{u} \hat{\epsilon_i} \leq E$ is always underestimated (red cross on Figure 6.9). The calculated p-value is therefore biased upward:

$$P_{H_0}(X > E) \leq P_{H_0}(X > \hat{E})$$

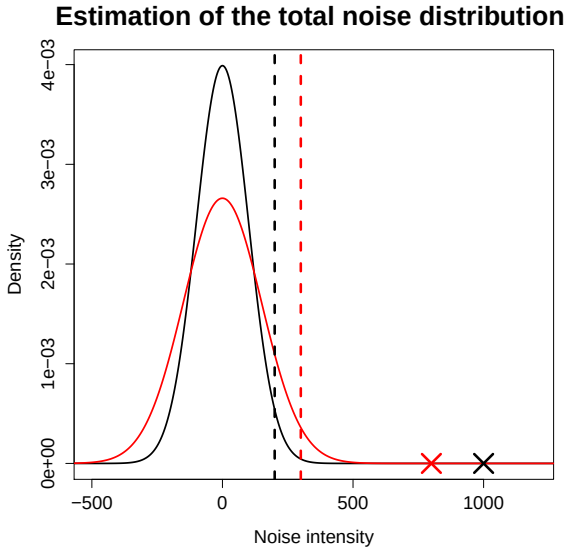


Figure 6.9: **Noise distributions and test statistics under** H_0. The true (respectively, estimated in the absence of ME) distribution and statistic (cross) are shown in black (respectively, red).

Moreover as the variance estimation V is an increasing function, $\hat{B} > B$ results in $V(\hat{B}) > V(B)$. Consequently, the variance of E is over-estimated. Finally, let us consider the following property of Gaussian distributions: if X and Y are random variables following two Gaussian distributions of mean 0 and standard deviations $X \geq Y$, we have for all $U > 0$:

$$P(X > U) \geq P(Y > U)$$

Since the standard deviation of E is inferior to the true standard deviation of E, the above property results in $P_{H_0}(X > \hat{E}) \geq P_{H_0}(Y > \hat{E})$ with $X \sim N(0, \sum_{i=l}^{u} \hat{B}_i))$ and $Y \sim N(0, \sum_{i=l}^{u} V(B_i))$. The p-value calculated previously under the hypothesis $I = B + \epsilon$ (i.e. in the absence of ME) is therefore a superior to the p-value that would be obtained by adding the ME to the model. In conclusion, using the p-value obtained without considering the matrix effect does not increase type II errors, i.e. errors resulting in a signal considered as feature whereas it is in fact the baseline solvent (to our experience there are few additional errors of type I, and such errors are not limiting for the quality of the preprocessing).

6.5.5 Quality metrics calculation

Compared to LC-MS peak picking algorithms, *proFIA* takes advantage of the sample peak to provide additional quality metrics (Figure 6.10). First, an indicator of

retention of the peak in the pipework is provided: if the detected maximum of the convolved sequence is closer to u_{inj} than to m_{inj} and no second maximum is detected on the convolved sequence on the left of the peak, the feature is flagged as $shifted$ (Figure 6.10d). Second, if the EIC is not shifted, an indicator of ME is computed as the correlation between P and the EIC within the sample peak time limits, cor. Correlation values above 0.6 suggest that ME is low (Figure 6.10a), whereas negative values warns against putative strong distortion of the signal (Figure 6.10b).

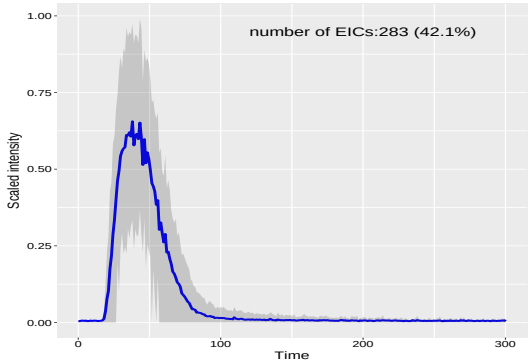
At the end of these steps, the features have been detected and quantified within each sample file. They now need to be matched between the different samples to build a global peak table of the data set for subsequent statistical analysis.

6.6 Inter samples features grouping

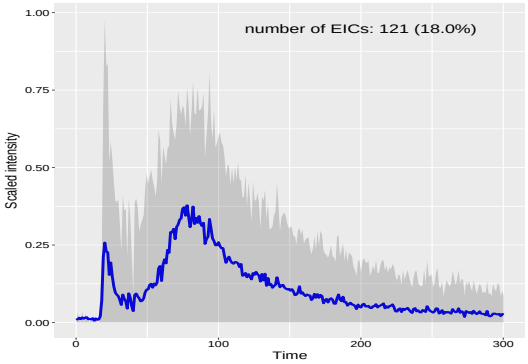
We used only the m/z dimension to group similar features, as variations in the retention time and the effect of ME in each class was not previously studied in FIA. A density estimation method with a Gaussian kernel, which proved successful in apLCMS (Yu, Park, et al. 2009), was used. However, since the mass accuracy of a mass spectrometer is expressed in ppm, we used the density on multiple overlapping windows with an increasing bandwidth along the mass axis. The main parameters are $ppmGroup$ the maximum authorized deviation in ppm and the minimum deviation threshold in Dalton $dmzGroup$. Each peak of the estimated density is considered as a feature candidate. Similarly to XCMS (C. Smith et al. 2006), features which are not detected in a sufficient amount of samples in at least one of the sample classes are discarded (the sample classes should define a priori groups of samples with similar peak profiles: e.g., blank vs matrix samples, or control vs cases; (C. Smith et al. 2006)). The threshold may be either a fixed number of samples or a fraction of the total number of samples in each class, therefore allowing the processing of unbalanced class sizes. After this step a feature by sample table is generated (often called a peak table in the MS jargon), using either the area of the peak, or its maximum intensity.

6.7 Missing value imputation

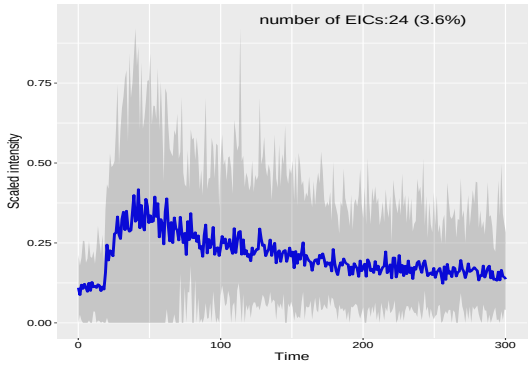
Some of the missing values in the peak table may result from a technical failure to detect a compound in a sample (e.g., for concentrations close to the limit of detection). An imputation step is therefore useful in the preprocessing workflow. The imputation



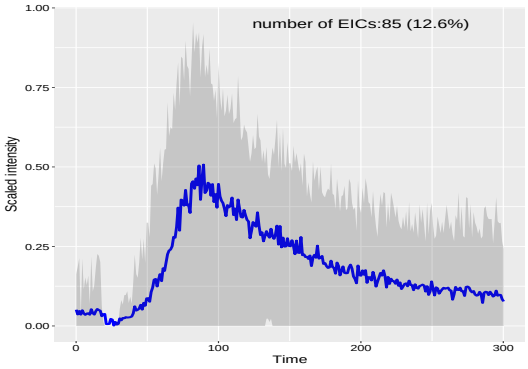
(a) $cor > 0.6$



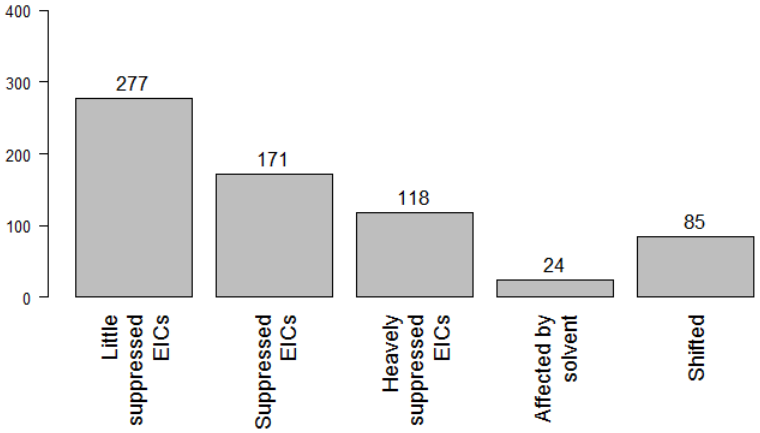
(b) $cor < 0$



(c) p-value $< 1\%$ (discrimination between signal and solvent)



(d) $shift > 0$



(e) number of EICs in each group

Figure 6.10: **Representative EICs from a sample of the serFusion dataset.** EICs have been scaled for visualization purpose. Blue curve: mean profile; grey area: 80% symmetric quantile of the selected EICs at each time point.

approach based on integration of raw signal within the expected region of interest, as implemented in the xcms.fillPeaks algorithm; (C. Smith et al. 2006), may fail to detect any signal in case of high-resolution data, because the detected bands are thin. As an alternative, a k-Nearest Neighbors method (KNN) applied to the peak table (which imputes a missing value as the average of the closest features) was shown to be optimal for DI-MS data (Hrydziuszko and M. Viant 2012). Such approach requires that features be scaled before nearest neighbors are computed. To refine the estimation of the intensity distribution for feature scaling, Shah et al. 2017 recently proposed to take into account the limit of detection of the instrument by using a truncated normal distribution model (KNN-TN). The described methodology, however, assumes that the distribution can be modeled by a single Gaussian, which may lead to errors in case of multiple sample classes. We therefore implemented a modified KNN-TN imputation method where the similarity with neighbors is computed for the samples of the same class only. Alternatively, the Random Forest approach is provided in proFIA, as implemented in the MissForest R package (Stekhoven and Bühlmann 2011) which was recently shown to achieve efficient imputation of missing values in MS datasets (Guida et al. 2016; Gromski et al. 2014).

At this final processing stage, the peak table is available for subsequent statistical analysis, and may be exported as a csv tabulated file or an Expression Set R object, which is a standard data structure for processed omics data in Bioconductor (Gentleman et al. 2004). Export is also possible to the Workflow4Metabolomics (W4M) 3 table format, which enables further analysis on the online platform (Giacomoni et al. 2014) (the proFIA tool has also been integrated into W4M, to enable comprehensive FIA workflow design and computation online with a Graphical User Interface (Delabrière et al. 2017; Guitton et al. 2017)).

Chapter 7

Evaluation of proFIA on experimental datasets

In this section, the performance of the proFIA software are evaluated on two FIA-HRMS datasets (3.1). To optimize the analytical conditions of metabolomics analysis of serum by FIA, a commercial sample of human serum was analyzed by Orbitrap HRMS instruments. In the **serFusion** dataset, the serum sample was spiked with a mixture of 40 representative compounds, at increasing concentrations. Acquisition was performed at very high resolution (100,000 at m/z = 200; Orbitrap Fusion). In the **serExactive** dataset, several dilutions of the spiked serum were studied at a high-resolution (50,000 at m/z = 200; Orbitrap Exactive). Both datasets are presented in more details in the introduction (section 3.1).

The performance of proFIA was assessed according to three evaluation criteria. Firstly, since there was no reference software for the processing of FIA-HRMS, proFIA was compared to the reference algorithm for peak-picking in LC-MS, centWave, which is implemented in the XMCS R/Bioconductor package (C. Smith et al. 2006): in particular, the reproducibility of the peak-picking and the total number of detected signals were compared (section 7.1). Secondly, the reproducibility of the peak-picking of proFIA in terms of detection and quantification was evaluated on replicate acquisitions (section 7.2). Thirdly, proFIA was compared to the manual peak-picking performed by an experimenter with the vendor software for data visualization and quantification (section 7.3). At the end of the chapter, an overview of the impact of the proFIA parameters on the reproducibility and sensitivity of the detection is provided (section 7.4).

Parameters	serFusion	serExactive
ppm	2	8
dmz	0.0005	0.001
p-value	0.001	
bandCoverage	0.4	
ppmGroup	1	4
dmzGroup	0.0005	

Table 7.1: **proFIA parameters** used for the processing of the **serFusion** and **serExactive** datasets

***centWave* parameters**	**Tested values**
ppm	0.5, 1, 2, 3
snthresh	3, 5, 10
prefilter(k)	2,3,4
prefilter(I)	100, 500, 1000
peakwidth(Max)	25, 50, 75, 100

***proFIA* parameters**	**Tested values**
ppm	1, 2, 3, 4, 5
bandCoverage	0.2, 0.3, 0.4, 0.5, 0.6, 0.8
pvalue	0.05, 0.01, 0.005, 0.001, 0.0001

Table 7.2: **Optimization of peak-picking parameters** for the *centWave* and *proFIA* algorithms.

The results described in the 7.2 and 7.3 sections were all generated using the sets of parameters described in Table 7.1. No missing value imputation was performed since the goal of this section is to evaluate the upstream peak-picking steps.

7.1 Comparison between proFIA and XMCS

proFIA and the *centWave* algorithm for peak-picking from XCMS (Tautenhahn et al. 2008) were compared on the **serFusion** dataset. Since the recently described "Isotopologue Parameter Optimization" method for LC-MS processing *centWave* described in Libiseller et al. 2015 could not be applied to our FIA data, we tested multiple sets of values. The minimal peak width of the wavelet scale was set to 5 s to ensure that suppressed peaks could still be detected. Multiple parameter values were also tested for *proFIA* (Table 7.2).

Two metrics were used to compare the peak-picking: 1) the reproducibility of quantification, as determined by the Coefficient of Variation (CV) of each feature intensity between the replicates (CV is the standard deviation divided by the mean), and 2) the sensitivity of detection, by counting how many features were detected in all triplicates. The first metric, intensity CV, was only computed for signals from the top quintile of each sample, ensuring that only peaks corresponding to ions from the sample are taken into account, and not the chemical noise. The results on the

serFusion dataset are summarized in Figure 7.1.

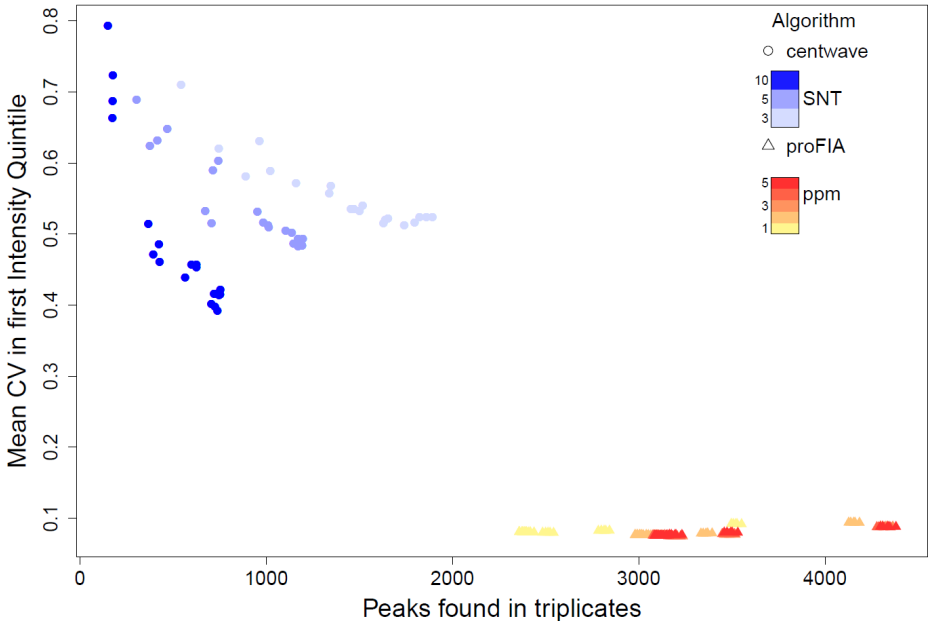


Figure 7.1: **Comparison of *centWave* and *proFIA* for FIA-HRMS data preprocessing:** Each point represents a combination of parameters. Points are colored according to the values of the most significant parameter (*SNT* for *centWave* and *ppm* for *proFIA*).

Figure 7.1 shows that *proFIA* outperforms *centWave* on the **serFusion**data set for both metrics. The CV for all the parameter sets of *proFIA* is inferior to 10% but superior to 35% with *centWave*. This may be explained by many factors, including the fact that the peak model of *centWave* does not take into account the asymmetry of the peaks, and that this model is not compatible with split peaks (due to ME). The reproducibility of peak detection by *proFIA* also outperformed *centWave*, which may result from the requirement of continuity in the time dimension by *centWave* (preventing the detection of suppressed peaks), and by the baseline estimation procedure of *centWave* which is not suited to FIA EICs.

These results were encouraging but not sufficient to prove the performance of *proFIA*, since the *centWave* algorithm was not designed for such data. We therefore assessed next whether *proFIA* is sufficiently robust for routine preprocessing of FIA-HRMS data sets.

7.2 Reproducibility of peak picking with *proFIA*

The reproducibility of *proFIA* was evaluated both in terms of quantification and detection on the **serFusion** and **serExactive** datasets. We first counted for each feature f the number of triplicate samples where f was detected. The distribution of the number of features corresponding to 1, 2, or 3 replicate detections are plotted on Figure 7.2 for both datasets. A total of 63% (respectively 65%) of the features were detected in all three replicates of the **serFusion**(respectively **serExactive**) dataset. The numbers dropped to 18% (respectively 16%) in 2 of the triplicates, and 19% (respectively 16%) in one of the triplicates only.

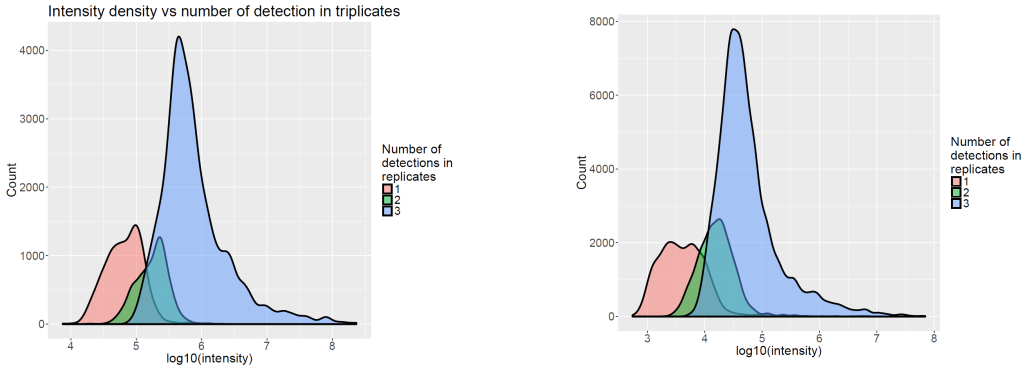


Figure 7.2: **Reproducibility of peak detection with *proFIA***. Distributions of the maximum intensity for the features detected in 1, 2, or all sample triplicates, in the **serFusion** (a) and the **serExactive** (b) datasets. Zero values were removed as they indicate that a feature is not present in the sample.

As expected, the most intense features were usually detected in all triplicates, whereas all features detected only in a single replicate have lower intensities compared to those detected in 2 or 3 of the triplicates. Overall, these results highlight the high level of reproducibility of the peak detection from *proFIA*.

The reproducibility of quantification was then investigated by computing, for each intensity quintile, the average of the CVs between replicates (Table 7.3). Overall, proFIA achieved a good quantification reproducibility of 12% (respectively 9%) for the **serFusion** (respectively the **serExactive**) datasets (Table 7.3). Such a performance is similar to the preprocessing of LC-MS data with the initial *matchedFilter* algorithm from *XCMS* (C. Smith et al. 2006).

Quintile	Intensity range	$CV_F(\%)$	$CV_E(\%)$
1^{st}	$F : [1.0 \times 10^4, 9.2 \times 10^4]$	16.1	
	$E : [9.9 \times 10^2, 6.0 \times 10^3]$		10.0
2^{nd}	$F : [9.1 \times 10^4, 1.4 \times 10^5]$	13.1	
	$E : [6.0 \times 10^3, 9.8 \times 10^3]$		9.1
3^{rd}	$F : [1.4 \times 10^5, 2.1 \times 10^5]$	11.6	
	$E : [9.8 \times 10^3, 1.5 \times 10^4]$		8.7
4^{th}	$F : [2.1 \times 10^5, 4.2 \times 10^5]$	10.9	
	$E : [1.5 \times 10^4, 3.0 \times 10^4]$		9.2
5^{th}	$F : [4.2 \times 10^5, 7.8 \times 10^7]$	7.7	
	$E : [3.0 \times 10^4, 2.3 \times 10^7]$		9.1

Table 7.3: **Reproducibility of *proFIA* quantification**. The coefficient of variation (CV) between the triplicates is computed within five consecutive intensity windows (quintiles) along the dynamic range for the **serFusion**(F) and **serExactive**(E) Orbitrap datasets.

7.3 Comparison with manual measurement

proFIA was compared to the classical "manual" peak-picking method routinely used by experimenters for the processing of FIA data. To this end, our partners from the LEMM laboratory at CEA, who had produced the **serFusion** dataset (human serum, either pure or spiked with 40 molecules at five increasing concentrations, analyzed in triplicate), used the *Xcalibur* vendor software (Thermo Fisher Scientific) to: 1) visualize the EICs of the 40 signals from each of the 18 samples, 2) evaluate whether a signal was detected or not, and 3) determine visually the time limits of the peak for subsequent integration by *Xcalibur*.

7.3.1 Comparison of detection

The detection results for the $18 \times 40 = 720$ signals achieved by the automated (*proFIA*) and "manual" methods are shown as a confusion matrix (table 7.4).

With a recall of 98.5% and a precision of 96.8%, *proFIA* results were in very good agreement with manual peak picking. In addition, they were obtained in a couple of minutes compared to 1 hour of work using manual integration.

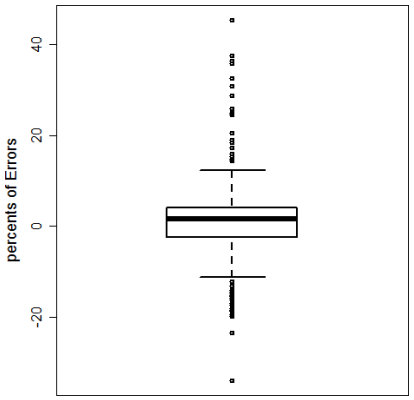


Figure 7.3: **Relative absolute differences of quantification between proFIA and manual integration**

		proFIA		
		Yes	No	Total
Manual	Yes	460	6	466
	No	15	239	254
	Total	475	245	720

Table 7.4: **Comparison of *proFIA* versus manual detection (serFusion** dataset).

7.3.2 Comparison of quantification

The quantification values from both methods are visualized as a heat map (Figure 7.4). *proFIA* intensities were very close to manual integration (the mean of the relative absolute differences was 5.1%).

To further understand the few discrepancies between the two methods, the 60 EICs with a difference above 10% (Figure 7.3) were further checked visually. Three types of discrepancies were observed, and only the first one was a limitation from *proFIA* which was corrected. First, for peaks with a strong right tail, the upper time limit of the peak often differed between *proFIA* and manual integration (leading to a difference of intensity). Therefore, to optimize the processing of these EICs, an additional parameter, $fullInteg$ was included in *proFIA* to allow the integration along the full time axis in the cases where no solvent has been detected on the considered EIC. Second, in the case of EICs affected by solvent, the integration with Xcalibur resulted in automatic removal of a flat baseline. The assumption of such a flat baseline, however, is not valid in the case of FIA-MS data (therefore this difference of quantification cannot be considered as an error from *proFIA*). Third, in some cases, some unexplained noise of high intensity was included in the integrated area by the chemist and not by *proFIA*.

Altogether, the previous results show that *proFIA* outperforms the state of the art LC-MS software in terms of peak picking, has a high reproducibility in both detection and quantification, and outperforms manual integration by an expert chemist. This highlights the unique value of *proFIA* among the existing MS preprocessing algorithms for metabolomics. A final study was conducted to assess the impact of the few *proFIA* parameters on the quality of the peak-picking.

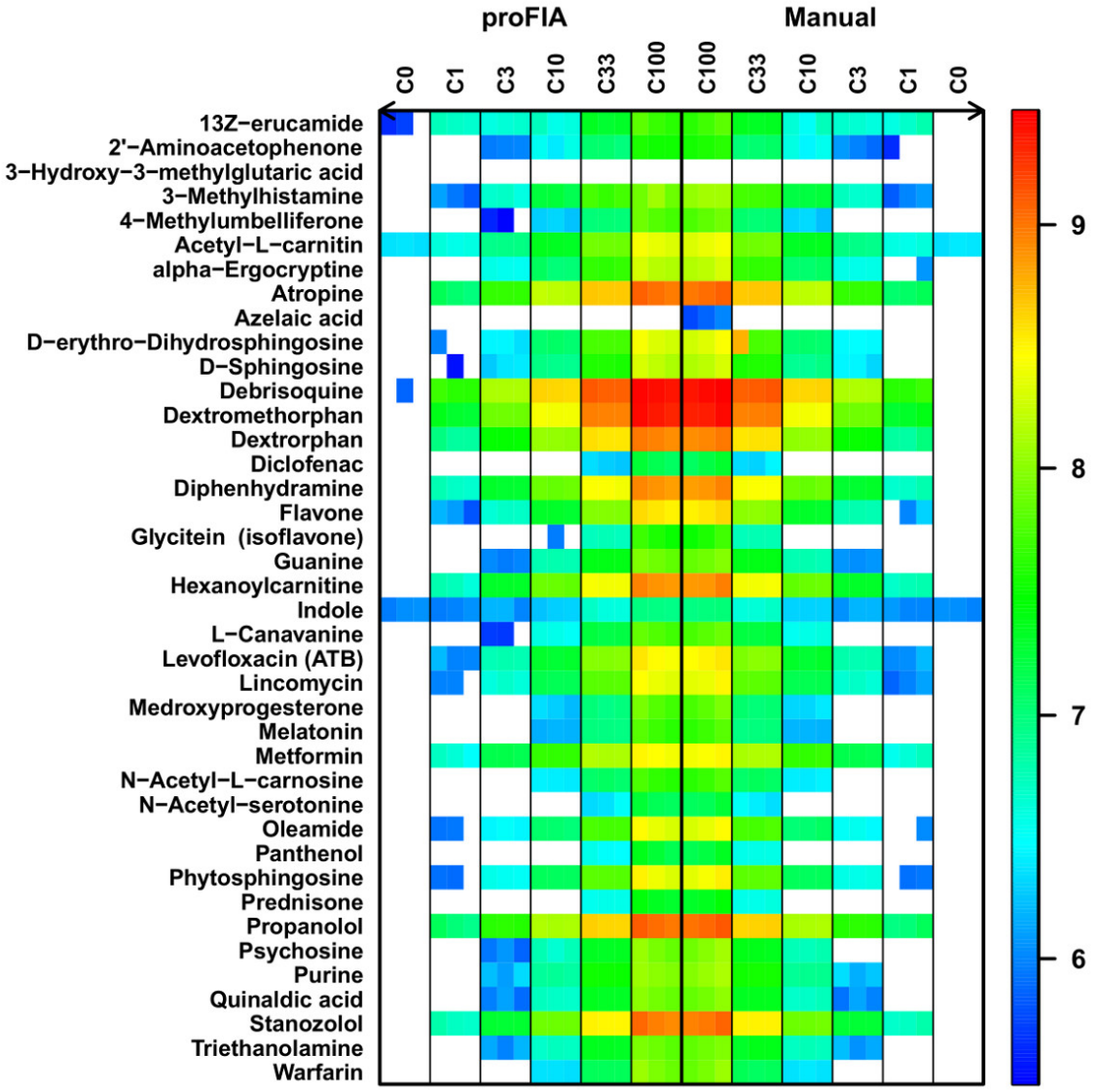
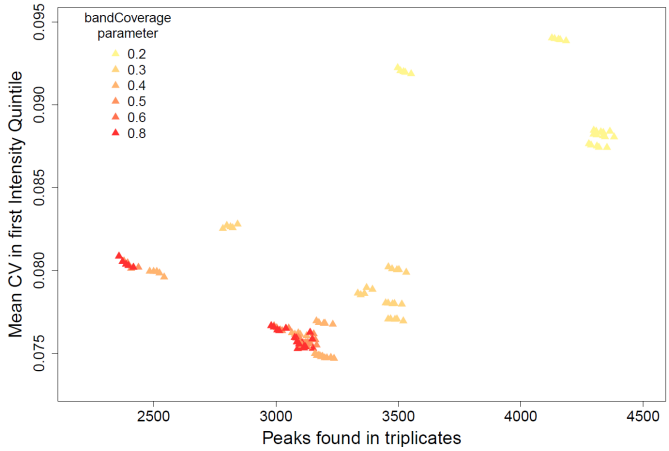


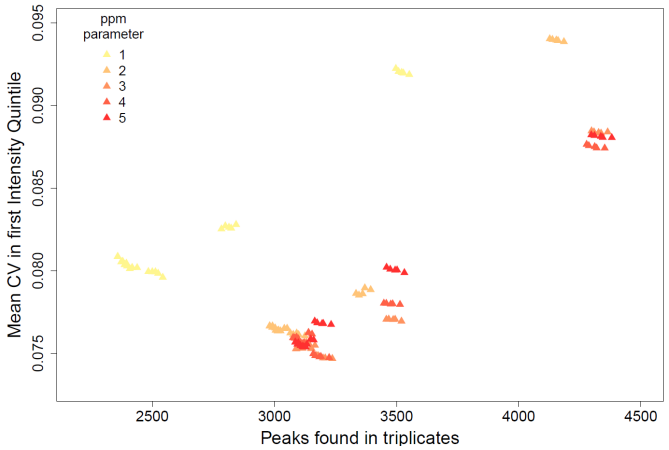
Figure 7.4: **Evaluation of peak detection and quantification on serFusion.** For each of the 40 compounds spiked in the serum samples at various dilutions and analyzed in triplicate, the automated quantification with *proFIA* (left) was compared to the manual integration of peaks with the vendor software (right). The concentration of the spiking mixture is indicated in the sample label (C1 = 10 ng/mL). The white color denotes the absence of signal.

7.4 Impact of *proFIA* parameter values

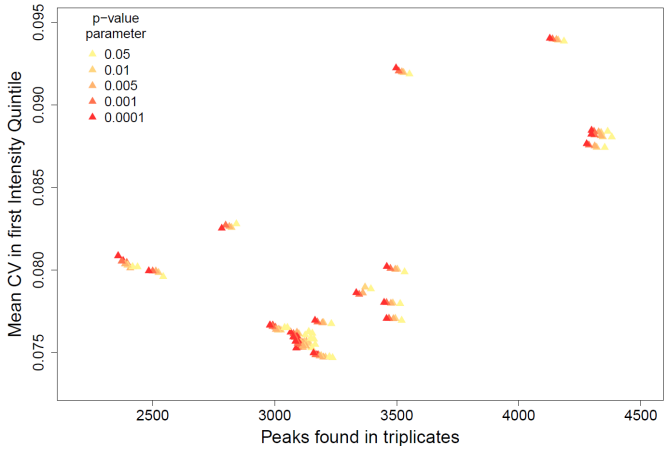
The impact of the most important parameters from Table 7.2 on the sensitivity (number of detected peaks) and reproducibility (CV of intensity) was analyzed. Figure 7.5 shows the results of the two metrics for all combinations of parameters described previously (Table 7.1): on each sub-figure, the values of one of the parameters is color-coded.



(a) *bandCoverage*



(b) *ppm*



(c) *pvalthresh*

Figure 7.5: **Influence of *proFIA* main parameters on reproducibility and sensitivity (serFusion)**. On each figure, the points are colored according to the values of one of the parameter.

The detection was impacted by *ppm* values below the accuracy of the mass spectrometer (e.g., < 1 *ppm*), but was otherwise robust to this parameter (Figure 7.5b). As expected, higher values of $bandCoverage$ lead to an increase of reproducibility but a decrease of detection (Figure 7.5a). The strong increase of the CV criterion when $bandCoverage$ decreases from 0.3 to 0.2 suggests that a value of 0.2 results in too many noisy peaks for this dataset. Finally, $pvalthresh$, which is the threshold used to filter out the features whose intensity is not significantly higher than the solvent baseline (6.5.3), has little influence on the reproducibility of signal detection and quantification since most of the bands containing solvent have been discarded during the band detection step (Figure 7.5a).

While *proFIA* parameters are of crucial importance for high-quality preprocessing (as for any peak-picking software), it is important to note that thanks to the concept of sample peak, *proFIA* does not have an equivalent to the signal-to-noise ratio parameter in the general case. This is due to the fact that the presence of signal in the sample peak only is usually sufficient to ensure that the signal originates form the samples. Therefore the number of parameters of the *proFIA* peak detection is reduced compared other to LC-MS peak-picking algorithms (Table 7.2).

Chapter 8

Discussion

This part of the thesis was devoted to the processing of Flow Injection Analysis coupled to High-Resolution Mass Spectrometry (FIA-HRMS) data. We have first built a specific model for the Extracted Ion Chromatogram (EIC) in chapter 5, which we have then used to design a workflow for feature extraction from raw data (chapter 6). Experiments showed that our *proFIA* software offered a good reproducibility of peak picking, and outperformed the reference manual peak-picking (i.e., with a similar accuracy but at meast 20 times faster; chapter 7). Using the EIC model, additional characterization of the features are provided to the user, such as an estimation of the sample peak and an indicator of matrix effect (ME).

Future improvements and addition of new features are discussed below, such as the annotation of adducts and neutral losses (section 8.1), the application to medium resolution data (section 8.2.1), the optimization of the ME indicator metric (section 8.3), and the benchmarking of alternative processing methods (section 8.4).

8.1 Intra sample grouping

In MS experiments, a single metabolite often generates multiple m/z signals, such as natural isotopes, or adducts and neutral losses generated in the ESI source (Keller et al. 2008). Annotation of these signals, which are highly correlated, would be very interesting both for the chemist (as a help in identification) and for the statistician (e.g., to reduce the number of univariate tests and hence the impact of the correction for multiple testing). However, no such annotation methodology currently exists for

FIA-HRMS signals. On the one hand, isotopes are easily detected because of their known mass difference with the mono-isotopic ion. Moreover, in metabolomics, the most intense peak always corresponds to the mono-isotopic signal (in contrast to proteomics). As a result, isotope annotation has sometimes been included in the mass traces detection process (Kenar et al. 2014), or using a second round of targeted detection of region of interests with relaxed parameters (Treutler and Neumann 2016). On the other hand, annotation of adducts and neutral losses is more challenging since their presence 1) is not known in advance and 2) depends on the molecular formula. An algorithm for the annotation of *pseudo spectra* grouping the multiple signals originating from a single molecule has been proposed in the *CAMERA* software for LC-MS data (Kuhl et al. 2012): peaks are first grouped according to their retention time; then a similarity metric is computed by taking into account the shape of the peaks and the m/z differences corresponding to known adducts; finally, this similarity is used to build a network, and co-eluting compounds are separated by label propagation.

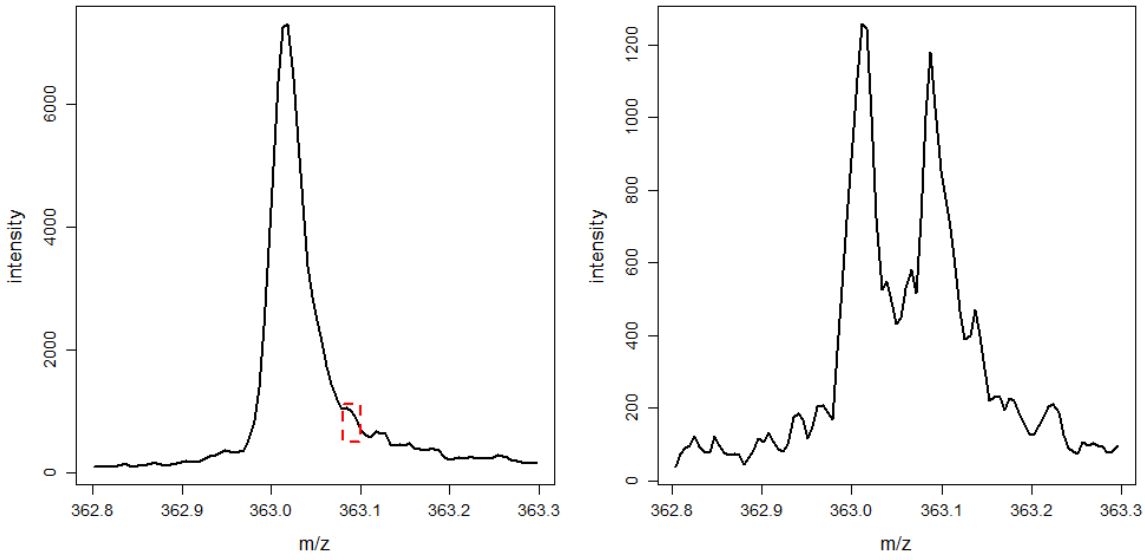
The *CAMERA* workflow is based on the fact that adducts and neutral losses occur after the chromatographic steps: as a result, peaks originating from the same compound are expected to have a similar shape. This assumption, however, does not hold in FIA-HRMS, since a lot of peaks have a shape similar to the sample peak. Moreover the density of signals detected in FIA-HRMS prevents a direct annotation of adducts using mass differences. Finally the impact of ME on the peak shape from adducts is not well-described in the context of FIA-HRMS. Therefore there is a need to include annotation strategies in *proFIA*, using principles specific to FIA-HRMS. One approach would be to study the similarity of EIC shapes among signals originating from the same compound. A second strategy would be to group all the annotation tasks into a global identification step downstream from *proFIA*: in their publication, Draper et al. (John Draper et al. 2009) derived rules from metabolite databases to predict adduct formation. However, since these approaches use a knowledge developed on LC-MS data, they cannot be directly transposed to FIA-HRMS (for example, the formation of adducts in the case of a strong ME is different from the LC-MS context).

8.2 Current limitations from *proFIA*

8.2.1 Medium resolution data

In the previous section we have seen the application of **proFIA** to two different datasets with m/z resolutions equal to 500 K and 100 K, respectively. *proFIA* was

also successfully to process data at a 60 K resolution (Habchi et al. 2017). Since all these high-resolution datasets have been acquired on Orbitrap instruments, we decided to assess *proFIA* performances on alternative, Time-Of-Flight (TOF), mass spectrometers. During the PhD, a very interesting high-throughput dataset relying on FIA-HRMS using a TOF was published (Sévin et al. 2016) and the dataset was made publicly available (MTBLS373). Processing of these data with *proFIA*, however, gave poor results: many of the signals described in the study were not detected by our workflow. Visual inspection of the results indicated that the main cause was a failure of the centroidization algorithm due to the lower resolution of the acquisition (about 5 K at 200 m/z), as shown on Figure 8.1: in Figure 8.1a, only the peak with m/z 363.02 is clearly visible. The second peak at m/z 363.09 on Figure 8.1b is only visible as a small shoulder peak in the red box of 8.1a, which prevents the detection of the centroid by usual peak picking method. This results in missing points in the sample peak, and eventually in the discarding of m/z bands by the *bandCoverage* filter.



(a) scan at the apex of the sample peak (b) scan at the border of the sample peak

Figure 8.1: **Example of overlap between isobaric ions.** Both spectra belong to the same sample extracted from Sévin et al. 2016, at acquisition times corresponding to either the apex (a) or the border of the sample peak (b). Two m/z features are visible on b). However, the feature of highest m/z, which has a much lower intensity (compare the y-axis scales), is not detected on a) (small shoulder peak highlighted with a red dashed rectangle).

To process these kind of data, optimized centroidization algorithms performing a decomposition of the overlapping signals are required, oppositely to those described in the introduction (4.2.1).

8.3 Refinement of *proFIA* workflow

The FIA-MS method of acquisition has two main limitations: the inability to separate isomers and the ME. While the former is inherent to the FIA technique, some analytical protocols have been shown successful in reducing the latter, such as sample dilution. Therefore a good characterization of ME is critical to achieve an optimal sensibility in an FIA-MS experiment. *proFIA* is to our knowledge the first workflow to integrate ME into the EIC model, and to provide a metric of ME. Here we discuss the quality of the ME measurement and the potential effect of ME on the ionization process.

8.3.1 Bias of the ME's indicator

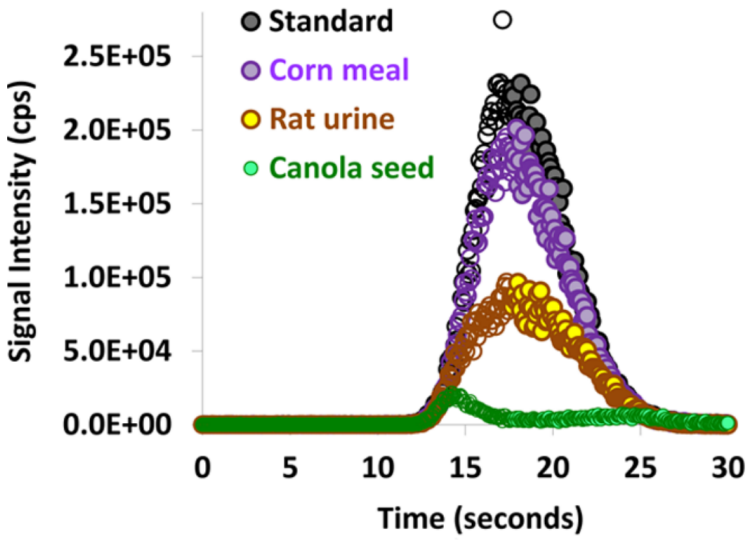


Figure 8.2: **Matrix effect on an analyte signal (Nanita 2013)**. All peaks correspond to the same analyte (Hexazinone), at the same concentration (1 ng/mL), either pure (black dots), or spiked into various biological matrices (colored dots). The severe ME observed with canola seed is likely due to high oil levels (Nanita 2013).

The *proFIA* indicator of ME is computed as the correlation between the EIC and the sample peak. This indicator, however, does not reflect quantitatively ME, in particular for strong effects resulting in the suppression of more that 80% of the signal intensity (Figure 5.6a). A better estimation is possible by using replicates and relying on the fact that ME at the borders of the peak is supposed to be negligible compared to the ME at the apex. As standard metabolomics methodology recommend the use of replicates, a refined version of proFIA may use such information to better estimate ME: a regression could be performed simultaneously on matched EICs from replicates

to determine the extent of ME while considering the sample peak of each sample.

8.3.2 Improvement of regression process

Section 6.4 described the procedure for the determination of the sample peak. While this procedure has been designed using simulations in section 6.4.1, this validation is not sufficient to ensure that the correct sample peak is picked. A validation of this sample peak could be done by using the method proposed by Nanita 2013 on the EICs selected for regression by *proFIA*. The second issue with this regression procedure is the hypothesis of the presence of well-behaved EICs. Such an assumption is difficult to verify in the general case, and is probably sample dependent. While it is clear that some molecules ionize better than others, it is likely that all EICs exhibit significant matrix at high concentration of samples. Therefore the estimated sample peak is probably less sharp than the real sample peak. The estimated sample peak however is closer to the real sample peak than the TIC or any of the individual EICs. These considerations were taken into account in *proFIA*, as the estimation of the sample peak is only used as a model for the matched filtration which is known to be robust to error in peak width.

Estimation of the sample peak could be further refined. In particular, the EMG model may not be able to describe all the sample peak shapes generated by the FIA process. As an example, a peak with a flat top was observed in the dataset from Sévin et al. 2016. Additionally, the regression could also be performed by using likelihood maximization and the estimation of the noise variance described in section 6.3.1.

The use of replicates and the optimization of regression may eventually allow to apply our FIA peak model to every EIC, therefore providing unique quantitative information about ME.

8.4 Extension of the *proFIA* software

The alternative processing method described in section 4.1 and used successfully at large scale in Sévin et al. 2016 based on statistical filters was shown to be efficient for lower resolution data (8,000-12,000; Fuhrer et al. 2011; Sévin et al. 2016). In contrast, such a resolution is too low for the currently used centroidization methods to allows an efficient processing by *proFIA* (Section 8.2.1). Integration of these alternative algorithms into *proFIA* (their current implementation is not publicly available) would

allow to further extend the scope of *proFIA* for high-throughput applications.

Part III

Development of a tool for structural similarity mining: MineMS2

Approach

The second part of the PhD is devoted to the structural annotation of compounds by mass spectrometry, which is a major challenge in metabolomics. To fully exploit the structural information contained in large collections from MS/MS spectra, we developed a suite of algorithms to mine fragmentation patterns. The resulting workflow is implemented in the MineMS2 R/C++ package. At the time of writing this thesis, the last step (selection of the top k patterns of interest) is still being optimized. MineMS2, however, has already been validated on two metabolomic case studies.

This part is structured as follows. In Section 9.4, we first review the computational strategies to analyze MS/MS spectra, to either predict the structure (e.g. using *in silico* fragmentation methods) or mine the similarities between spectra (e.g. with molecular networks or the detection of motifs). Focusing on the latter approach, we then introduce the algorithms for Frequent Subgraph Mining (FSM; useful definitions are provided in Section 9.5), which are at the core of our strategy (later described in Chapter 11). We conclude this introduction by presenting the preprocessing methods we developed to obtain high-quality MS/MS spectra as input from our pattern mining workflow (Section 9.9).

In Chapter 10, we define a new representation of MS/MS spectra as Losses Graphs without the need for prior knowledge of the elemental formulas. The algorithm developed to build these graphs is detailed in Section 10.2. Finally, one formal hypothesis about the Losses Graphs and some resulting properties which will be used in Chapter 11 are introduced (Section 10.3).

In Chapter 11, we design an FSM algorithm to specifically mine the Losses Graphs Ṡince such algorithms are computationally intensive, several strategies to reduce the search space are implemented: we first demonstrate that the subgraphs of interest belong to the class of Acyclic Flow Graphs (AFGs; Section 11.1.1). We further show that, instead of full AFGs, trees can be used (Section 11.1.2). We therefore define a new data structure, the k-Path Tree(Section 11.1.3), to generate all frequent AFGs(Section 11.2). This set of AFGsis large. However, we show that only closed AFGsshould be mined (Section 11.3). Finally, the performance of the proposed suite of algorithms is evaluated on two real datasets (Section 11.5). The results highlight one of the main challenges of exact FSM approaches: the high number of generated patterns. In Section 12.1, we therefore propose an innovative method to reduce the number of patterns, based on the reduction of the Hasse Diagram associated to the set of closed patterns.

Chapter 9

Introduction

In this introduction we first give an overview of the main uses of MS/MS spectra, and of the existing identification strategies. We start from database matching, we then present *in silico* fragmentation, and finally describe similarity-based methods. In a second part, we introduce the Frequent Subgraph Mining (FSM) algorithms and principles, which are a basis of the proposed strategy.

The measure of an m/z ratio alone in an MS spectrum is not sufficient to identify metabolites, which are characterized by a broad diversity of chemical structures: even with an accuracy inferior to 1 ppm, multiple elemental compositions are possible for molecules above 200 Da (Kind and Fiehn 2007). As a result, additional information is needed for metabolite identification. While retention time (RT) may be used for the characterization of isomers between MS spectra from the same analytical platform, the chromatographic process is difficult to reproduce on a distinct setting, preventing the use of RT in public databases. In contrast, fragmentation spectra (MS/MS) provide unique structural information and are therefore well-suited for metabolites identification.

The first intuitive use of MS/MS spectra is the direct comparison with spectral databases containing fragmentation spectra of known compounds.

9.1 MS/MS spectral database matching

The direct identification method relies on a spectral database and a matching algorithm. Public spectral databases differ according to the species (e.g., the human

metabolome database HMDB; David S Wishart et al. 2018), or the class of metabolites (e.g., the lipid database LipidMap; Fahy et al. 2007). Databases with a more general scope also exist as MassBank (Hisayuki Horai et al. 2010) and GNPS (M. Wang et al. 2016), but are often less curated. Curation is critical for the quality of the spectra: MassBank relies on an automatic processing implemented in the RMassBank package (Stravs et al. 2012), while other databases such as METLIN or MzCloud use manual curation, which can be considered as more robust. A recent review points out that the coverage of the MS/MS public databases is still limited (Vinaixa et al. 2016): even compounds centric databases such as METLIN or HMDB contain spectra from only $5-10\%$ of the known metabolites, which prevents the successful matching of the majority of unknown MS/MS spectra. Moreover there is a lack of MS/MS spectra acquired in the negative ionization mode in all the considered databases.

The second issue of current spectral database queries is the matching algorithm (Vinaixa et al. 2016). A filter on the m/z of the precursor ion is performed to generate a list of candidate MS/MS spectra: however, there is no simple procedure for the next step which is the actual matching of MS/MS peaks. Despite their critical importance, matching algorithms are scarcely described in the majority of the databases listed in Vinaixa et al. 2016: among the cited methods are the normalized dot product implemented in MassBank (Hisayuki Horai et al. 2010) and the X-rank scoring implemented in METLIN (Mylonas et al. 2009). In the past, alternative approaches have also been described, such as the Euclidean distance and a Probability Based matching algorithm (Pesyna et al. 1976).

9.1.1 Cosine Similarity

The cosine similarity have been initially proposed in 1978 (Sokolovv et al. 1978) and was evaluated as the best performing method for MS/MS spectra comparisons in Stein and D. R. Scott 1994: the two MS/MS spectra to compare (say U and V) are represented as two vectors of the same dimension $N_{U\cap V}$, which is the number of common peaks between U and V (as defined by a tolerance in the m/z dimension provided by the user). For each vector, each element is a number depending of the m/z and the intensity of the peak (see below). The similarity metric is the cosine of the angle between the 2 vectors:

$$Cos(U,V) = \frac{\sum_{i\in U\cap V} W_{U,i}W_{V,i}}{\sqrt{\sum_{i\in U\cap V} W_{U,i}^2}\sqrt{\sum_{i\in U\cap V} W_{V,i}^2}}$$

with $W_i = (m/z)_i^n intensity_i^m$. The exponents n and m are used respectively to increase the contributions of peaks with low intensity, and with high masses (the latter being more specific that common fragments; Stein and D. R. Scott 1994). The $n = 3$ and $m = 0.6$ values used in MassBank have been determined empirically (H. Horai et al. 2008).

Furthermore, this metric has been refined as a *composite* cosine score to take into account all the peaks from the query spectrum (and not just the common peaks; Stein and D. R. Scott 1994):

$$CompCos(U, V) = \frac{N_U Cos(U, V) + N_{U \cap V} F(U, V)}{N_U + N_{U \cap V}}$$

N_U represents the number of peaks in the query spectrum U, and $N_{U \cap V}$ the number of peaks common to U and V. F is a factor capturing a similarity (of trend between the two spectra:

$$F = \frac{1}{N_{U \cap V}} \sum_{i \in U \cap V} \underbrace{\left(\frac{W_{U,i} W_{V,i-1}}{W_{U,i-1} W_{V,i}} \right)^{k_i}}_{Y_i}$$

with k_i defined as:

$$k_i = \begin{cases} 1 & if\ Y_i < 1 \\ -1 & otherwise \end{cases}$$

The composite distance is implemented in MassBank (Hisayuki Horai et al. 2010) and a modified version is used in GNPS (M. Wang et al. 2016). As highlighted recently (Scheubert et al. 2017), the tuning of the cosine similarity parameters is critical to increase the number of annotations (up to 5 fold).

Because of the limited coverage of MS/MS spectral databases (spectra from 13,000 distinct standards in METLIN and 28,000 spectra in MassBank; Vinaixa et al. 2016), a new type of annotation methods has emerged in the last decade, which are based on the matching to candidate spectra generated *in silico* by modeling the fragmentation of compounds. These methods have the advantage to work with huge molecular databases such as PubChem (Kim et al. 2016), ChemSpider (Pence and Williams 2010), ChEBI (Hastings et al. 2016), and KEGG (Kanehisa et al. 2017). These databases contain 27 millions, 25 millions, 46,000 and 14,000 compounds, respectively, which cover a wide range of chemical classes and biological matrices. Importantly, *in silico* fragmentation methods allow identification of compounds for which no standard is available.

9.2 *in silico* fragmentation methods

In this part we briefly describe the common principles and the main differences between the most-used freely available *in silico* fragmentation software: MetFrag (Wolf et al. 2010), MS-Finder (Tsugawa, Kind, et al. 2016), CSI-FingerID (Shen et al. 2014) and CFM-ID (F. Allen et al. 2014). Two types of strategies have been used: machine learning approaches of fragmentation processes based on spectral databases (CFM-ID, CSI-FingerID), and methods relying on physical criteria to select the best spectrum among all candidates generated by combinatorial fragmentation (MetFrag, MS-Finder). A common feature of all these methods is the determination of an underlying fragmentation graph, with the complete molecule as the root and the fragments as vertices.

9.2.1 Machine-learning based approaches

Such methods learn a model from an MS/MS spectral database and the corresponding structures of the parent molecules. The two chosen model-based methods both model the fragmentation of a molecule using a graph. However while CFM-ID considers the graph of the precursor molecule to build the model, CSI-FingerID only considers the formula of the molecules.

CFM-ID first breaks all the bonds combinatorially while allowing some rearrangements, and then models the probability of transition from one fragment (node) to the other. To reduce the search space and generalize the model, each fragment is represented as a binary fingerprint ϕ including information about the neighboring atoms, the presence of a specific chemical group, or the presence of a fixed 2 or 3 paths. The interested reader is referred to F. Allen et al. 2014 for the full description of the fingerprint. The model can then be run forward to predict a spectrum from a compound as a mixture of Gaussians.

Alternatively, CSI-FingerID (Shen et al. 2014) first determines a fragmentation tree for each parent molecule of the database, by selecting the best fitting set of sub-formula from a set of candidates to explain the experimental spectra. The underlying method, which relies on solving an instance of the maximally weighted colored subtree, has been extensively studied by the team of S. Böcker (Böcker and Rasche 2008; Rasche et al. 2010; Böcker and Dührkop 2016). A combined kernel based on both the fragmentation tree and the spectrum is computed, and subsequently used to train a Support Vector Machine (SVM) to learn a binary fingerprint representing the molecular graph. This

directly maps the spectrum and the fragmentation tree to each feature of the fingerprint. For each query spectrum, the fragmentation tree is computed and the chemical features are predicted by the SVM model, and for each molecules a score is computed using these features. This model has been extended to bypass the fingerprint by using Input Output Kernel Regression in Brouard et al. 2016.

9.2.2 Physic-based approaches

Here, the fragmentation graph of each candidate molecule is obtained by the systematic *in silico* breaking of all bonds between the atoms within the chemical structure (or pairs of bonds in case of rings). The final score takes into account not only the matching of the simulated fragments to the measured MS/MS peaks but also a physical criterion: in MS-Finder, priority is given to fragments that are obtained according to known chemical rules (Tsugawa, Cajka, et al. 2015), whereas MetFrag relies on the bond dissociation energy (Wolf et al. 2010). In these combinatorial approaches, the choice of the fragmentation path is important (e.g., in MetFrag, the shortest path is selected).

Several of the *in silico* fragmentation methods have been refined by using isotopic patterns, such as MetFrag (Ruttkies et al. 2016) and CSI-FingerID (Shen et al. 2014) (the latter relies on isotopic patterns to compute the fragmentation tree with the SIRIUS software Böcker, Letzel, et al. 2009).

9.2.3 Comparison of the *in silico* fragmentation methods

While the presented methods are based on distinct principles, they all have the same objective: the annotation of an MS/MS spectrum using *in silico* fragmentation of candidates from compound databases. In 2012, S. Neumann and E. Schymanski proposed to benchmark the approaches during an annual challenge: the Critical Assessment of Small Molecule Identification (CASMI, Schymanski and Neumann 2013): each year, an organizing team provides MS/MS spectra (i.e., *challenges*) and the goal for the participants from the Category 2 is "to determine the correct molecular structure using *in silico* fragmentation techniques alone".

Interestingly, recent results (Schymanski, Ruttkies, et al. 2017) highlight the complementarity of the different approaches: in the 2016 contest, CSI-FingerID (Shen et al. 2014) and one of his derivative (Brouard et al. 2016) got the highest number of gold medals (awarded for each spectrum to the contestant with the lowest rank of the

	Positive mode			Negative mode		
	Baseline performance	+Element filter	+Known formula	Baseline performance	+Element filter	+Known formula
Small database (HMDB, MassBank, ChEBI, NIST14) Correct in **TOP 1**						
FingerScorer	37.5 ± 3.5	44.5 ± 3.5 (+7.0)	46.0 ± 3.0 (+8.5)	32.5 ± 4.5	37.5 ± 4.5 (+5.0)	38.5 ± 4.5 (+6.0)
FragScorer	50.0 ± 2.0	54.0 ± 2.0 (+4.0)	56.0 ± 2.0 (+6.0)	44.0 ± 3.0	45.5 ± 3.5 (+1.5)	47.0 ± 4.0 (+3.0)
FingerScorer&FragScorer	49.0 ± 2.0	52.0 ± 2.0 (+3.0)	53.0 ± 2.0 (+4.0)	42.5 ± 3.5	45.0 ± 3.0 (+2.5)	46.0 ± 3.0 (+3.5)
MetFrag	51.7 ± 0.0	51.7 ± 0.0 (+0.0)	51.7 ± 0.0 (+0.0)	45.0 ± 0.0	45.0 ± 0.0 (+0.0)	45.0 ± 0.0 (+0.0)
CFM-ID[a]	41.0 ± 4.0	47.5 ± 3.5 (+6.5)	48.5 ± 2.5 (+7.5)	36.5 ± 4.5	41.5 ± 3.5 (+5.0)	42.5 ± 3.5 (+6.0)
CSI:FingerID	N/A	42.5 ± 2.5	45.0 ± 3.0 (+2.5)	N/A	38.0 ± 5.0	39.0 ± 5.0 (+1.0)
Small database (HMDB, MassBank, ChEBI, NIST) Correct in **TOP 5**						
FingerScorer	80.0 ± 3.0	83.0 ± 3.0 (+3.0)	83.5 ± 2.5 (+3.5)	71.5 ± 5.5	74.5 ± 5.5 (+3.0)	75.0 ± 5.0 (+3.5)
FragScorer	82.5 ± 2.5	84.0 ± 2.0 (+1.5)	84.5 ± 2.5 (+2.0)	76.5 ± 4.5	77.5 ± 4.5 (+1.0)	77.5 ± 4.5 (+1.0)
FingerScorer&FragScorer	86.0 ± 2.0	87.0 ± 2.0 (+1.0)	87.0 ± 2.0 (+1.0)	78.0 ± 4.0	80.0 ± 4.0 (+2.0)	80.0 ± 4.0 (+2.0)
MetFrag	86.5 ± 0.0	88.6 ± 0.0 (+2.1)	89.1 ± 0.0 (+2.6)	83.5 ± 0.0	85.6 ± 0.0 (+2.1)	85.6 ± 0.0 (+2.1)
CFM-ID[a]	76.5 ± 3.5	80.5 ± 3.5 (+4.0)	82.0 ± 2.0 (+5.5)	68.0 ± 7.0	74.0 ± 2.0 (+6.0)	74.0 ± 2.0 (+6.0)
CSI:FingerID	N/A	76.0 ± 3.0	76.5 ± 2.5 (+0.5)	N/A	69.0 ± 5.0	69.5 ± 4.5 (+0.5)
Large database (PubChem, HMDB, MassBank, ChEBI) Correct in **TOP 20**						
FingerScorer	48.5 ± 4.5	52.0 ± 5.0 (+3.5)	56.5 ± 5.5 (+8.0)	47.0 ± 7.0	52.5 ± 8.5 (+5.5)	56.0 ± 9.0 (+9.0)
FragScorer	26.5 ± 1.5	34.5 ± 1.5 (+8.0)	41.5 ± 1.5 (+15.0)	36.5 ± 2.5	45.0 ± 4.0 (+8.5)	45.5 ± 3.5 (+9.0)
FingerScorer&FragScorer	45.0 ± 3.0	49.5 ± 3.5 (+4.5)	52.5 ± 3.5 (+7.5)	50.5 ± 7.5	56.0 ± 8.0 (+5.5)	58.5 ± 8.5 (+8.0)
MetFrag	32.8 ± 0.0	41.4 ± 0.0 (+8.6)	50.0 ± 0.0 (+17.2)	41.2 ± 0.0	48.5 ± 0.0 (+7.3)	50.2 ± 0.0 (+9.0)
CFM-ID[a]	23.0 ± 1.0	31.5 ± 1.5 (+8.5)	37.5 ± 1.5 (+14.6)	30.0 ± 0.0	34.5 ± 0.5 (+4.5)	38.0 ± 1.0 (+8.0)
CSI:FingerID	N/A	34.5 ± 4.5	38.0 ± 5.0 (+3.5)	N/A	41.5 ± 6.5	43.5 ± 6.5 (+2.0)

Table 9.1: **Retrieving rates of *in silico* fragmentation (Laponogov et al. 2018).**

true molecule), respectively 86 and 82 out of 208, followed by MS-FINDER (Tsugawa, Cajka, et al. 2015) and CFM-ID (F. Allen et al. 2014), with 70 and 63 gold medals, respectively. However, MS-FINDER outperformed the model based methods regarding the mean rank of the correct molecule.

In a very recent publication, Lapogonov et al.(Laponogov et al. 2018) proposed to combine both approaches. To do so they defined a composite score TotalScore, based on a score similar to MetFrag(FragScore) score and a score similar to the CSI-FingerID(FingerScore). The proposed methodology named ChemDistiller was compared to CSI-FingerID, CFM-ID and MetFrag. Performances of the approaches were evaluated on the number of correct molecules within the top 1 and top 5 candidates for a small database (HMDB, MassBank, ChEBI, NIST), and within the top 20 for a large database (PubChem, HMDB, MassBank, ChEBI) on 6,297 compounds (Table 9.1). The combined approach outperforms the other methods on the large database, but not on the small one where the correct molecule is better ranked among the predictions from MetFrag. Moreover physics based MetFrag always outperforms the machine learning based CFM-ID and CSI:FingerID (Table 9.1), which is not in agreement with the results from CASMI. However, it should be noted that the machine learning based methods inherently include more parameters, which results in a more complex tuning. In contrast, the physics based methods are simpler to use. In the CASMI context, algorithms are generally tuned by the teams which developed them, thereby providing a high level of expertise. This may explain their higher performances, however it is

not the typical level of expertise of the end-user of these approaches.

Another important conclusion from this study is that, on large databases, even with a known elemental formula, no software exceed 56.5% of correct ranking of the true chemical formula within the top 20 predictions. This may be explained by the very high number of candidates in the compound databases and the resulting complexity of selecting the unique correct identification. Therefore, new methodologies are still required to address the challenge of metabolite identification.

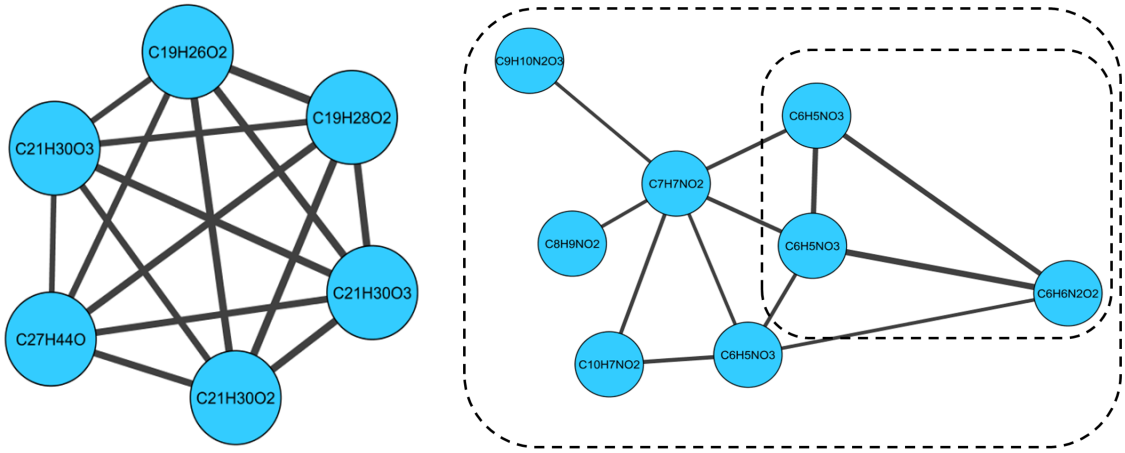
9.3 Similarities based approaches

In parallel to the prediction of chemical structure, alternative unsupervised strategies have been developed recently to extract structural information from MS/MS spectra by studying their similarities. The idea is that the annotation from one known spectrum may be propagated to similar spectra. Two types of approaches have been described: MS/MS networks implemented in *GNPS* (M. Wang et al. 2016) and a pattern based *MS2LDA* method (Hooft, Wandy, Barrett, et al. 2016).

9.3.1 MS/MS similarity network

In such networks, each node is a spectrum and each edge indicates a high similarity between two spectra: in the Global Natural Products Social molecular network (*GNPS*), an edge corresponds to a cosine similarity above a user-defined threshold and at least k peaks in common (the value of k is also set by the user M. Wang et al. 2016). *GNPS* networks have been used to propagate annotation and visualize the main metabolites components (Boya P. et al. 2017; Hooft, Padmanabhan, et al. 2016). Another implementation of such MS/MS networks is available in the *MetCirc* package (Naake and Gaquerel 2017). Furthermore, to increase the readability, a filtering step is often added: in *GNPS*, displayed vertices have at most n edges, where n is selected by the user.

An important conceptual limitation of molecular networks, pointed out by Hooft, Wandy, Barrett, et al. 2016 is shown on Figure 9.1a: even though the complete graph suggests a common structural information between the spectra (set of shared peaks or losses), this component cannot be automatically extracted from the pairwise comparisons carried out by the network. Furthermore, the characterization of clusters within a given network is difficult (Figure 9.1b). As a result, a new kind of approaches has



(a) (b)

Figure 9.1: **Limitations of the interpretation of molecular networks (LemmDB data set)**. a): a six node cluster extracted from the network generated by *GNPS* (M. Wang et al. 2016); b): two distinct levels of interpretation (clusters) within a network are displayed as black boxes.

emerged, based on pattern mining.

9.4 MS/MS pattern mining

The objective of pattern mining methods is to extract building blocks from a set of spectra, where each block corresponds to a given molecular family or to a molecular substructure. These blocks are generally a set of fragments and/or losses. A first example is the MS2LDA approach (Hooft, Wandy, Barrett, et al. 2016), which relies on Latent Dirichlet Allocation modeling (Blei et al. 2003), originally used in text mining. Briefly, within a set of MS/MS spectra to be studied, each spectrum is seen as a mixture from a set of topics (i.e. the patterns) common to these spectra, where each topic is a distribution from the set of common words (i.e. the m/z of losses). Recently, MS2LDA has been extended to simultaneously mine multiple sets of spectra (Hooft, Wandy, Young, et al. 2017). Although the MS2LDA approach is very innovative, our experience suggests that the parametrization of the model is quite complex, often leading to patterns generated mainly by a single very common word, such as an H_2O loss.

A second strategy, based on frequent itemset mining, has been described very recently on bioRxiv (Mrzic et al. 2017). In this approach named MESSAR, the set of all the mass differences between the fragment ions is computed, and considered as an

item-set representative of the spectrum. Then, for a set of spectra, the corresponding set of itemsets is used to perform associative rule mining to predict structural features. It is to my knowledge the most similar approach to the one proposed in this PhD.

A major difference between in-silico fragmentation approaches and the similarities based approaches presented in this section is the fact that the former rely on graphs to model the fragmentation process, in contrast to the latter. Graphs are an intuitive representations of chemical fragmentation (in particular when elemental formulas are associated to the nodes), and are therefore highly meaningful for MS experts. For these reasons, we have developed a new pattern mining approach of MS/MS spectra collections based on a graph representation of the spectra. Chapter 10 defines this representation which may be computed without the knowledge of the elemental formula. Chapter 11 describes our *MineMS2* algorithm based on Frequent Subgraph Mining (FSM) to extract recurring subgraphs from this set of graphs. In the rest of this introduction, a more technical overview of the main algorithms for FSM will be given.

9.5 Graph Theory Reminder

This part gives some definitions and notations about the Graph Theory and the more specific Frequent Subgraph Mining concepts. The interested reader is referred to J. L. Gross et al. 2013 for the graph theory part and to Jiang et al. 2013 for the frequent subgraph mining related part.

In this part we will focus on **directed edge-labeled graphs**. Such a graph is a 4-tuple $G = (V, E, L, l)$, with:

- V the set of vertices.
- $E \subseteq V \times V$ a set of ordered pairs.
- L the set of edge labels
- $l : E \to L$ the edge-labeling function.

The set of vertices, edges, and the edge-labeling function of a graph G will be denoted $V(G)$, $E(G)$, and l_G, respectively. In the rest of this section, we will only consider edge-labeled graphs so for commodity purpose the term labeled is omitted.

For each edge $e = (u, v)$, u is called the **source** of the edge and v is called the **target** of the edge.

A **Directed Acyclic Graph (DAG)**, is a finite directed graph with no directed cycle, meaning that there is no consistently directed sequence of edges from a vertex v which loops back to v.

A **directed path** is a sequence of alternating edges and vertices in which the source of the edge is preceding the edge and the target of the edge is following it. By definition all the vertices are disjoints.

A **rooted graph** is a graph in which one vertex r has been distinguished as the of **root**. If there is a directed path from r to any other vertex of the graph, the graph is called a **flow graph**. If there is a unique path from the root to any vertex, it is called

an **arborescence**.

A graph G is a **subgraph** of a graph H if $V(G) \subset V(H)$ and $E(G) \subset E(H)$ and $\forall (u,v) \in E(G), u \in V(H)$ and $v \in V(H)$.

A **spanning arborescence** T of a graph G is a subgraph of G containing all the nodes of G and such that T is an arborescence.

An **isomorphism** between two graphs G and G' is a bijective function $f : V(G) \to V(G')$ such that:

$$\forall (u,v) \in E, (f(u), f(v)) \in E' \text{ and } l_G(u,v) = l_{G'}(f(u), f(v))$$

This relation will be denoted $G \simeq_e G'$ in this thesis.

The fact that G is **isomorph to a subgraph** H' of H will be denoted as $G \subseteq_e H$. We will also call H a supergraph of G. The isomorphism from $V(G)$ to $V(H')$ is called an **embedding** of G in H. We write the set of all the embeddings of G in H as ϕ_G^H

We define the **support** of a graph G on a set of graphs D :

$$Supp_D(G) = \sum_{H \in D} |\phi_G^H|$$

This definition counts the multiples occurrences of subgraph in a graph multiple times, oppositely to the most common definition. The frequency of a graph may be defined as $Freq_D(G) = Supp_D(G)/|D|$

The **Frequent Subgraph Mining(FSM)** is the task to extract all the subgraphs G from a set of graphs with support superior to a fixed threshold ϵ. In practice one often limits the mining to connected subgraphs.

A **lattice** is a partially ordered set in which every two elements have a unique supremum (also called a least upper bound or join) and a unique infimum (also called a greatest lower bound or meet). In the case of subgraph, the ordering relation is the subgraph relationship $\subseteq_e$.

This lattice can be visualized as a graph, where each vertex represents a subgraph from a graph of D. The lowest vertex is the empty graph, while the top vertices are the graphs from D. A vertex p is a parent of a vertex q on the lattice, if the subgraph associated to q is a subgraph p with a single edge or node added. An example of such a lattice limited to connected subgraph and ordered by the subgraph relationship is shown in Figure 9.2.

Figure 9.2 highlights one of the main problem of FSM, the multiple way to construct a graph. This is often tackled by considering a specific **canonical form** of a graph. If two graphs G and H are isomorphic, they have the same canonical form. We define a **canonization function** as a function associating its canonical form to a graph.

9.6 Introduction to Frequent subgraph Mining

The objective of **Frequent Subgraph Mining (FSM)** algorithms is to extract all the subgraphs from a given data set with a frequency above a specified threshold. A good review of FSM algorithms can be found in Jiang et al. 2013. The definitions and vocabulary used in this part are defined in section 9.5. Their main fields of application are chemistry (notably the mining of molecular substructure), web, and biology (notably to mine protein-protein interactions; Jiang et al. 2013). FSM problems have been divided into two classes, depending on whether the frequent subgraphs are mined within a set of medium-size graphs (often called "transactions"), or within a single very large graph. Moreover one may want to mine all or only a subset from the frequent subgraphs, and allow some errors in the subgraph matching.

In this part we focus on the mining of all the frequent subgraphs from a set (or database) of graphs D. The frequency of G in D is $Freq_D(G) = |Supp_D(G)|/|D|$, where $Supp_D$ is the support defined in section 9.5. FSM corresponds to the extraction of all the subgraphs G from D with a frequency above an ϵ threshold provided by the user:

$$\mathcal{F}_D = \{G | Freq_D(G) \geq \epsilon\}$$

FSM algorithms require two elements: the generation of a set of candidate graphs, and the computation of their support. This raises one of the core problem of Frequent Subgraph, the computation of subgraph isomorphism.

9.6.1 Graph isomorphism problem

Here we give a short description of the problem of graph isomorphism. Interested readers can refer to J. Lee et al. 2012 for a recent review of algorithms for subgraph isomorphism within a database, and to Mckay and Piperno 2014 and Ullmann 1976 for a description of the principles of such algorithms and their comparison. In Chapter 10 we will describe the two dedicated algorithms we developed to solve the subgraph isomorphism in quadratic time and the graph isomorphism in linear time on our specific graphs.

While the general problem of subgraph isomorphism is known to be NP-complete (Garey and D. S. Johnson 1979), there is no proof of the NP-completeness of graph isomorphism. Efficient algorithms have however been proposed for graph isomorphism,

using a branch and bound approach (Ullmann 1976), and expanded notably later in the VF2 algorithm (Cordella et al. 2001). Algorithms usually include a partial mapping between the two graphs, and have specific strategies to prune the search space and refine this mapping. More recently, algorithms which are better suited to the research of a subgraph within a set of graphs have been implemented (e.g. by storing the presence of specific graph features among the set of graphs), the most efficient being quickSI (Shang et al. 2008), as shown by J. Lee et al. 2012. Polynomial algorithms are also available in the case of specific graphs, for example trees (Shamir and Tsur 1999) or planar graphs (Eppstein 1995).

While the algorithms detecting subgraph isomorphism may also be used for graph isomorphism, some more efficient strategies have been described, based on the canonical labeling of a graph. A group theory based approach was first used in NAUTY (McKay et al. 1981) and further extended into Traces software (Mckay and Piperno 2014). These algorithms are based on group theory to compute the graph automorphism group, which is then used to generate a canonical form of the considered graph. These automorphisms allow in practice a huge pruning of the search space, and similar principles have been reused in multiple software aiming at the testing of graph isomorphism (Junttila and Kaski 2011; Darga et al. 2004).

In practice, FSM algorithms often bypass a lot of the isomorphism computation burden by generating graphs in a specific fashion. For example, the SLEUTH algorithm, which mines unordered subtrees, relies on a scope list to avoid any graph isomorphism calculation (M. J. Zaki 2005).

9.7 Overview of FSM algorithms

Because of the tremendous size of the search space (which consists of all the subgraphs within a given database), FSM algorithms have focused on different ways to reduce this space, in particular by traversing it in different ways (Section 9.7.1), or by using specific canonical forms to ensure that each element of the search space is visited only once (Sections 9.7.2 and 9.7.3). The most important components of FSM algorithms will be presented hereafter; readers interested to a more detailed description or a more exhaustive list of FSM algorithms are referred to Jiang et al. 2013 and Ayed et al. 2016.

9.7.1 Traversal of the search space

The FSM algorithms share some common principles: they consider a set of subgraphs, and try to make them grow (Jiang et al. 2013). When a single subgraph of size is considered and grown as much as possible, this process is referred as Depth-First Search (DFS) exploration as it corresponds to a DFS traversal of the lattice. When multiple graphs of size k are combined to give a set of graphs of size $k+1$, this approach is referred as an Apriori approach, and the lattice is traversed in a Breadth-First manner. These approaches are similar in essence to the methods initially developed in Frequent Itemset Mining (i.e., the Apriori algorithm (Agrawal and Srikant 1994) and the FP-growth algorithm (Han et al. 2000)). In contrast, GASTON is another type of algorithm which traverses the lattice in a modified BFS strategy, by first considering the frequent paths, then by combining them to build the trees, and finally by aggregating these trees to build the full set of frequent graphs (Nijssen and Kok 2004). Another paradigm is implemented into the MARGIN algorithm, which moves among the nodes of the search space at the *border* between frequent and non-frequent subgraphs (Thomas et al. 2006).

All the approaches makes use of the **Downward-Closure Property** which states that if a graph is frequent then all of its subgraphs are also frequent (Jiang et al. 2013). By doing so, they inherently explore the *lattice* of all subgraphs from D (See section 9.5). An example of such a lattice is shown in Figure 9.2.

Figure 9.2 highlights some of the most salient features of such lattices. Firstly, the number of subgraphs is huge compared to the size of the database, and therefore a full exploration of the lattice is computationally intractable. Although the downward closure property of frequent subgraphs is a good way to prune some part of this lattice, it is often not sufficient to reduce its size. The difference between Apriori and DFS based approaches is the order used to explore this lattice (Figure 9.3). On the one hand, the Apriori methods require to store a lot of patterns at each step. However a large part of the search space can be pruned by combining only frequent patterns. On the other hand, DFS based approaches store a reduced number of graphs for the exploration process, but explore a wider part of the search space, as stated in Jiang et al. 2013. Secondly, there are multiple ways to generate a subgraph from a subgraph of lower size. Therefore to avoid to the duplication of subgraphs, a **canonical form** is used.

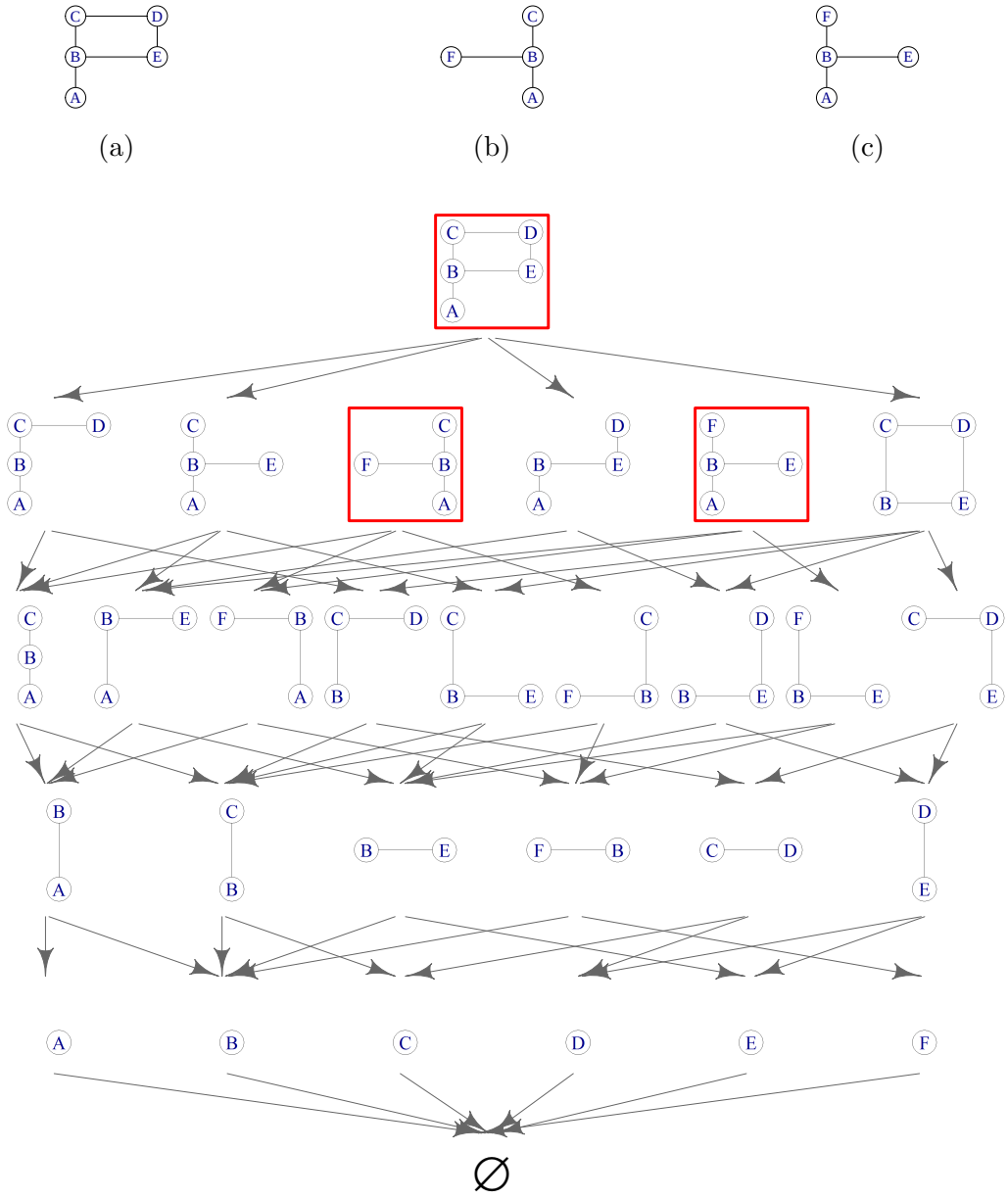
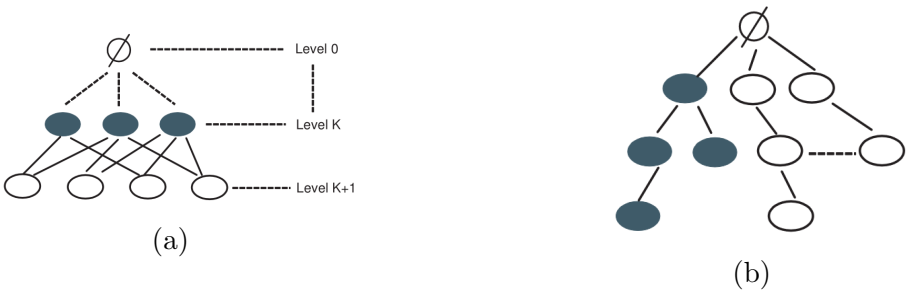


Figure 9.2: **Example of subgraph lattice.** 9.2d: lattice containing all the connected subgraphs from a database D consisting of 3 graphs a), b), and c), highlighted with red boxes.



(a) (b)

Figure 9.3: **Apriori and DFS strategies to explore a lattice of subgraphs (from Jiang et al. 2013)**

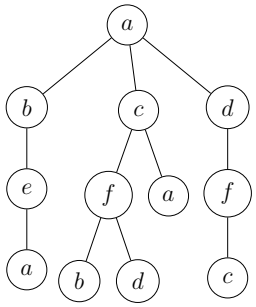
9.7.2 Canonical forms in FSM

A canonical form is a standard way to represent an object, in the case of graphs, it is a representation which is the same for all the isomorphic graphs. A common strategy for graph canonization is to generate a labeling of the graph which corresponds to a given ordering of its vertices of the graphs, and then to define the canonical form as the minimum of these labelings using the lexicographic order among the permutations of vertices. Such a method has been implemented to define a minimum DFS code in gSpan (Han et al. 2000), or to define the minimum Canonical Adjacency Matrix code in AGM (Inokuchi et al. 2000), or maximum code in FFSM (J. Huan et al. 2003). These canonical forms have been used because they often incorporate an interesting property, allowing a huge pruning of the search space: if G is a subgraph of H then $code(G) < code(H)$. GASTON uses the Nauty software to compute the canonical form of the detected cyclic graphs (Nijssen and Kok 2004).

Canonical forms however are easier to compute for trees, especially for rooted trees. They are usually based on two kinds of tree traversal, depth-first search (DFS) or breadth first search (BFS), with a specific symbol added when the algorithm needs to backtrack (Figure 9.4). Canonical forms are generally the root of the candidate generation step, and they are checked each time a candidate is generated in many generic algorithms.

9.7.3 Candidate generation

Two kinds of methods exist to generate candidates depending of the exploration of the lattice, DFS or Apriori. Both of these methods aim at minimizing the number of redundant subgraphs generated. The first extension strategy combines two size k frequent patterns to build a new candidate. To avoid duplicated candidates, these



(a)

Name	Canonical Form
DFS Label Sequence	$abea\$\$\$cfb\$d\$\$a\$\$dfc\$\$\$$
Depth-Label Sequence	$(0,a),(1,b),(2,e),(3,a),(1,c),(2,f),$ $(3,b),(3,d),(2,a),(1,d),(2,f),(3,c)$
Breadth-First Canonical String	$a\$bcd\$e\$fa\$f\$a\$bd\$\$c\#$

(b)

Figure 9.4: **Canonical forms of a tree used in FSM algorithms.** Example of a tree (a) and several of his canonical forms used in FSM (b). A full description of these labeling algorithms is available in Jiang et al. 2013

methods often rely on an ordering based on a canonical form of the chosen graphs. Such approaches have been introduced into graph mining by the Apriori algorithm (Inokuchi et al. 2000). Suites of algorithms have been designed to break down the complexity of the task, starting by the generation of frequent paths, then combining them to build trees, and finally graphs, as implemented in GASTON (Nijssen and Kok 2004). A similar approach is described in Gudes et al. 2006, which merges paths into graphs, and then recursively generates the full set of graphs.

Another example of Apriori approach, with a reduced number of patterns stored at each level, is described for trees in Mohammed J. Zaki 2002. At each step, each tree of size $k+1$ is generated from two trees of size k which share a similar canonical code except on the last character. Subtrees meeting this requirement are grouped into a structure called equivalence class, allowing an efficient storage and candidate generation (Mohammed J. Zaki 2002; Mohammed J. Zaki 2004).

In the second kind of methods based on DFS, a single edge is added to the data, without combining existing patterns. Such a method is used for example in the gSpan algorithm (Han et al. 2000).

In the case of rooted trees with ordered labels, there is an efficient enumeration order, called right-most extension, which ensures to visit each tree only once (Figure 9.4): given a frequent subtree T, the tree may only be extended by adding a node at the right of the right-most path. In practice this implies that the canonical labels such as the **DFS-Label Sequence** (see Table 9.4b) from the previous trees are a prefix from the canonical label of the extended tree. Right-most extension is used to generate candidates in the majority of the tree mining software (Mohammed J. Zaki 2002; M. J. Zaki 2005; Chi et al. 2004).

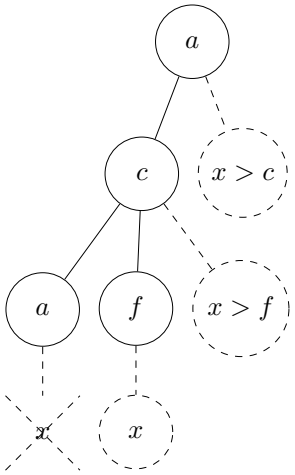


Figure 9.5: **Right most extensions of a tree.** Possible extensions are shown as dashed lines. $x > c$ label indicates that only labels superior to c are allowed. The node which is crossed out is forbidden because it is not on the right most path

While the candidate generation of unconstrained graphs normally requires to check for isomorphisms, right most-path extension strategies ensure each graph is uniquely constructed, therefore sparing many computations. However, an issue is that many non frequent candidates are generated. As a result, a new scheme of extension for trees based on extension of the left and right most leaf has been proposed in the AMIOT algorithm (Hido and Kawano 2005).

Candidate generation strategies require to make a compromise between the generation of each candidate unique time, and the generation of frequent candidates over non-frequent candidates.

The final step is the computation of the supports from the subgraphs. Raw isomorphism methods may be used, as in gSpan (Yan and Han 2002). However, in many software, embedding lists are stored to reduce the computation burden. When mining rooted trees, it is also possible to store only the roots of the subtrees, therefore reducing the storage required. A stated in Jiang et al. 2013, exact algorithms for subgraph enumeration have been mainly developed between 2000 and 2007. Recent researches have often focused on using these algorithms in a distributed way, for example using the map reduce paradigm and an a prior algorithm in Bhuiyan and Hasan 2015.

9.8 Reducing the number of mined patterns

Once built, the full lattice of frequent patterns is often very large, and the information content is too large and redundant to be interpreted directly. Therefore a special class of algorithms have been designed to mine maximal and closed subgraphs. Let us denote F_D the set of frequent patterns from a graph database D, then the set of maximal frequent subgraphs MF_D may be defined as:

$$MF_D = (G|G \in F_D \text{ and } \nexists H \in F_D | G \subseteq H)$$

Two algorithms have been proposed to efficiently mine Maximal Frequent Subgraphs, SPIN (Jun Huan et al. 2004) and MARGIN (Thomas et al. 2006). SPIN uses a specific type of equivalence class to generate the patterns, namely graphs which share a minimum spanning tree code. Such graphs are locally maximal in these equivalence classes, which results in an efficient pruning of non maximal candidates. MARGIN is based on the property that a subgraph is maximal if all its possible supergraphs in the database D are not frequent. In contrast to all the other algorithms which traverse the lattice in a bottom-up manner, MARGIN traverses the lattice at the interface between frequent and non frequent patterns, therefore detecting maximal patterns. These two algorithms have proven to be generally faster than general FSM algorithms, and to generate less candidates.

While MF_D is several orders of magnitude smaller than F_D. However if a low support threshold is used, it will miss a large part of the pattern with high support. As a result, another set of patterns have often been mined: the set of closed patterns CF_D:

$$CF_D = (G|G \in F_D \text{ and } \nexists H \in F_D | G \subseteq H \text{ and } Supp_D(G) = Supp_D(H))$$

The three sets of patterns satisfy: $MF_D \subset CF_D \subset F_D$. We will focus on closed patterns since they are more relevant to our application (as we shall see in Chapter 11). An algorithm to define CF_D named CloseGraph has has been designed by the authors of gSpan (Yan and Han 2003). CloseGraph is based on an early termination criterion, which states that, given a graph g and an edge e, if g and $g+e$ have the same support, then all the graphs including g without e should not be grown, and that growing the graphs containing $g + e$ is sufficient. Efficient checking of these property allows to reduce the search space. Alternatively, an algorithm mining complete graphs in the specific case of relational graphs was proposed by Yan, Zhou, et al. 2005. This

algorithm, however, is very specific to these kinds of graphs. More efficient approaches have been developed for other specific graphs, such as graphs with unique edge labels (El Islem Karabadji et al. 2016), or graphs without edge label but with unique node labels, which may be used to model biological networks (J. y. Peng et al. 2008). An algorithm for DAG mining has also be proposed (Termier et al. 2007), but it is limited to graphs with unique labels.

No recent comparison of these graph mining algorithms is available: in 2005, four FSM algorithms were compared, including gSpan and GASTON (Wörlein et al. 2005). This study notably highlights the exponential nature of the FSM mining problem. In particular, all the algorithms presented in this section consider a reduced number of vertices and of edge labels: comparisons for graph mining software is usually performed on molecules with less than 10 distinct edge labels, and less than 100 distinct node labels. Even with additional constraints such as unique edge labels (El Islem Karabadji et al. 2016) or without edge label but with unique node labels (J. y. Peng et al. 2008), the runtime increases exponentially and becomes intractable even with a reduced number of labels. These considerations highlight the critical need to simplify the problem as much as possible by reducing the search space. In Chapter 10, we derive a set of interesting properties from the lattice of MS/MS fragmentation subgraphs, which allows us to transform the graph problem into a frequent subtree mining issue addressed in Chapter 11.

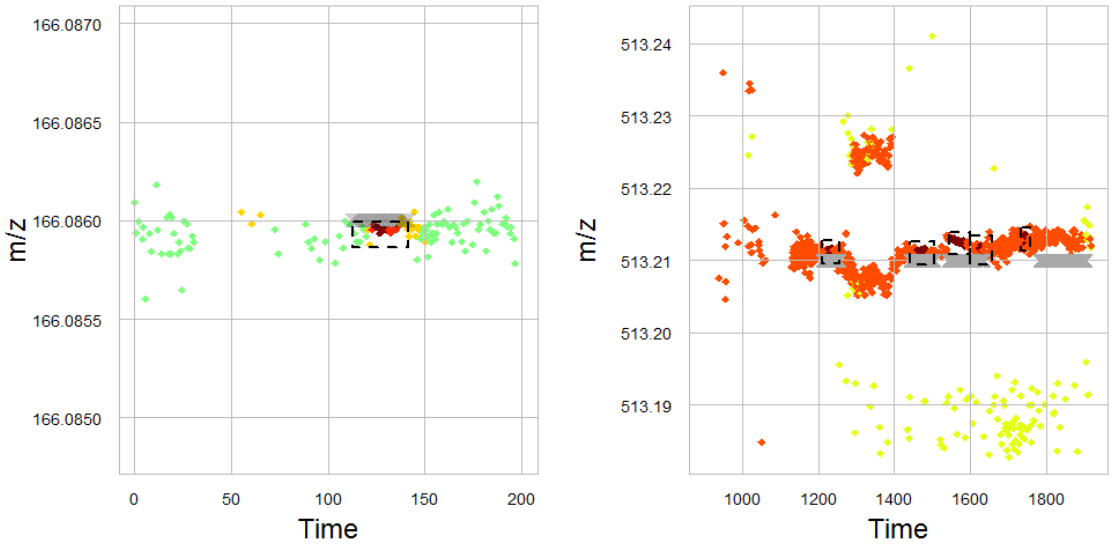
9.9 MS/MS spectra preprocessing

When building the MineMS2 workflow, we stumbled upon an unexpected issue: the preprocessing of MS/MS spectra. Preprocessing takes as input the raw data mixing MS and MS/MS scans, with multiple MS/MS acquisitions for a single precursor, and generates a single fragmentation spectrum for each precursor, while discarding as much noise as possible. Although many methods have been proposed to analyze MS/MS spectra, the preprocessing has been comparatively seldom described. This may be explained by a shift in the use, and performance, of the MS instruments. Historically, MS/MS spectra were targeted on a reduced set of molecules of (putative) biological interest. These spectra were then processed using vendor software in a targeted manner, either by manually selecting the set of raw MS/MS spectra and generate the single MS/MS spectrum, or by specifying an m/z and rt ranges to do so. However the rise of Data Dependant Acquisition has considerably increased the number of spectra, and as the considered precursors are not known in advance, this kind of processing has become increasingly difficult. Four main open-source softwares are currently able to preprocess such MS/MS data: MZmine (Pluskal et al. 2010), RMassBank(Stravs et al. 2012), MS-DIAL (Tsugawa, Cajka, et al. 2015), msPurity(Lawson et al. 2017). A summary of the intended use and the issues with these tools is provided in Table 9.2 (MS-DIAL was not included as we were not able to run it on any of our computers).

Software	Reference	Type	Intended MS/MS use	Limitation
MZmine	Pluskal et al. 2010	Java GUI and command line	Coupling with GNPS	Weak cleaning procedure Many noisy peaks remaining
RMassBank	Stravs et al. 2012	R package	Cleaning of spectra from known molecules	Requires knowledge of the formula of the molecules
msPurity	Lawson et al. 2017	R package	Link between MS and MS/MS spectra Evaluation of precursor purity	No direct output of MS/MS spectra, no cleaning and fusing of MS/MS spectra

Table 9.2: **Comparison of the main open-source MS/MS preprocessing software tools.**

As shown in Table 9.2, preprocessing with the existing solutions was not optimal as input for MineMS2: in particular, removal of noisy peaks (spectra cleaning) is crucial in our methodology, which does not use any intensity threshold, and does not prioritize high intensity peak. We therefore developed an R package, named *MS2process*, aimed at the preprocessing of MS coupled to MS/MS data: it takes as input a peak list (e.g. generated by XCMS or MZmine for LC-MS data, or by proFIA for FIA-MS data),



(a) (b)

Figure 9.6: **Matching of the selected MS/MS precursor and the detected MS feature.** Isolated precursors (gray crosses) and MS features detected by XCMS-centWave (black rectangles, extended for visualization purpose) are shown (Data Independent Acquisition of the **PenicilliumDIA** dataset). a): the specified precursor isolation window matches the detected MS feature; b): example of a significant shift between the isolated ions and the actual signal.

including the m/z and rt windows for each feature, and returns the set of cleaned MS/MS spectra in the reference Mascot Generic Format (.mgf).

The steps from the MS2process workflow were inspired by RMassBank (Stravs et al. 2012) :

- MS/MS and MS feature matching
- MS/MS spectra fusing
- MS/MS spectra filtering

The first step aims at matching the detected peaks in the m/z dimension to the precursor ions isolated by the mass spectrometer (Figure 9.6). This task is complex because significant deviations may be observed, especially in targeted acquisition, e.g. when the precursor m/z value is experimental (Figure 9.6a), or when multiple peaks have close m/z values (Figure 9.6b). Our algorithm therefore matches the m/z and rt of all precursor ions to the peak list of detected features, using a specified m/z tolerance and a strict matching in the time dimension.

Then, for each MS feature matching a precursor ion, only MS/MS spectra corresponding to scans of high intensity on the Extracted Ion Chromatogram are kept, since these spectra are less noisy (Figure 9.7): in practice, the selected scans have an intensity above $(1 + fwhmthresh) \times baseline$ on the EIC of the precursor ion (where $fwhmthresh$ is set to 0.33 by default, and $baseline$ is the maximum of intensity at the limits of the EIC peak.

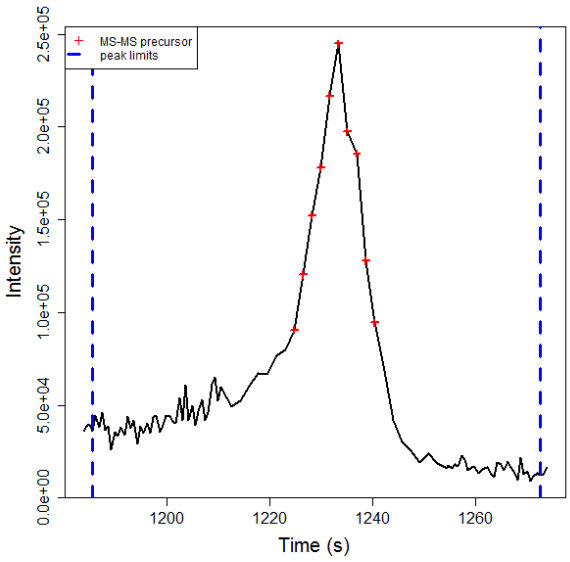
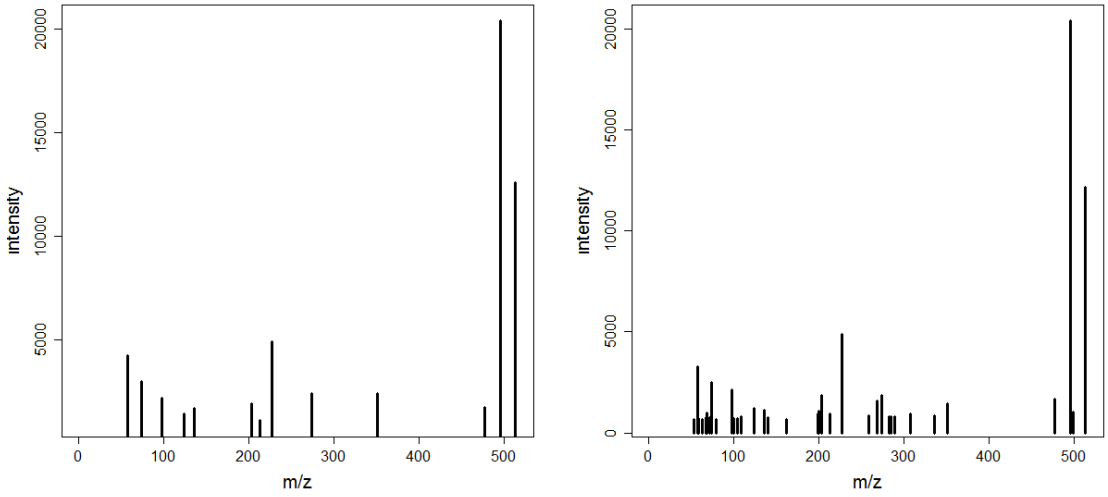


Figure 9.7: **Selection of MS/MS spectra on the MS EIC (PenicilliumDIA dataset).** Red cross: selected scans; blue lines: limits of the feature detected by centWave.

In addition, within the selected MS/MS spectra from each precursor, the peaks corresponding to the same fragment ion are grouped, by using a density based clustering similar to the one described in Section 6.6. The number of peak occurrences within each group is computed for quality control (see below). The intensity of the group is set to the mean of the peak intensities.

Finally, a quality control step is performed on the processed MS/MS spectra, by using two filters: 1) *multiplicity* (default set to 0.5): minimum frequency of the peak occurrence in the initial (selected) raw spectra; 2) *noiselevel* (default set to 0.5%): minimum relative intensity of the peak in the processed spectrum. In practice, these two filters remove a lot of noisy signals and drastically reduce the number of peaks as compared to the raw spectra (Figure 9.8).

At that stage, high quality MS/MS spectra are available. If the elemental formulas of the precursors are specified, a formula generator based on the algorithm by Böcker and Lipták 2005 may further be used to annotate the fragments.



(a) (b)

Figure 9.8: **Cleaning performed by MS2process.** a): output MS/MS spectrum generated by MS2process; b): raw MS/MS spectrum at the apex from the EIC (**PenicilliumDIA** dataset).

9.9.1 Comparison of the preprocessing from MS2process (automated) and Xcalibur (manual)

The automated preprocessing with MS2process was compared for 19 precursors from the **PenicilliumDIA** DIA dataset to the routine manual processing by chemists with the Xcalibur software (Thermo Fisher). A total of 40% of the peaks detected with Xcalibur were also selected by MS2process. The other 60% were found to be noisy peaks which were filtered out only with the MS2process software (*multiplicity* filter). This was confirmed by the fact that all the peaks with a relative intensity above 2% detected with Xcalibur were also selected by MS2process. Furthermore, there was less than 5% difference of peak intensity between the two software. We concluded that MS2process was at least equivalent to the manual spectrum extraction, and probably better suited for automatic workflows as it removes more irreproducible noisy peaks. The MS2process package has not been published yet due to time constraints (since this would require a comprehensive benchmark with the other tools available); it was used, however, for the MS/MS spectra extraction of both the **LemmDB** and the **PenicilliumDIA** datasets.

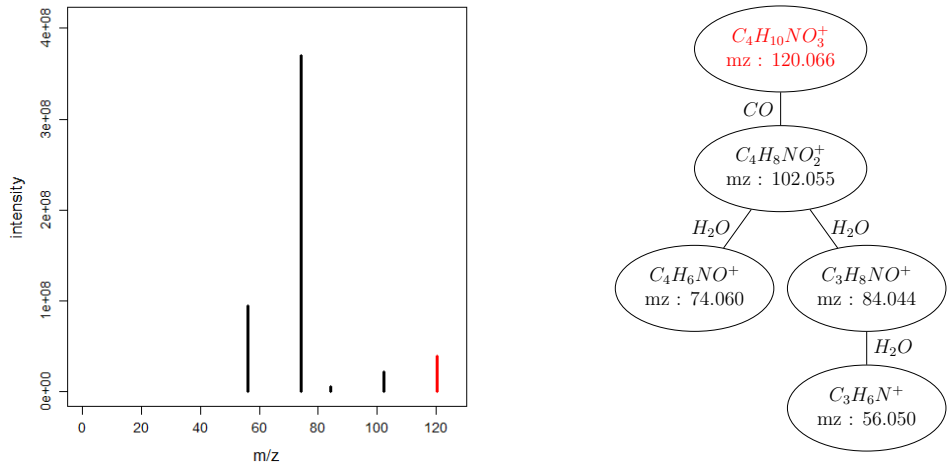
Chapter 10

Definition of a graph representation for a set of fragmentation spectra highlighting their structural similarities

We propose here a new representation of a collection of MS/MS spectra as a set of graphs named Losses Graphs , which highlights the similarities of fragmentation among the corresponding (unknown) molecules. Some interesting properties of these graphs are demonstrated, for further use in the mining of similarities in the next chapter.

10.1 Definition of a graph representation of a set of collisional spectra

We consider D a set of collisional spectra : each spectrum is the result of the fragmentation of a parent ion (or precursor) noted *prec*, of m/z ratio *prec.mz* and intensity *prec.int*. M is the elemental formula of *prec*. The spectrum is a 2D vector with dimensions named mz and int. We consider that the spectra are relatively noise free, we proposed in section 9.9 an adapted MS/MS spectra extraction and cleaning procedure, however MzMine(Pluskal et al. 2010) also furnishes spectra in the right format.



(a) (b)

Figure 10.1: **Representation of an MS/MS spectrum as a fragmentation tree.** a): spectrum from homoserine, $C_4H_9NO_3$, from the **LemmDB** dataset; b): corresponding fragmentation tree computed by the SIRIUS software (Böcker and Dührkop 2016). The precursor peak is shown in red.

10.1.1 Initial graph representation: Fragmentation tree

S. Böcker and F. Rasche(Böcker and Rasche 2008) initially proposed to represent an MS/MS spectrum as a fragmentation tree. The use of fragmentation tree is different from the fragmentation trees used to describe MS^n spectra(Kasper et al. 2012).

In the fragmentation trees from Böcker and Rasche 2008, vertices and edges correspond to fragments and neutral losses, respectively. In higher-energy collisional dissociation (HCD) cells from Orbitrap instruments, multiple consecutive fragmentations occur: at each fragmentation, an ion $M(+/-)$ (where M and $(+/-)$ are the elemental formula and the charge of the ion, respectively) leads to two fragments, one bearing the charge (A) and the other remaining neutral (B), as follows:

$$M(+/-) = A(+/-) + B \tag{10.1}$$

The charged fragment A will be observed as a peak on the spectrum, contrary to the neutral fragment B which is not detected by the instrument. B is therefore referred as a **neutral loss**. In the fragmentation tree, an edge labeled B is added between the vertices M and A. An MS/MS spectrum and its fragmentation tree representation is shown in Figure 10.1.

Determination of the true fragmentation tree is a very difficult task, which has been studied since 2008 by the team of S.Böcker: In Rasche et al. 2010 they showed that

only 71% of the predicted losses were confirmed by experts. While the computing of an optimal fragmentation tree has been improved in terms of speed an optimality by using Integer Liner Programming (Rauf et al. 2012), the rate of errors remains substantial. The rates of errors in the determination of the molecular formula at the root of the fragmentation tree and therefore of the fragmentation tree remains superior to 10% on different datasets in the most recent evaluation in Böcker and Dührkop 2016.

As errors in the fragmentation tree prevent the discovery of similarities between spectra, we defined a new kind of fragmentation graph which can be computed without the elemental formulas of the precursor ions in D.

10.1.2 A new graph representation of MS/MS spectra: the Losses Graphs

We propose to use Edge-Labeled Directed Graph (see 10.2) which we will abbreviate by **Losses Graph** . These graphs have a natural topological order, as vertices can be ordered by m/z values. Furthermore, the vertex corresponding to the precursor ion is always included. Each edge label corresponds to a m/z bin, resulting from a discretization from a set of m/z differences. In the case where this m/z bin is sufficiently small it mays be matched to a single formula (for example H_2O is the only neutral molecule with a nominal mass of 18) and can thus be labeled by the elemental formula. However multiple formula may often be matched to a bin, in this cases the corresponding edge label is shown as the mean of the corresponding m/z bin (Figure 10.2).

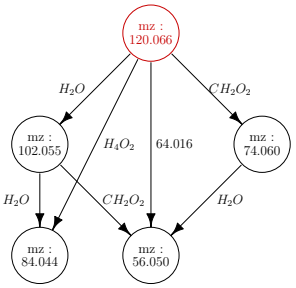
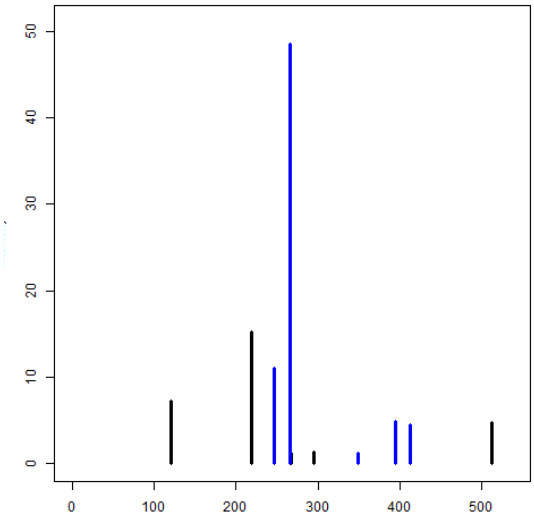


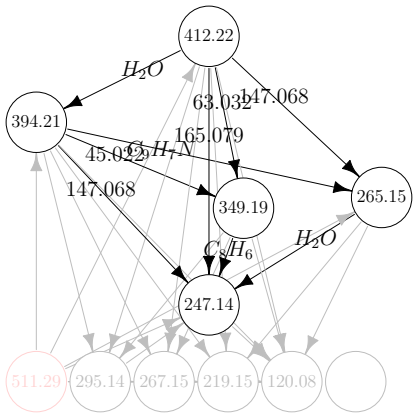
Figure 10.2: **Losses Graph representation from homoserine.** The MS/MS spectrum is showed in 10.1a. The red label correspond to the precursor ion.

Importantly, these graphs do not require any knowledge of the molecular formula, in contrast to the fragmentation trees described earlier: when present, formula labeling of an edge only indicates that a single elemental formula matches the m/z difference between the two considered peaks. Moreover these graphs does not use the intensities of the peaks on the MS/MS spectra.

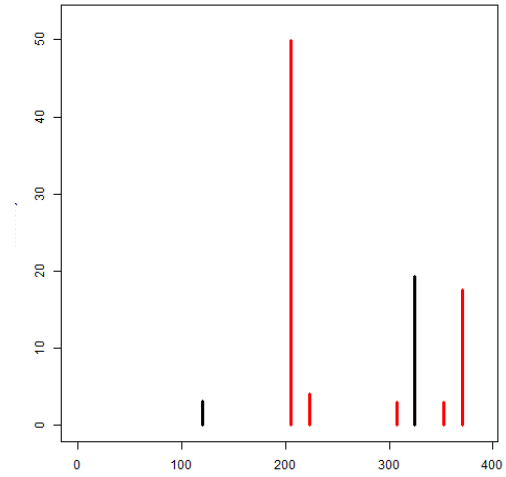
10.1.3 Interest of Losses Graph representation



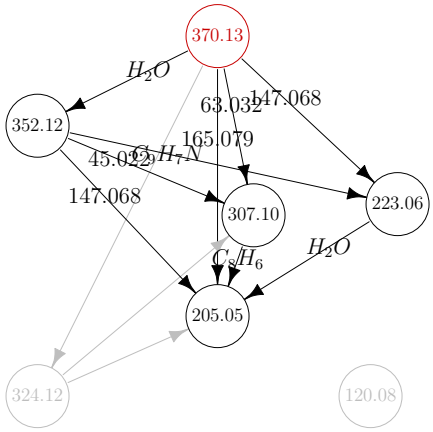
(a) precursor: m/z 511.29



(b) Losses Graph from (a)



(c) precursor: m/z 370.13



(d) Losses Graph from (c)

Figure 10.3: **A common fragmentation pattern between MS/MS spectra from two distinct precursors (PenicilliumDIA dataset)**, shown in a) and c). Only the 10 most intense peaks are displayed for visibility purposes. b) and d): **Losses Graph** from spectrum a) and c), respectively. The red vertex corresponds to the peak of the precursor ion. A common subgraph is highlighted in bold in the **Losses Graph** , and the corresponding peaks are colored in each MS/MS spectrum.

The proposed **Losses Graph** representation enables to detect fragmentation similarities even in the case where they originate from different precursors. As an example, Figure 10.3 displays the **Losses Graph** from two spectra from the **PenicilliumDIA** dataset where a common subgraph has been detected; however, only in the second subgraph does the root correspond to the peak of the precursor ion of the MS/MS

spectrum (m/z 370.13). The mining of **Losses Graph** therefore provides major structural insights into the similarities of the fragmentation patterns, which cannot be obtained by the GNPS nor MS2LDA strategies.

10.2 Construction of Losses Graphs from MS/MS spectra

The building of Losses Graphs from MS/MS spectra was developed by designing the following steps:

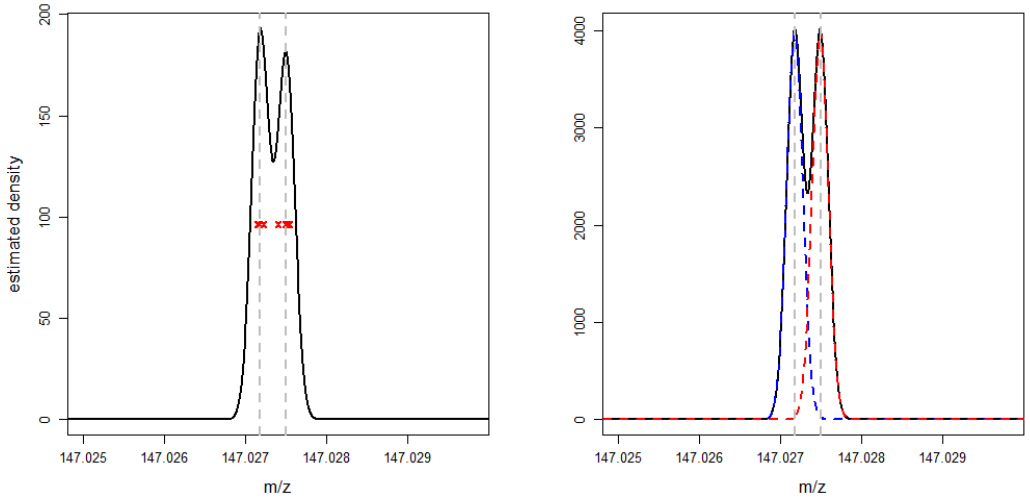
- Mass difference discretization
- Label merging
- Formulas generation
- Graph construction

The first step generates a set of discretized masses corresponding to all the losses observed in at least ϵ MS/MS spectra. ϵ is a user-furnished parameter which should be superior to 2. It computes the m/z differences between peaks within all spectra, before discretizing these differences. In the second step, labels are merged if they are too close according to the mass spectrometer accuracy. In the third step, a set of neutral formula is generated and matched to these discretized mass differences (within an m/z tolerance). In the final step, the Losses Graphs are built.

10.2.1 Mass differences discretization

In this step, all the 1 on 1 m/z differences between k most intense peaks of the MS/MS spectra are computed (k is set to 15 by default). If the precursor is not observed, a peak with an m/z corresponding to the m/z of the isolated ion is added to the spectrum. The set of the mass differences from all spectra are then discretized using a density estimation on overlapping windows, similarly to the method described in section 6.6: the algorithm has also three parameters, dmz, ppm and ϵ, where ppm and dmz account for the mass accuracy of the mass spectrometer, and ϵ corresponds to the minimum number of spectra in which a mass difference needs to be detected to be kept in the

final set (default set to 2). Within each overlapping window delimited by $mzmin$ and $mzmax$ (default width set to 0.2), the bandwidth of the kernel density is computed as $bw = max(ppm \times mzmin \times 10^{-6}/3, dmz/3)$. Each peak detected on the density is then output as a Gaussian of mean μ (apex of the peak) and of standard deviation bw (density bandwidth within the current windows; Figure 10.4b).



(a) Density estimation (b) Gaussian modeling to measure overlap

Figure 10.4: **Example of mass discretization.** a): two peaks are detected by the raw density estimation; b): Gaussian modeling used by MineMS2 to determine if the peaks (i.e., the labels) should be merged.

In some cases, the distance between two peaks detected by the density is at the limit of the resolution from the mass spectrometer, resulting in a risk of mislabeling (Figure 10.4). An additional step of label merging was therefore added, where labels are fused when they have enough overlap (Algorithm 3).

The algorithm works sequentially on all the discretized mass differences in increasing order of mass. Each label p from a set of label in increasing m/z order P is modeled by two parameters $\mu(p)$ and $\sigma(p)$which are the mean and the standard deviation of the underlying Gaussian. The algorithm relies on two main parameters $maxOverlap$ (maximum overlap between two consecutive Gaussians) and f (number of standard deviations from both Gaussians used to compute the overlap): a Gaussian p is truncated to the interval $[\mu(p) - f \times \sigma(p), \mu(p) + f \times \sigma(p)]$ to simplify the computation. The default value of f is 3, ensuring that 99% of the area of each Gaussian is considered.

At each step the overlap between the the current Gaussian p and the next Gaussian $P[i+1]$ is computed using the OVERLAP function (Line 5). If this value is superior to the $maxOverlap$ value (0.05 by default) the two labels are merged using MERGE procedure, which updates p. If the overlap is inferior to $maxOverlap$, p is set to

Algorithm 3 Label merging algorithm

1: **procedure** MERGEGAUSSIAN($P, maxOverlap, f$)
2: $n \leftarrow$ size of P
3: $p \leftarrow (-1, 0.1)$ ▷ initializing to the Gaussian with $\mu = -1$ and $\sigma = 0.1$
4: **for** $i \in 1 \ldots n$ **do**
5: **if** OVERLAP($p, P[i+1], f$) $< maxOverlap$ **then**
6: $p \leftarrow$ MERGE($p, P[i+1], f$)
7: **else**
8: **output** p
9: $p \leftarrow P[i+1]$
10: **end procedure**
11: **function** OVERLAP(p_1, p_2, f)
12: $b_{min} \leftarrow \mu(p_2) - f \times \sigma(p_2)$
13: $b_{max} \leftarrow \mu(p_1) + f \times \sigma(p_1)$
14: **if** $b_{min} > b_{max}$ **then return** 0
15: $a_1 \leftarrow$ area of p_1 on $[b_{min}, b_{max}]$
16: $a_2 \leftarrow$ area of p_2 on $[b_{min}, b_{max}]$
return $max(a_1, a_2)$
17: **end function**
18: **function** MERGE(p_1, p_2, f)
19: $b_{min} \leftarrow \mu(p_2) - f \times \sigma(p_2)$
20: $b_{max} \leftarrow \mu(p_1) + f \times \sigma(p_1)$
21: $\mu = \dfrac{b_{min} + b_{max}}{2}$
22: $\sigma = \dfrac{|\mu - b_{min}|}{f}$
return (μ, σ)
23: **end function**

$P[i+1]$ and the merging evaluation is iterated (Line 8).

At the end of this step, each Gaussian is then transformed back to a bin using the f parameter: $bin(p) = [\mu(p) - f \times \sigma(p), \mu(p) + f \times \sigma(p)]$, and eventually adjusted to ensure that the bins are disjoints.

10.2.2 Formula generation

The goal of this step is to assign an elemental (neutral) formula to each of the discretized m/z difference computed previously, when possible. All possible formulas are generated for $m/z \in [mz_{min},\ mz_{max}]$, with $mz_{min} = 14.5$ and $mz_{max} = 200$. Bins inferior to 14.5 are considered too low for a meaningful neutral loss, while neutral losses superior to 200 usually match multiple formulas.

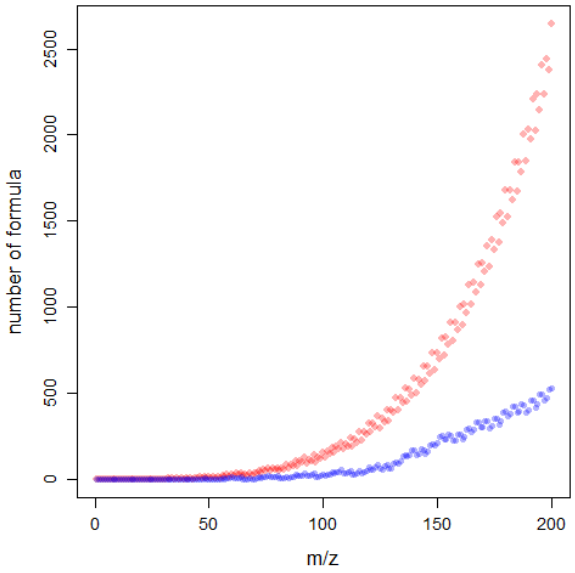


Figure 10.5: **Number of possible neutral formula as a function of m/z.** Blue: formulas corresponding to chemically existent species. Red: combinations to all such formulas by **MineMS2** to generate the set of possible losses.

The algorithm described in Böcker and Lipták 2005 is used to generate all possible formulas with a mass between mz_{min} and mz_{max} was implemented in C++. A set of rules extracted from the well known "Seven Golden Rules" (Kind and Fiehn 2007) was then used to check whether the predicted formula indeed corresponds to a chemically existent (neutral) species:

- The sum of valences or the total number of atoms having odd valences is even;

- The sum of valences is greater than or equal to twice the maximum valence;
- The sum of valences is greater than or equal to twice the number of atoms minus 1.

This filter ensures that a fully connected molecular graph (chemical structure) exists. Nonetheless an ionized molecule mays lose multiple neutral molecules consecutively. An example is a consecutive loss of two molecules of water (H_2O), resulting in a total loss of H_4O_2 which does not admit a molecular graph. Therefore all the sums from previously computed formulas were added to the set of chemically compatible losses, whose size increased considerably (Figure 10.5).

At this stage, all bins within $[mz_{min}, mz_{max}]$ and without any possible elemental formula were discarded, therefore reducing the number of possible labels for the edges. All bins with labels superior to mz_{max} were kept.

In conclusion, a table of bins was generated, where each bin b has a lower and upper bounds, b_{min} and b_{max} and a set of formulas b_f (if $b_{max} < mz_{max}$). As the bins do not overlap, they are naturally ordered: for two bins c and d we say that $c \leq d$ if $c_{min} < d_{min}$. The bins are then ordered by mass, and an integer is associated to each bin, with 1 corresponding to the bin of lowest mass, and n_L for the bin with the highest mass (n_L is the number of bins). This set of labels for our collection of spectra D is denoted $\mathcal{L}(D)$.

Losses Graph construction

Losses Graphs are built by adding a vertex for each peak of the mass spectrum, and an edge for each mass difference in $\mathcal{L}(D)$ (Figure 10.2), plus eventually an added peak for the precursor. For each of the constructed Losses Graph G the peak corresponding to the precursor is designed as $root(G)$. During the process, the frequency of each label is stored. To further strengthen the graph building procedure, especially in the case of a poor parametrization, the absence of duplication of edges is checked. If for a vertex v in a graph G there exist multiple incoming or outgoing edges with similar label a, only the edge with the mass difference closest to the center of the a bin is kept. In practice this ensures that for a fixed vertex u there is at most one incoming edge with label a and at most one outgoing edge with label a:

Property 1. *Let G be in the set of Losses Graphs D with labels $\mathcal{L}(D)$. Then:*

$$\forall u \in V(G), \forall a \in \mathcal{L}(D) \text{ there is at most } 1 \text{ vertex } v \text{ s.t. } l(u, v) = a$$

and:

$$\forall u \in V(G), \forall a \in \mathcal{L}(D) \text{ there is at most } 1 \text{ vertex } v \text{ s.t. } l(v,u) = a$$

Moreover Losses Graphs are simple graphs, since the bins are disjoint. Losses Graphs also have other interesting properties which are explained in the next section and will be used in the next chapter.

Intuitively, similarities in the decomposition patterns as shown in Figure 10.3 can be spotted on the constructed Losses Graphs by finding similar subgraphs. This problem is known as Frequent Subgraph Mining **(FSM)** problem. The goal of the next section is to give some properties which will be used in chapter 11 to construct a FSM algorithm. To do so we will use hypotheses which ensue from our Losses Graphs construction procedure.

10.3 Losses Graphs properties

As stated in the previous section, a set of **Losses Graphs** D have been constructed from a set of mass spectra S, using a set of *frequent* mass difference bins (i.e. present in at least ϵ spectra). Thanks to the procedure used to reduce the labeling errors in Section 10.2.1, we make the following assumption about the graphs:

Property 2 (Perfect Binning 1)**.** *Consider a connected subgraph $G = (V, E)$ of any graph of D. Consider $G_e = (V, E/e)$, the subgraph obtained by removing any edge e from G such that G_e remains connected; then $Supp_D(G) = Supp_D(G_e)$.*

Let's note u and v the endpoints of e, i.e $e = (u, v)$ and the label of e is l. Property 2 states that for each occurrence of G_e with isomorphism f, there is an edge between $f(u)$ and $f(v)$ of label l. This property becomes more intuitive if we remember that labels correspond to masses differences, a missing edge would indicate a mass difference of similar value but without the label, which is an error of labeling. This property leads to an interesting property of the frequent rooted and induced subgraphs of D, which are the main type of patterns that we will consider in chapter 11.

Theorem 10.1. *Two rooted induced subgraphs from D, A and B are isomorphic i.f.f there exists one spanning arborescence from A, T_A and one from B, T_B such that T_A is isomorphic to T_B.*

Proof. The proof that $A \simeq_e B$ implies an isomorphism between any of the spanning tree is trivial, by considering the same isomorphism, so $(A \simeq_e B) \Rightarrow (T_A \simeq_e T_B)$

Let us call f the isomorphism from $V(T_A)$ to $V(T_B)$, and r_A and r_B their respective roots. We show hereafter that there is an equality between the edges and vertices sets of A and their image by f in B.

Vertex set equality

By definition of the spanning arborescence, it is clear that $V(B) = \{f(v), v \in V(A)\}$, as all the vertices are contained in the spanning tree.

Edge set equality

The edge set equality may be proved by contradiction. Let us consider an edge $e = (u, v)$ such that $(u, v) \in E(A)$ but $(f(u), f(v)) \notin E(B)$. Let us denote $T_A + e$ the graph obtained by adding e to T_A. Because $T_A \simeq_e T_B$ and $T_B \subseteq_e B$, we have by transitivity of the subgraph relationship: $T_A \subseteq_e B$. Moreover $T_{A+e} \not\subseteq_e B$. As T_A is a subgraph of $T_A + e$ we have that$Supp_D(T_A) \geq Supp_D(T_A + e)$. But as T_A is a subgraph of both A and B and $T_A + e$ is a subgraph of A but not B , we have $Supp_D(T_A) > Supp_D(T_A + e)$. However, based on property 2, we have $Supp_D(T_A) = Supp_D(T_A + e)$ since $T_A + e$ is a equal to T_A plus an edge e. Therefore we have a contradiction and e cannot exist: $E(A) \subseteq E(B)$. Conversely, we show that $E(B) \subseteq E(A)$, and thus $E(B) = E(B)$. We have therefore demonstrated that $(T_A \simeq_e T_B) \Rightarrow (A \simeq_e B)$. In conclusion:

$$(T_A \simeq_e T_B) \Leftrightarrow (A \simeq_e B)$$

. □

This property implies that the support of any connected frequent subgraph of D is equal to the support of any of his spanning tree. The considered property of perfect labeling is also reflected by a second property :

Property 3 (Perfect Binning 2)**.** *If $P = u, x, v$ is a directed 2-edge path frequent in D and $\mathcal{L}(D)$ is the edge labels of D :*

$$\exists! a \in \mathcal{L}(D) s.t \forall G \in D \text{ s.t. } P \subseteq_e G \text{ then } (u, v) \in E(G) \text{ and } l(u, v) = a$$

Here the frequency refers to the ϵ parameter used to discretize edge labels. This property is illustrated on Graph 10.6 with $u = H_2O$, $x = CH_2O_2$, and $w = CH_4O_3$. If

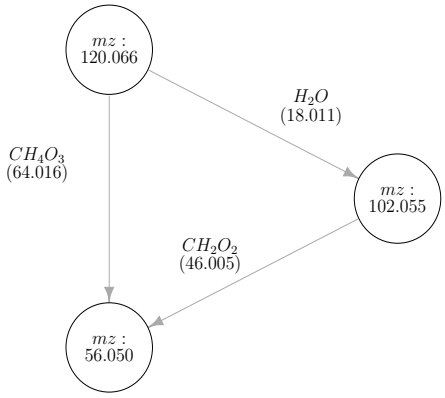


Figure 10.6: **Subgraph of 10.2**

the path is frequent, then there is an edge between both ends of the path, with a label corresponding to the sum of the mass differences on the path. The label is identical whatever the subgraph containing such a path P. This property may be extended to all the paths in any subgraph of D by induction:

Theorem 10.2. *If $P = u, \ldots, v$ is a directed path frequent in D and $\mathcal{L}(D)$ is the edge labels of D then:*

$$\exists! a \in \mathcal{L}(D), \forall G \in D \text{ s.t. } P \subseteq_e G, (u,v) \in E(G) \text{ and } l(u,v) = a$$

Proof. If $|P| = 2$ the property is evident, as there is a single edge in the path. If $|p| = 3$ the property corresponds to Property 3. Using the $k = 2$ and $k = 3$ cases as initialization, we can prove the property for every $k \geq 4$ by induction on $|P|$. Suppose that the theorem is true for all path P s.t $|P| = k \geq 2$, let us prove it for path any path P' s.t $|P'| = k + 1$.

If Q is a path of size $k+1$ from r to v, it may be written as $P = r, x_1, \ldots, x_{k-1}, v$. Now let us consider $P' = r, x_1 \ldots x_{k-1}$, which is of length k. As P' is of length k and rooted in r, there exists an edge from r to x_{k-1} because of the induction property. Therefore r, x_{k-1}, v is a frequent path of size 2 from G and, on the basis of Property 3, there exists an edge from r to v. Therefore the property is true for $k+1$, and therefore by recurrence for all $k \geq 2$. □

Using this property it is possible to derive an interesting property for any frequent arborescence in D :

Theorem 10.3. *For a given frequent arborescence P with root r in D, let us denote by G_P the subset of D in which P occurs, i.e. $G_P = [G \in D | P \subseteq_e G]$. For an element*

G of G_P let's write f_G the corresponding isomorphism, then :

$$\forall G \in G_P, \forall v \in V(P), \exists! a \in \mathcal{L}(D) \ s.t \ (f_G(r), f_G(v)) \in E(G) \ and \ l(f_G(r), f_G(v)) = a$$

Proof. To prove this theorem, consider a vertex v of the spanning arborescence P and consider a path q from r to v. As the arborescence is frequent in D, q is frequent in D. Therefore by theorem 10.2 there is an edge from $f_G(r)$ to $f_G(v)$ with a unique label a. □

Using both Theorems 10.1 and 10.3, it is possible to derive a simple criterion to test if a frequent arborescence is a subgraph of a graph of D:

Theorem 10.4. *Let us denote the set of labels of the edges originating from u within a graph H whose labeling function is l by $L_u(H)$ i.e.:*

$$L_H(u) = [a | \exists v \in V \ s.t. \ (u, v) \in E(H) \ and \ l(u, v) = a]$$

Let P be a frequent arborescence from D with root r, then for $G \in D$:

$$\underbrace{P \subseteq_e G}_{A} \Leftrightarrow \underbrace{\exists v \in V(G) \ s.t. \ L_P(r) \subseteq L_G(v)}_{B}$$

Proof. The proof of $A \Rightarrow B$ is trivial given the definition of our isomorphism: if G contains an induced subgraph isomorphic to P, v exists and is the image of r by the isomorphism. We will therefore focus on the proof of $B \Rightarrow A$ and assume that $\exists v \in G'$ s.t. $L_G(r) \subseteq L_{G'}(v)$.

Let us consider $T_G(r)$, the subgraph of G consisting of all the vertices from G and all the edges originating from the root of G. Because it covers all the vertices, it is rooted and there is a single path from the root to all the vertices (1-edge long), it is a spanning arborescence.

Similarly, let us consider the subgraph of G' rooted in v and consisting of all edges originating from v with labels in $L_G(r)$, noted $T_{G'}(v)$. It includes as many edges as $T_G(r)$, because $L_G(r) \subseteq L_{G'}(v)$, and therefore as many vertices. Now let us consider the function $f : \ V(T_G(r)) \to V(T_{G'}(v))$ defined by:

$$f(u) = \begin{cases} v & \text{if } u = r \\ u' \text{ s.t. } l(v, u') = l(r, u) & \text{if } u \neq r. \end{cases}$$

Then $f(u)$ exists for all $u \in V(T_G(r))$ because $V(T_G(r)) = V(G)$, and $L_G(r) \subseteq L_{G'}(v)$. Since, by construction, there is a single outgoing edge from v to each label, this function is bijective. In addition, by definition of f, $\forall u \in V(G)$, $l(r,u) = l(f(r), f(u)) = l(v, f(u))$. Therefore f defines an isomorphism and $T_G(r) \simeq_e T_{G'}(v)$. And since $T_G(r)$ and $T_{G'}(v)$ are spanning arborescence of G and G' respectively, using Theorem 10.1, G and G' are isomorphic. □

Using this theorem, the following algorithm was developed to find a mapping between a frequent arborescence P and a Losses Graph G.

Algorithm 4 Subgraph isomorphism algorithm

```
1:  function SUBGRAPHISOMORPHISM(P, G)
2:      n_P ← |P|
3:      n_G ← |G|
4:      r ← root of P
5:      P_0 ← L_P(r)
6:      V_G ← V(G) in reverse topological order
7:      while n_G ≥ n_P do
8:          v ← V_G[n_G]
9:          S_v ← L_G(v)
10:         if P_0 ⊂ S_v then
11:             M ← empty map
12:             for i successor of r in P do
13:                 M[i] ← successor of v in G with edge label l(r, i)
                return M
14:         else
15:             pass to the next iteration of the While loop
        return empty map
16: end function
```

This algorithm sequentially tests the vertices of G in topological order on line 8. At each step it tests if the set of labels of the outgoing edge of the considered node in G is a superset of the set of labels form the outgoing edges from the root in P (Line 10). If the condition is true by theorem 10.4 an isomorphism if found and the mapping is constructed (Line 12). If not (Line 15), the algorithm passes to the next vertex of G. If at the end not enough vertices remains to be considered the algorithm return an empty map indicating that no isomorphism was found. Therefore the worst time complexity of this algorithm (when no isomorphism is detected) is $O((|P| + |G|)(|G| - |P|))$: the left term corresponds to the computation of the inclusion (Line 10), and the right term is the number of visited vertices (Line 7). However if the root of the pattern in G is known, there is a derived algorithm (by considering only this root over the full V_G set) with complexity $O((|P| + |G|)$, which is used extensively in the next chapter.

Similarly, the complexity of the algorithm which checks for graph isomorphism is $O(|P| + |G|)$, because only the root of the graph H needs to be tested. Such fast algorithms to check isomorphism are crucial for Frequent Subgraph Mining (FSM) tasks.

Intuitively, similarities in the decomposition patterns as shown in Figure 10.3 can be seen as isomorphic subgraphs. Thus, a way to find these similarities is to extract the recurring subgraphs from D. Such a task is known as **Frequent Subgraph Mining (FSM)**, and a specific FSM algorithm for **Losses Graph** will be detailed in the next chapter.

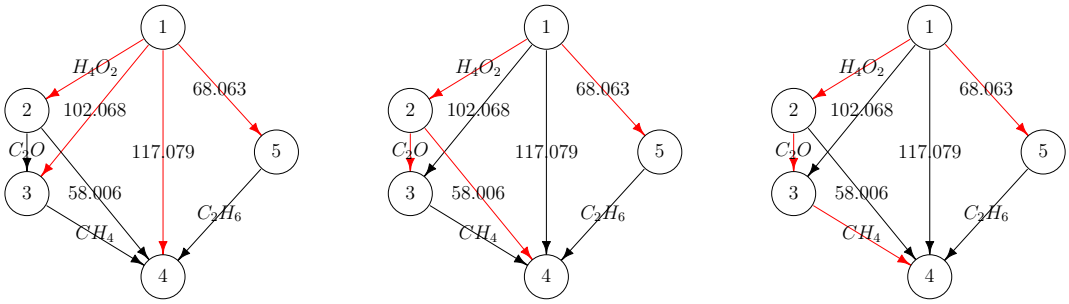
Chapter 11

MineMS2: A Frequent Subgraph Mining Algorithm for Fragmentation Spectra

In this chapter we describe the Frequent Subgraph Mining (FSM) algorithms developed to mine the Losses Graphs generated in Chapter 10, which is implemented in the MineMS2 software. In Section 11.1, we present the structure of the mined patterns, and we show how specific constraints on the patterns topologies may be used to reduce the search space. First, we define an adapted canonical form allowing us to build an efficient data structure in Section 11.2. Second, the full suite of algorithms for closed subgraph generation is described in Section 11.2.2. Third, preliminary results on experimental data are presented in Section 11.5.

11.1 Reduction of the pattern search space for Losses Graph mining

Because the graph search space increases exponentially with the number of edges, reduction of this search space is critical. In the case of the **Losses Graphs** , the challenge is the high number of edge labels (179 and 1368 for the **PenicilliumDIA** and **LemmDB** datasets, respectively). This is to be compared with the usual number of labels in classical graph FSM algorithm, which is generally lower than 100. In fact, we observed that neither gSpan (Yan and Han 2002) nor GASTON (Nijssen and Kok 2004) were able to process our datasets in less than 2 hours. We therefore decided to develop new algorithms taking into account the specificities of our graphs.



(a) **1-LMDF** (b) **2-LMDF** (c) **3-LMDF**

Figure 11.1: **Examples of k-LMDF spanning trees for the same pattern but distinct values of k (PenicilliumDIA dataset).** The k-LMDF spanning tree is shown in red.

11.1.1 Patterns specificity for Losses Graph mining

To facilitate the mining process, we chose to consider only rooted patterns. This means that the peak corresponding to a substructure is assumed to be present in the spectrum. Although this hypothesis may seem quite strong a priori for single MS-MS acquisitions, fused spectra acquired at multiple energies are commonly used in other MS-MS softwares such as CSI-FingerID(Shen et al. 2014), therefore ensuring that more fragments are present. As we want to mine the losses occurring from a single substructure, there should be a path between the root of the pattern and any node. Therefore the mined patterns are Acyclic Flow Graphs (abbreviated **AFG**), as they admit a path from the root to each node. Moreover we chose to consider only induced subgraphs, as the addition of an edge does not add any information (in term of support), as shown in Theorem 10.1.

We will therefore refer to these patterns using the acronym **AFG** in the rest of this chapter, and make use of the Theorems 10.1 and 10.2, which apply to AFGs.

11.1.2 A canonical form of AFG : the k-LMDF Spanning Tree

Since AFGs are rooted and induced subgraphs from D, they satisfy Theorem 10.1: two subgraphs from D are isomorphic if they have a common spanning tree. In practice, this means that if we define a function f from G to $\mathcal{T}_G$, the set of spanning trees from G, therefore distinguishing a spanning tree of G among the set of all its spanning trees, then f is a canonization function.

To reduce the search space, we defined the k Left-Most Depth First spanning tree

as canonical form (noted **k-LMDF**). It is the spanning tree defined by performing a DFS on the a graph G, such that i) the edge with the minimum label is selected first and ii) the search stops at depth k. Alternatively, the k-LMDF canonical form can be defined as the spanning tree with minimum **DFS Label Sequence**, where the DFS label sequence is defined in M. J. Zaki 2005, and illustrated in Figure 9.4. Three examples of such spanning trees for distinct values of k are shown in Figure 11.1.

The k-LMDF spanning tree is always defined, because there is at least one spanning tree with a maximum depth of 1 by considering all the edges originating from the root (Property 10.3). It is evident that the graphs admitting a spanning tree of maximum depth k admit a spanning tree of maximum depth $k+1$. Therefore, as all the considered graphs admit a spanning tree of depth 1, they admit a spanning tree of depth k for any $k > 1$. So the k-LMDF shape is defined for all k for the considered patterns. The specific set of edges originating from the root is of particular interest because it defines a labeling function for each node from the AFG pattern, thus simplifying the generation of the k-LMDF spanning trees.

Although 1-LMDF trees are simpler to generate (as they are basically sets of labels), the 2-LMDF form limits the complexity of graph generation and simplifies the computation of the support. The 2-LMDF form is therefore used by the MineMS2 FSM algorithm to generate the trees. Furthermore, this canonical form allows us to define a new data structure, the k-Path Tree, which, in turn, drastically reduces the search space.

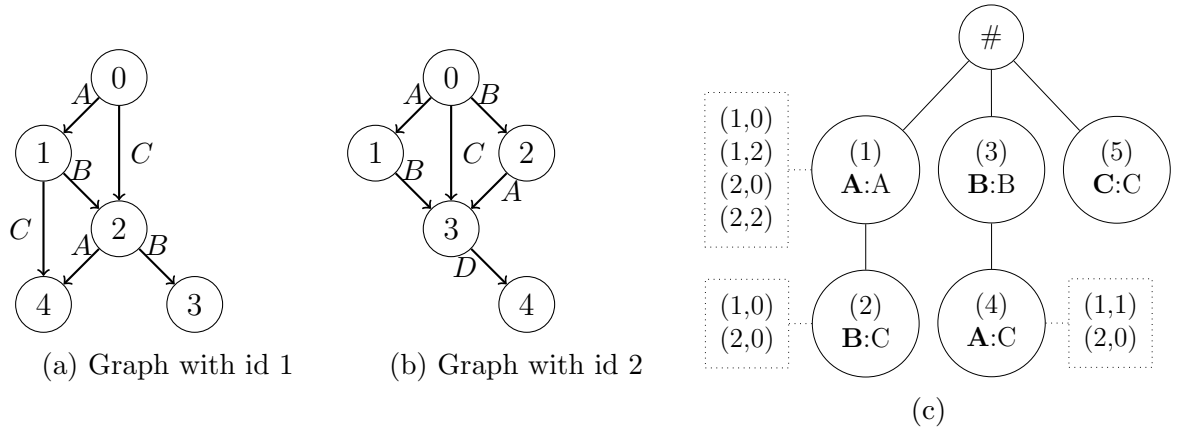
11.1.3 A dedicated data structure for FSM on Losses Graphs : the k-Path Tree

The proposed k-Path Tree is very similar to the Frequent Pattern tree ($FP-tree$) described in Han et al. 2000, except that it stores paths over itemsets. The intuition behind this k-Path Tree is that the set of frequent paths is sparse compared to the set of all the possible paths of size k which can be generated using the set of labels.

To facilitate the mining of AFGs, the k-Path Tree stores three additional informations about each vertex v (Figure 11.2):

$l_T(v)$ the label of v.

$h_T(v)$ the label of the edge linking the first to the last vertex of the path (which exists and is unique as demonstrated in Theorem 10.2



(a) Graph with id 1 (b) Graph with id 2 (c)

Figure 11.2: **Example of the 2-path tree generated from a database D consisting of two graphs.** The 2-path tree 11.2c contains all the paths of size 2 within graphs 11.2a and 11.2b. For each node of this tree are attached: i) within the node circle: an ID in parenthesis, $l(v)$ in bold, and $h(v)$ and ii) in the dotted box: $O(v)$, which is used to prune the paths with a frequency below the ϵ threshold.

$O_T(v)$ the list of occurrences.

The building of the k-Path Tree T is described in Algorithm 5. To build the k-Path Tree the Losses Graphs from the database D are processed sequentially (Line 2), and an integer id is associated to each Losses Graph . For each graph, the path enumeration is performed using a modified Depth-First Search. The first difference with standard DFS is that an indicator of visit named vis is stored for each edge, with value in $\{unknown, closed, visited\}$. These labels are used to speed up computation and to avoid to visit multiple times the same path:

unknown indicates that the edge has never been visited, therefore all the paths including this edge may be added.

closed means that the edge has already been visited from the current vertex, and should not be explored further.

visited indicates that the edge has been visited in a previous branch of the DFS tree. Therefore no path starting from this edge or from a successor of this edge needs to be explored.

At each step forward, all the paths of size 1 to k are added (Line 28) to a vertex v by adding a pair (g_{id}, r) where g_{id} is an id corresponding to the graph and r is the root of the path. Each time the algorithm backtracks from a vertex v, all the outgoing

Algorithm 5 k**-Path tree construction**

```
1: function constructKPathTree(D,k)
2:     for i ∈ 1,...,|D| do
3:         AddToKPathTree(T,D[i],k,i)
4: end function
5: procedure AddToKPathTree(T,G,k,id)
6:     N ← empty stack
7:     R ← nodes from G without parents                          ▷ roots in G
8:     for r ∈ R do
9:         n ← r, s ← NULL
10:        append n to N
11:        while n ≠ s do
12:            s ← NextNode(n, G, N)
13:            if s = n then
14:                n ← pop an element of N
15:                label all outgoing edges from n as visited
16:            else
17:                n push s into N
18:                AddPaths(T, N, G, k, id)
19: end procedure
20: function NextNode(n, G, N)
21:     E_n ← non closed outgoing edges from n in G
22:     if |E_n| ≠ 0 then
23:         e ← edge of E_n with minimum label
24:         vis(e) ← closed
            return Target vertex of e
        return n
25: end function
26: procedure AddPaths(T,N,G,k,id)
27:     i ← 0
28:     while i < k and |N| − k > 0 do
29:         if vis(N[|N| − i], N[|N| − i + 1]) = visited then return
30:         o = (id, N[|N| − i])
31:         if (N[|N| − i], N[|N|]) ∈ E(G) then
32:             N_t ← position of path N[|N| − i], ..., N[|N|] of G to T
33:             Add occurrence o to O(N_t)
34:         elsereturn
35: end procedure
```

edges are labeled as *visited*. This prevents the algorithm from adding the same path multiple times.

A pruning of the non-frequent paths is performed at Line 31, which follows directly from Theorem 10.2: if there is no edge between two nodes but there is a path between them, this path may not be frequent. Moreover Theorem 10.2 shows that the edge labeling is unique among all the frequent paths.

The k-Path Tree has multiple interesting properties. First, it contains all the k-LMDF spanning trees of all the frequent AFGs. This is because the k-Path Tree contains (in the sense of the subgraph relationship) all the paths up to size k present in D, and therefore all the trees of maximum depth k. Therefore it also contains all the trees in k-LMDF shapes. Moreover using Algorithm 5, it can be built such that a large part of the non frequent paths are not considered, therefore reducing the search space considerably.

The problem of generating all frequent spanning trees over the dataset then becomes the problem of enumerating all the subtrees of the constructed k-Path Tree which include the root while ensuring that they correspond to k-LMDF spanning trees.

11.2 Mining Closed AFGs using the k-path tree

In this section we consider that a k-Path Tree T has been constructed from a set of Losses Graphs D with a set of labels $\mathcal{L}(D)$. We will start by giving an overview of the pattern structure in 11.2.1. We will then describe the full FSM algorithm used by MineMS2 in 11.2.2. Each building block of this algorithm will then be explained in the following subsections. 11.2.3 gives a condition to ensure that a subtree of T may exist in D. We will then show how the support of these subtrees may be computed easily using the k-Path Tree(11.2.4). Section 11.2.5 details the process of 2-LMDF spanning tree enumeration.

11.2.1 Pattern structure in MineMS2

In the MineMS2-FSM algorithm, a pattern consists of 5 elements:

G**:** The graph representing the pattern.

Exts: The list of its possible extensions where each extension is listed as a triplet (n_G, l, n_T), with n_G the starting vertex of the extension in G, l the label of the edge, and n_T the corresponding vertex in the k-Path Tree. This list is always sorted in Depth First Left-Most order.

S: A set of the forbidden values of $h(n)$.

O: A set of occurrences of the pattern in D.

n: The number of vertices in the graph.

The graph G represents the pattern. Two supplementary informations for each vertex v of G are computed: the depth, denoted $d(v)$, and the label of the edge between the root of G and the node v, which is denoted $h(v)$ as a label of node v, and corresponds to the h_T label extracted from the corresponding node in T.

11.2.2 Overview of the MineMS2-FSM algorithm

The MineMS2 FSM algorithm performs a Depth First exploration of the pattern space (recursive call on Line 21). It starts from a set of seed patterns including 1 or 2 edges (Line 13). At each step, a single seed pattern P is extended by a single edge, and the k-Path Tree is used to check that the extended candidate: i) is frequent (Line 15), and ii) is a possible subtree of a graph from D (Line 13). The closure of the frequent tree is then checked (Line 20). If this is the case, the full AFG graph is rebuilt (Line 20), and returned.

11.2.3 Ill-formed subtrees of T

Here we describe a simple criterion to detect a subtree of T which may not be a subtree of any graph of D. An example of such a subtree is shown in Figure 11.3. This subtree cannot occur in the original graphs of Figure 11.2 because it features two paths with the same endpoints in the original graphs: it is not a subgraph of any graph of D. We will call such subtrees **ill-formed**. An example of ill-formed subtree is shown in Figure 11.3. Ill-formed subtrees can be detected by using the h_T label of the k-Path Tree.

Theorem 11.1. *Let D be a set of Losses Graphs and T be their corresponding k-Path Tree for a fixed k and a fixed frequency ϵ. Let $G \in D$ and T_G be a frequent subtree of D*

Algorithm 6 Overview of the MineMS2 FSM algorithm

function MINECLOSEDAFGS(D, ϵ, k, tol)
    $T \leftarrow$ CONSTRUCTKPATHTREE(D, k)
    ENUMERATECLOSEDAFGS(T, D, ϵ)
end function
function ENUMERATECLOSEDAFGS(T, D, ϵ)
    $S \leftarrow$ 1-edge or 2-paths patterns
    for $P \in S$ **do**
        $e \leftarrow$ first elements of $Exts(P)$
        EXTENDKLMDFTREE(P, e, T, D, ϵ)
end function
function EXTENDKLMDFTREE(P, e, T, D, ϵ)
    $P_e \leftarrow$ addition of edge e to P
    if P_e is ill-formed **then return**
    $Occs(P_e) =$ CALCOCCURENCES(P, e, T)
    if $|Occs(P_e)| < \epsilon$ **then return**
    $E_{new} \leftarrow$ NEWEXTENSIONS(T, e)
    $E_{inf} \leftarrow [e' \in Exts(P) | e' \leq e]$
    $Exts(P_e) \leftarrow Exts(P) \cup E_{new}/E_{inf}$
    Remove impossible extensions from E
    if ISCLOSED(P_e, D) **then**
        $P_e \leftarrow$ RECONSTRUCTAFG(P_e, D)
        output P_e
    for $f \in Exts(P_e)$ **do**
        EXTENDKLMDFTREE(P_e, f, T, D, ϵ)
end function

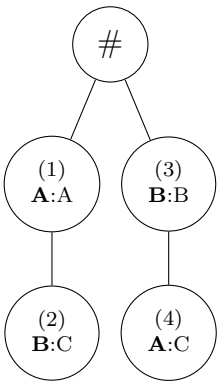


Figure 11.3: **Example of an ill-formed subtree**. Extracted from the k-Path Treeshown in Figure 11.2c.

occurring in G, rooted in r, and with a maximum depth inferior to k. By construction of T, $T_G \subseteq T$. Let ϕ be the mapping from T_G to T. Then:

$$\forall v, v' \in V(T_G), v \neq v' \Leftrightarrow h_T(\phi(v)) \neq h_T(\phi(v'))$$

.

Proof. Let us suppose that there are $v, v' \in V(T_G)$ satisfying $h_T(\phi(v)) = h_T(\phi(v'))$, and let us write $p = r, \ldots, v$ and $p' = r, \ldots, v'$ the paths between the root and the final node. Because of Property 10.2, there exists an edge between r and v, with a label $h_T(\phi(v))$. Similarly, there exists also an edge between r and v', with label $h_T(\phi(v')) = h_T(\phi(v))$. Since the outgoing labels from an edge are unique (Property 1), there exists a unique edge with a given origin and label in a graph. As a result, $v = v'$. □

In practice, we will call a subtree T_a of T **ill-formed** if there exists two nodes u and v of T_a such that $h_T(u) = h_T(v)$. In contrast, if a tree is not ill-formed, it will be called a **correct** subtree. It is clear that if a subtree is ill-formed, any supertree of this subtree is ill-formed. As the proposed algorithm performs a Depth-First exploration, this property is important to reduce the search space, as the algorithm may backtrack from its DFS branch every time it meets an ill-formed pattern.

11.2.4 Efficient Support Computation

For the correct subtrees, the storage of the root of each path in the k-Path Tree enables a simple computation of the support:

Theorem 11.2. *Given a k-Path Tree T generated from a set of Losses Graphs D, and Y be a correct subtree of T, then:*

$$|Supp_D(Y)| = \bigcap_{y \in Y} O_T(y)$$

Proof. Let us call $\phi_Y(D)$ the set of all isomorphisms between Y and the subgraph of graphs from D. It is clear that $|Supp_D(Y)| = |\phi_Y(D)|$. The inclusion $\phi_Y(D) \subset \bigcap_{y \in V(Y)} O_T(y)$ is evident, a tree rooted in r occurs imply that all the path constituting the trees occurs from r. We therefore just have to prove that $\bigcap_{y \in V(Y)} O_T(y) \subset \phi_Y(D)$.

Suppose that there is a pair (g, v) of $\bigcap_{y \in V(Y)} O_T(y)$. Write G the graph corresponding to g. For all $y \in V(Y)$ let us write $p_y = r, \ldots, y$ the path from the root of T, r, to y, which exists by definition of T.

By construction of the occurrence set O_T, $\forall y \in V(Y)$, there exists an isomorphism $f_y : V(p_y) \to V(G)$ with $f_y(v) = r$ such that $p'_y = v, \ldots, f_y(y)$ and $p \simeq_e p'$. Now let us consider the set V_Y in $V(G)$ such that $V_Y = [f_y(y), \forall y \in V(Y)]$. Let us consider the function f defined as $f : V(Y) \to V(G)$ such that $f = f_y(v)$, as f_y exists for all $y \in V(Y)$.

Now let us prove that f is injective by contradiction. Consider $y_1, y_2 \in V(Y)$ such that $y_1 \neq y_2$. By construction of T the label of edge $(v, f(y_1))$ in G is $h_T(y_1)$, similarly the label of edge $(v, f(y_2))$ in G is $h_T(y_2)$. As Y is a correct subtree we have $h_T(y_1) \neq h_T(y_2)$ so $(v, f(y_1))$ and $(v, f(y_2))$ are two distinct edges. As G is not a multigraph this means that $f(y_1) \neq f(y_2)$. Therefore f is injective.

So by defining Y_G the image set of $V(Y)$ through f, f defines a bijection between $V(Y)$ and Y_G, as f is injective and $V(Y)$ and V_Y have the same cardinal.

Now consider any edge of $(u, w) \in E(Y)$ with $u, w \in V(Y)$. As Y is a tree, u is a children of w or reciprocally. We will consider that w is the children of u, but the demonstration is symmetric. As there is a single path form the root to any node in the tree p_w the path from r to w includes u, therefore $(u, w) \in E(p_w)$ and therefore $(f(u), f(w)) \in E(p'_w)$. As $p'_w \subseteq_e G$, we have that $(f(u), f(w)) \in E(G)$.

Therefore we have defined a subgraph isomorphism which preserves edge label between Y and G and is therefore included in $\phi_{T_Y}(D)$. So $\bigcap_{y \in V(Y)} O_T(y) \subset \phi_{T_Y}(D)$. Therefore by double inclusion we have that $|Supp_D(T_Y)| = \bigcap_{y \in Y} O_T(y)$ □

This theorem ensure that the support of any correct subgraph of T and therefore of any AFG can be computed simply by intersecting the occurrences sets O of the corresponding vertices from T.

11.2.5 2-LMDF frequent spanning trees enumeration

We describe here an algorithm to enumerate the frequent spanning trees of 2-LMDF shape from the 2-path trees of a set of Losses Graphs D with labels $\mathcal{L}(D)$. We use a pattern-growth approach similar to Asai et al. 2002; Mohammed J. Zaki 2002, where patterns are extended using **right-most path extensions**.

11.2.6 Frequent subtree enumeration

Algorithm 7 2-LMDF spanning tree enumeration

```
1: procedure EnumerateLMDFSubtrees(T, ε)
2:     P_0 ← InitializePatterns()
3:     for p ∈ P do
4:         EnumerateExtensions(P, T, ε)
5: end procedure
6: procedure EnumerateExtensions(P, T, ε)
7:     output P
8:     for e ∈ P.exts do                    ▷ Adding an edge and a node to P
9:         P_e ← ExtendPattern(P, e, T, ε)
10:        if P_e ≠Empty pattern then
11:            EnumerateExtensions(P_e, T, ε)
12: end procedure
13: function ExtendPattern(P, e, T, ε)
14:    d ← getLabel(T, e)
15:    if d ∈ P.S then return Empty Pattern    ▷ ill-formed or non-LMDF subtree
16:    P_e ← Copy of P
17:    P_e.O ← P.O ∩ getOccs(T, e)
18:    if |P_e.O| < ε then                     ▷ Non frequent tree return
19:    P_e.S ← P.S ∪ d
20:    E_inf ← [e' ∈ P.exts| e' ≤ e]
21:    E_sup ← [e' ∈ P.exts|e' > e]
22:    for e' ∈ E_inf do                       ▷ Non LMDF extensions
23:        d' ← getLabel(T, e')
24:        Adds d' to P_e.S
25:    Exts_e ← NewExtensions(e, T)
26:    for e' ∈ Exts_e do                  ▷ extension leading to ill-formed subtree
27:        d' ← getLabel(T, e')
28:        Remove extension (n_G, d', .) from Esup
29:    P_e.exts = Exts_e ∪ Esup
30: end function
```

The algorithm first builds the set of 1-edge patterns as seed patterns (Line 2). These patterns are then expanded in a Left-Most manner by using the stored $Exts$ lists, and in a depth first manner by the recursive call on Line 11. At each extension step, a single edge e is added and the extensions which are not on the right-most path anymore are removed (E_{inf} in Line 20). To ensure that another spanning tree including a node n with similar $h(n)$ would not be generated later, $h(n)$ is included in the set of forbidden value $P_e.S$ (Line 24). Finally, when an extension is added, the set of the new extension is extracted from the k-Path Tree(Line 25). For these extensions,

the corresponding h values are computed and removed from the extensions of P_e with the same origin as e. A complete example is available in Annex A.

11.2.7 Completeness of the enumeration algorithm

The proof of completeness of the algorithm derives from the fact that right most extension procedures are exhaustive (as demonstrated in Asai et al. 2002).

In Section 11.2.3, we have seen that if a graph was ill-formed, all of its supergraphs were also ill-formed. Consequently, the extensions of a graph G may be stopped directly as soon as G becomes ill-formed: the MineMS2 FSM algorithm therefore checks that i) the tree is correctly formed (Line 15), and that extensions which would lead to ill-formed trees are discarded (Line 26). This ensures that all generated subtrees are correct.

As a consequence, it is possible to use Theorem 11.2 to compute the support of the pattern in D (Line 17). Similarly to ill-formed patterns, if G is not frequent, its supergraphs are not frequent, and G should not be expanded (Line 18).

We have shown that the generated subtrees are correct and frequent. Line 22 further ensures that they are left-most: every time an extension e is added, all the edges at the left are put in E_{inf} and their corresponding h_T value is added to S. By condition on Line 15 a node with a similar value of h_T is never added, therefore no extension at the left of an existing extension may be added. Similarly to the ill-formed and frequent characteristic, it is clear that if a spanning tree is not LMDF, neither are its supergraphs.

Therefore the algorithm stops only when it meets a tree which is not frequent or not k-LMDF. In contrast, any k-LMDF tree may be produced as all its subtrees are k-LMDF, and therefore by completeness of the right-most extension, it will be constructed by the proposed algorithm.

Therefore the proposed algorithm generates all the 2-LMDF frequent subtrees. However, as the full set of frequent 2-LMDF subtrees is explored, two additional steps are required to ensure that closed AFG patterns are mined: i) a way to ensure that the mined patterns are closed and ii) an algorithm to rebuild the AFGs from the 2-LMDF subtrees.

11.3 Mining closed patterns only

A subgraph is closed if there is no frequent supergraph of this graph. In practice, the concept of closed patterns have been initially developed to mine closed itemsets, e.g. in the CHARM algorithm (Mohammed J. Zaki and Hsiao 2002). The method relies on the fact that, given a set of occurrences of the patterns, their biggest common patterns (called the closure) may be generated directly from the items. This approach enables an efficient pruning of the search space. However, the construction of the biggest common graph from a set of graphs is known to be NP-complete (Garey and D. S. Johnson 1979), and is not correctly defined in the case of rooted graph (there can be multiple biggest common rooted graphs with different roots). Among the techniques developed to mine closed subgraphs, the most generic is the closeGraph algorithm (Yan and Han 2003). This algorithm highlights one of the main challenge of the mining of closed subgraphs: Frequent Subgraph Mining algorithms focus on the generation of each candidate a single time. Therefore there is often no link in the generation process between a subgraph and one of its supergraph, except in the specific case of right most extensions. As a result, a supplementary step to detect the potential supergraphs is necessary.

The task requiring to get the full set of the frequent subgraphs to be sure that a pattern is frequent may seem expensive. However, a useful property to reduce this search was proved for general labeled graphs in Part 5 of Yan and Han 2003:

Theorem 11.3. *Given two graphs G and G' s.t. $G \subseteq G'$ and $Supp(G) = Supp(G')$, then $\exists H$ an extension of G by a single edge, or a node and an edge and $H \subseteq G'$, s.t $Supp(G) = Supp(H)$*

Therefore it is possible to evaluate whether a subgraph S is complete by simply checking that there exists a 1-edge and 1-vertex (v) extension from S which have the same support. In our case it is even simpler as any 1-vertex extension will either add a node v linked to the root, as all vertices have an incoming edge from the root by Theorem 10.3, or a new root (in which case the old root of S will have an incoming edge from v).

Therefore a simple procedure may be derived to ensure that a graph is closed, by considering only out-going and incoming-edges from the root described in algorithm 11.3.

```
1: function isClosed(S,D)
2:     (g_0, v_0) ← first element of O(S)
3:     E_in ← labels of incoming edge of v_0 in D[g_0]
4:     E_out ← labels of out-going edge of v_0 in D[g_0]
5:     H_0 ← [h(v)|v ∈ V(S)]
6:     E_out ← E_out − H_0
7:     for (g, v) ∈ O(S)|(g, v) ≠ (g_0, v_0) do
8:         E_in,v ← labels of incoming edge of v in D[g]
9:         E_out,v ← labels of out-going edge of v in D[g]
10:        E_in ← E_in ∩ E_in,v
11:        E_out ← E_out ∩ E_out,v
12:        if E_in and E_out are empty then return True
       return False
13: end function
```

11.3.1 Reconstructing AFG form 2-LMDF tree

By property 2, an AFG G may be reconstructed from the initial set of Losses Graphs D, by selecting any occurrence of G and mapping it to the associated graph. This can be done by a single set matching using Theorem 10.3. Finally, the edges between the matched nodes may be added to the k-LMDF tree.

11.4 Implementation

The full suite of developed algorithms was implemented as an R package named **MineMS2**, and all the graph mining algorithm is implemented in C++ to speed up computation. MineMS2 takes as input the database of spectra as a single file in the standard mgf format for MS/MS data potentially generated by the MS2process software described in section 9.9, and may return visualizations of the found patterns, as well as their occurrences on the mass spectra, and eventually a network containing the patterns and the spectra for a more general visualization.

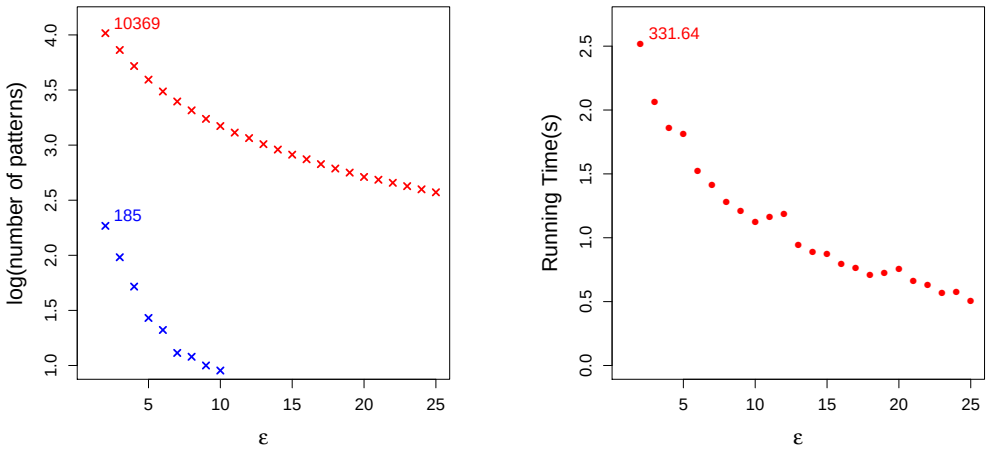
11.5 Experimental results on real datasets

MineMS2 was applied to two datasets corresponding to two case studies. The first dataset (**PenicilliumDIA**) results from Data Independent Acquisitions to character-

ize the secondary metabolome from the pathogenic fungus *Penicillium verrucosum* Hautbergue et al. 2017. A majority of the precursors are polypeptides. The second dataset (**LemmDB**) is a database of MS/MS spectra from pure compounds (Laboratoire d'Etudes du Metabolisme des Médicaments at CEA). It consists of 663 spectra representing complementary chemical families (organic acids, amino acids, hormones, plant metabolites, xenobiotics) selected according to their biochemical relevance, occurrence in biofluids, and commercial availability(Roux et al. 2012). For these two datasets, the running time and the number of mined patterns were computed as a function of the minimum frequency (Figure 11.5). The parameters used for the building of the Losses Graphs are given in Table 11.1. All computations were performed on a laptop with a 2.6 Ghz Intel i7 processor and 8 Go of memory.

dataset	**PenicilliumDIA**	**LemmDB**
Number of spectra	45	663
maxFrags	15	
ppm	8	2
dmz	0.005	0.001
heteroAtoms	No	Yes
Number of labels	179	1,368

Table 11.1: **Parameters used for the building of the Losses Graphs .**



(a) Number of mined patterns

(b) Running time in seconds

Figure 11.4: **Application of MineMS2 to two real spectral collections.** The number of detected patterns (a) and the running time (b) as a function of the ϵ (frequency) parameter are displayed for the **LemmDB**(red) and **PenicilliumDIA**(blue) datasets.

MineMS2 was shown to be highly efficient, since both datasets could be mined in

less than 6 min, although the **LemmDB** database contains more than 600 spectra. In contrast, the reference algorithms for graph mining, GASTON (Nijssen and Kok 2004) and gSpan (Yan and Han 2002), were still running after 2 h (data not shown). In addition, gSpan had already detected more than 2,000,000 patterns, showing that our targeting strategy for subgraphs is of critical importance to obtain meaningful results. Regarding the reference closed patterns mining algorithms, none of them could be tested due to limited availability or incompatibility with our edge labeled graphs.

MineMS2 is therefore, to the best of our knowledge, the only algorithm able to mine closed patterns from Losses Graphs . Since the current number of detected patterns remains high, thus preventing a manual browsing by the end-user, we propose an approach to reduce this number of patterns by using the inherent lattice formed by the closed pattern. This approach is described in the discussion section as it is still a work in progress.

Chapter 12

Discussion

In the previous part we propose a new method of pattern mining using an efficient subgraph mining algorithm, we will discuss the limits and possible extensions of this methodology and his coupling with other existing algorithms.

The first limitations of the proposed methodology is the very high number of mined motifs generated by our FSM approach. It therefore makes the implementation of step of summarization of the pattern set mandatory. In the next section, we propose a methodology to summarize the set of patterns by selecting the most informative. This is presented in discussion as at the moment of writing of this PhD, it still is a work in progress.

12.1 Summarizing the detected subgraphs

One of the main issues with Frequent Subgraph Mining Approach is the very high number of mined subgraphs (Jiang et al. 2013). In this section, we present potential strategies to summarize the set of mined closed subgraphs. We define the set L as $L = D \cap C \cap \emptyset$, with D the set of Losses Graphs , C the set of the closed AFGs, and $\emptyset$ the empty graph. L is a subset from the set of all induced subgraphs from D, and is therefore partially ordered by the subgraph relationship (denoted by $\subseteq$).

As L is a finite and partially ordered set, it can be represented by a Directed Acyclic Graph G where the vertices correspond to the elements of L, i.e. there exists a bijection $f : L \to G$, and there is an edge from $u \in V(G)$ to $v \in V(G)$ if $f^{-1}(u) \subseteq f^{-1}(v)$.

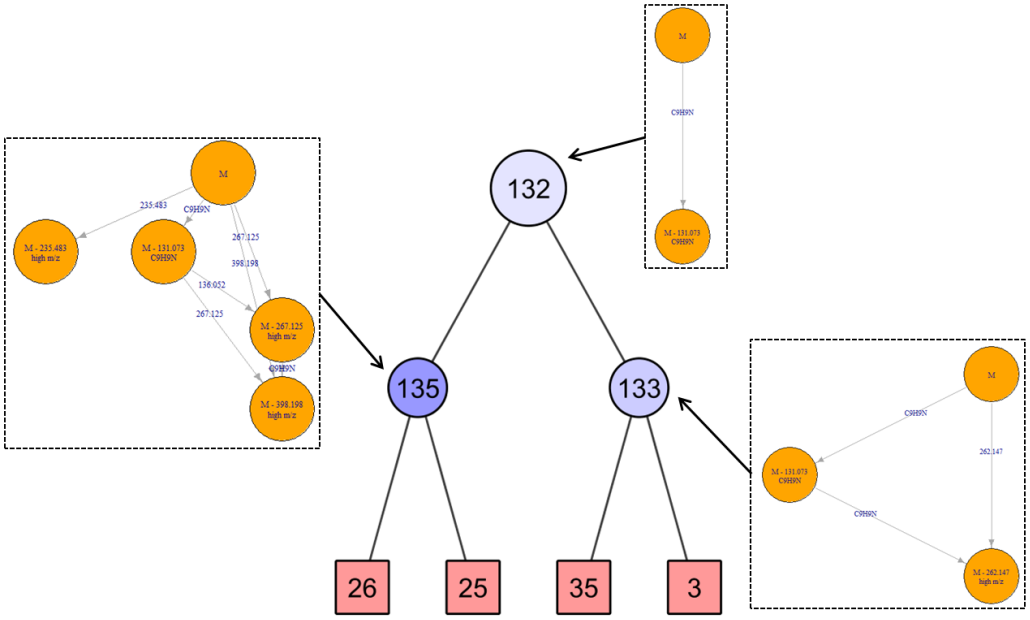


Figure 12.1: **Representation of patterns within the Hass Diagram (PenicilliumDIA dataset).** Pink squares correspond to Losses Graphs . Blue circles indicate AFGs: bigger nodes define super-patterns, and darker nodes correspond to patterns with a higher chemical score. The AFG corresponding to each nodes is shown.

Although G is very large, a more sparse representation keeping the same amount of information can be obtained by using the transitive reduction of G. The transitive reduction of G is another directed graph H with the same vertices and as few edges as possible, such that if there is a path from vertex v to vertex w in G, then there is also such a path in H. This graph is known as the Hass Diagram of ad ordered set and is defined for any finite partially ordered set. An example of Hass Diagram corresponding to the full set of patterns from the **PenicilliumDIA** dataset is shown in Figure 12.2.

Although the full Hass Diagram is difficult to interpret, a zoom on specific parts provides a useful hierarchical representation of the patterns and of their structural content (Figure 12.1).

Many approaches have been proposed to summarize big graphs (a recent review of such algorithms may be found in `arXiv:1612.04883v3`). Such methods, however, do not take into account the particularity of Hass Diagram and are not adapted to our problem. A more closely related approach is the lattice reduction, derived from Formal Concept Analysis (see the review in Dias and Vieira 2015). Nevertheless, these methods tend to focus on the most frequent patterns, whereas highly specific patterns are also important in our applications; in addition, they do not discriminate between

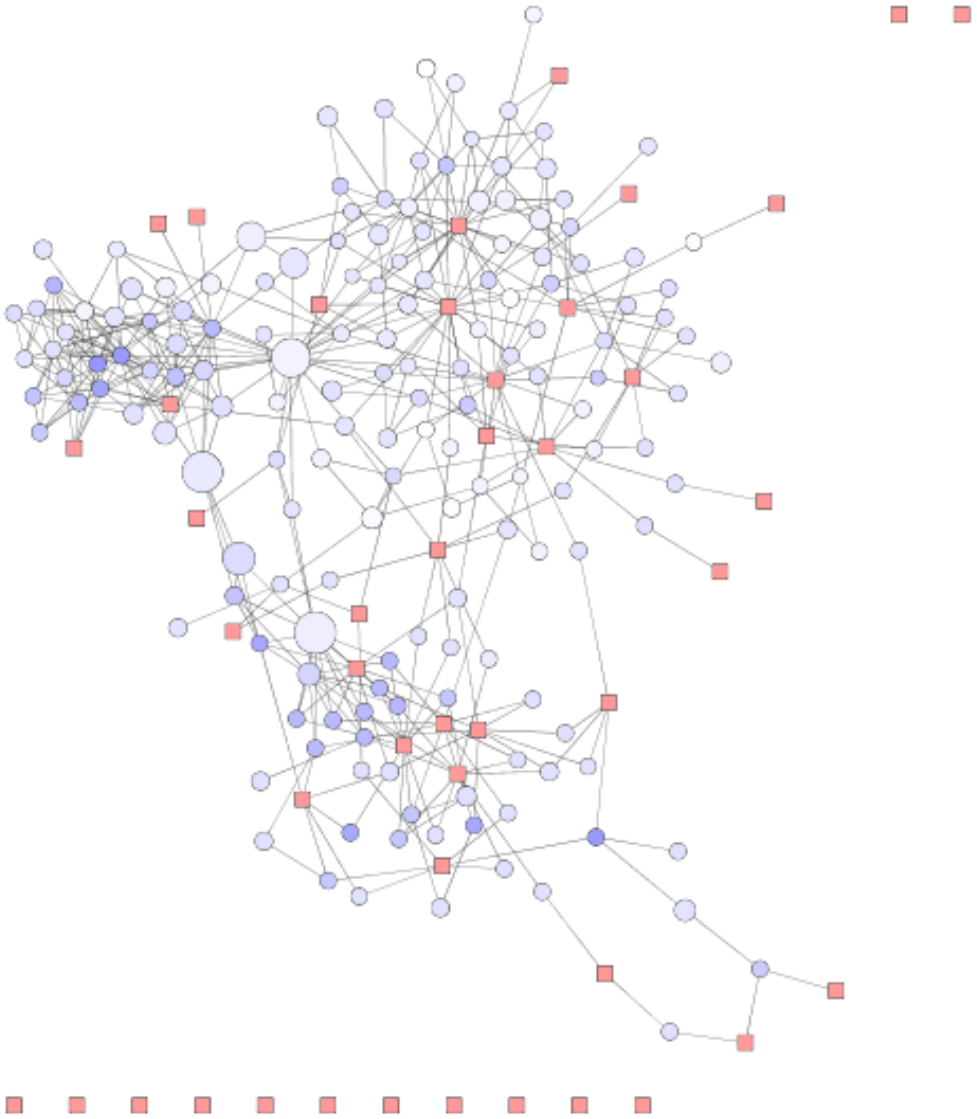


Figure 12.2: **Full Hass Diagram (PenicilliumDIA dataset).** The empty graph is omitted for visibility purpose. Pink squares correspond to Losses Graphs . Blue circles indicate AFGs.

the upper nodes in the hierarchy and the leafs, the latter being Losses Graphs which must not be removed from the Hass Diagram.

We have therefore developed a new approach to reduce our Hass Diagram to k nodes (where the number of selected patterns, k, is specified by the user), which:

1. keeps all the nodes corresponding to Losses Graphs .
2. keeps as many of the similarities between the Losses Graphs as possible.

12.1.1 Problem formalization

The proposed approach can be summarized as follows:

1. A score based on the Hass Diagramis defined between any pair of Losses Graphs (as a measure of their similarity): SP.
2. The matrix of scores for all Losses Graphs is computed: M.
3. A heuristic approach is used to reduce the number of nodes to k while minimizing the differences with the initial M.

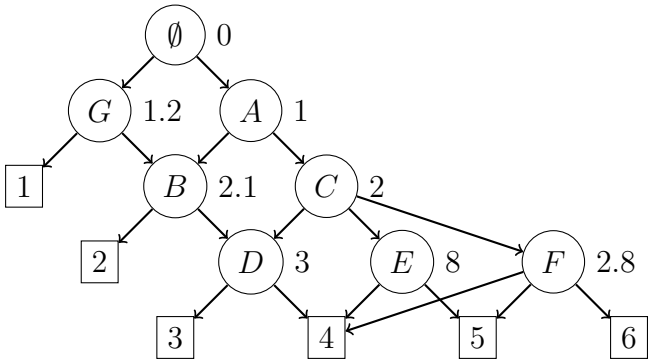


Figure 12.3: **Example of a Hass Diagram to be reduced.** I nodes (Losses Graphs) are shown as squares, and R nodes (AFGs) as circles. The score of each element from R is indicated at the right of the node

Consider a Hass Diagram $H_{D,C}$ generated from a set of Losses Graphs D and a set of closed AFGs C. For commodity purpose we denote it by H.Let us write $I(H)$ the vertices of H corresponding to Losses Graphs , and $R(H)$ the vertices corresponding to elements of C plus the empty graph.

We assume that a non negative score S may be assigned to each pattern (as detailed below in Section 12.1.2). S is a function from $R(H) \rightarrow \mathbb{R}^+$, with $S(\emptyset) = 0$. Moreover if X and Y are AFGs such that if $X \subseteq Y$ then $S(X) < S(Y)$.

Now, for each pair of Losses Graphs , we define a score SP as a measure of the similarity between them: it equals to the best score S among their common subgraphs. As $\emptyset \in H$, SP is always defined and is non negative. More formally let us define SP_H the function from $I(H) \times I(H)$ to $\mathbb{R}^+$ as:

$$SP_H(A, B) = \max_{u \in pred(A) \cap pred(B)} S(u)$$

Where $pred(A)$ denotes all the predecessors from A in H. For example, in Figure 12.3, $SP(3, 4) = 3$, $SP(5, 6) = 2.8$, $SP(4, 6) = 2.8$, and $SP(1, 5) = 0$.

SP may then be used to build a matrix of scores for all the Losses Graphs :

$$M_H[j, k] = SP_H(i^{-1}(j), i^{-1}(k))$$

where $i : I(H) \rightarrow [1, \ldots, |I(H)|]$ is an index associated to each Losses Graph .

$M_{H_{D,C}}$ is our measure of the structural similarity between any pair of Losses Graphs from D, based on the detected frequent patterns. This matrix is defined for any set of closed patterns. It is symmetric.

Our Hass Diagram reduction objective can now be viewed as the decrease of the number of patterns with the lowest impact on M_H. This can be formulated as follows: given a set of Losses Graphs D, a set of closed patterns C, and a specified number of patterns k, find C_o , a subset from C of size k, such that:

$$C_o = \underset{\substack{C' \subset C \\ |C'|=k}}{\operatorname{argmin}} \underbrace{RMSE(M_{H_{D,C}}, M_{H_{D,C'}})}_{F}$$

where $RMSE$ is the Residual Mean Squared Error between the two matrices. Since F can be seen as a cost function to optimize, finding C_o is a combinatorial optimization task.

12.1.2 Assigning a chemical score to an AFG

Our metric is derived from the scoring of fragmentation trees described in Rasche et al. 2010. In MS/MS spectra from metabolites, some frequent losses convey little

information (for example, the loss of H_2O, or NH_3). In contrast, some fragmentations are more determinant for the chemical and biological interpretations (e.g. in the **PenicilliumDIA** dataset, the loss of mass 147.068 corresponds to Phenylalanine). Moreover, fragments of higher mass tend to be more specific. It is thus possible to score each loss based on its mass and putative elemental formula. Compared to Rasche et al. 2010, we added a term to penalize the frequent losses in the dataset. For a loss L of mass m_L and frequency f_L in the initial set of Losses Graphs :

$$score(L) = sigmoid(f_L) \times P_{known} - P_{mono} + log_{100}(m_L)$$

with $sigmoid(f) = 1/(1 + e^{2(f-0.5)})$, P_{known} a fixed term promoting known losses, and P_{mono} a fixed term penalizing mono-atomic masses such as C_2 which are highly improbable, these terms may be furnished by the user. At the moment this score is still a work in progress, so I do not detail the value of these parameters.

The score of an arborescence is defined as the sum of the scores from its edges. The spanning arborescence with the minimum score is used to define the score of the AFG. This minimal spanning arborescence can be computed by selecting the incoming edge with the minimum score for each vertex of the each AFG.

12.1.3 Development of a greedy algorithm for pattern selection

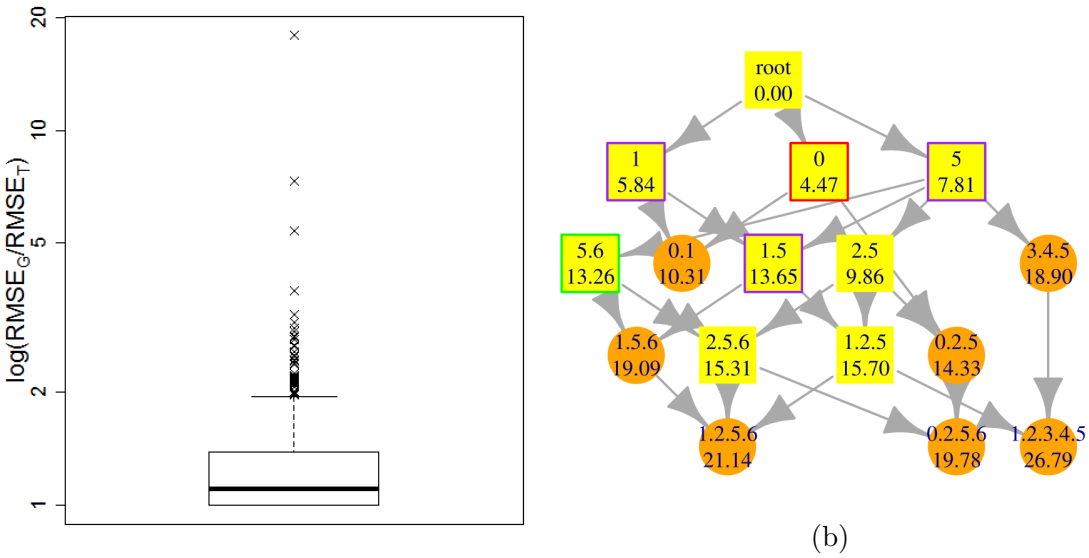
A greedy algorithm was implemented to select k patterns while minimizing the modifications of the $M_{H_{D,C}}$ matrix of spectra similarities. The algorithm was tested on simulated small scale examples, where the optimal subset could be computed by brute force search (see below). Since the selected subsets sometimes differ from the optimal on these simulations, and since a bias towards large patterns was observed with the large (real) spectral collections, the algorithm is not described here, and a more complex approach need to be developed.

Evaluation on simulated datasets

As the generation of a set of random Losses Graphs and AFGs is difficult, we focused on itemsets, which are also partially ordered sets (a Hass Diagram can therefore be built). We generated 400 sets of itemsets (corresponding to 400 simulated databases D) as follows: for each database, between 6 and 10 itemsets (corresponding to the

Losses Graphs) were created by combining up to 6 items (the losses). The set of the closed frequent itemsets (corresponding to the AFGs in C) was mined using an implementation of the CHARM algorithm (Mohammed J. Zaki and Hsiao 2002). A score was drawn for each of the 6 items using a normal distribution of mean 5 and standard deviation 2, and the score of each closed frequent itemset was computed as the sum of the scores from the included items.

For each of these 400 databases and their associated closed frequent itemsets, the Hass Diagram was built (Figure 12.4b) and M was computed. Then for $k \in \{2, 3, 4, 5\}$, if the number of frequent itemsets in the database was more than k, C_o was found, and the $RMSE$ between the two matrices was stored and compared to the greedy algorithm. This led to 1290 pairs of RMSE values obtained with the optimal selection ($RMSE_T$) and the greedy algorithm ($RMSE_G$; Figure 12.4a).



(a) (b)

Figure 12.4: **Results on simulated itemsets.** b): simulated Hass Diagram; the node selected only by the optimal solution C_o for $k = 4$ is in green, the node selected by both the optimal solution and the greedy algorithm are in purple, and the node selected by the greedy solution only is in red. a): boxplot of the log ratio between the RMSE corresponding to the two approaches, for the 1290 simulations.

As shown by these simulations, selection by the greedy algorithm is suboptimal (even with these small number of items). Observation of the optimum solutions, however, seems to validate our optimization criterion. The challenge is therefore to develop a better algorithm to find the optimal, or to relax the objective function so that the optimum is easier to compute.

12.2 Perspectives

12.2.1 Limits of patterns mining methods

MineMS2 is a method to mine exact subgraphs, to our knowledge the only existing method of exact pattern mining which has been used in MineMS2 is the MEtabolite SubStructure Auto- Recommender (MESSAR) software (Mrzic et al. 2017) mining itemsets. Both of them only mine exact patterns, in contrast to the patterns mined by MS2LDA (Hooft, Wandy, Barrett, et al. 2016). In both MineMS2 and MESSAR, a large number of patterns is detected. This may be explained by the skewness of the label distribution (Figure 12.5): many patterns may appear randomly (this is the reason why we penalize frequent losses in the Section 12.1).

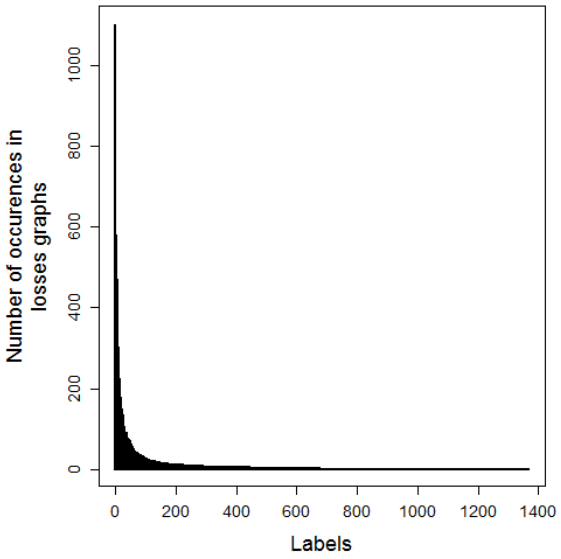


Figure 12.5: **Distribution of labels in the set of Losses Graphs (LemmDB dataset)**. The 6 most common losses labels are: H_2O, CH_2O_2, C_2H_2, C_2, C_2H_2O, H_3N.

The high number of patterns may also be the result from the partial fragmentation of the molecules: the fragmentation patterns from a substructure common to several compounds may differ according to the global structure of these compounds. If, for example, one of the compound is much bigger than the other one, the energy in the collision cell may not be sufficient to achieve the fragmentation up to the substructure of interest. Such a bias may be partially overcome by using the sum of spectra from different energies, but this still increases the number of different fragmentation patterns for a single substructure.

Finally while the utility of non-exact patterns for direct annotation has been demonstrated in Hooft, Wandy, Barrett, et al. 2016, the use of exact patterns seems more difficult as they are more dataset dependent and less flexible. On the other hand inexact pattern as proposed in MS2LDA are more difficult to interpret, and dependent of less intuitive parameters: during my experiments, I had a very hard time parameterizing MS2LDA caused by the fact that it includes multiple hyper-parameters. In practice this led to the detection of pattern including H_2O with a very high probability and hundreds of other losses/fragments with a probability inferior to 2%. In contrast, the parameters of MineMS2 are limited to the number of fragments considered and the accuracy of the mass spectrometer.

Besides the limits of the exact pattern mining methods, the MineMS2 addresses some inherent challenges, notably due to the discretization process: in practice we noted that the discretization of a set of masses differences is way less trivial than the discretization of a set of masses. We therefore added a step of label merging in the workflow. As this labeling step remains difficult, property 2 ("An edge may be removed from any connected graph which remain connected without changing his support") might not be always satisfied. In practice, however, we checked that the property was valid in 95% of the Losses Graphs (**LemmDB**dataset).

Furthermore the formula generation may possibly be improved by removing some really improbable formulas: for example C_2 is a valid formula in terms of a molecular graph but requires a quadruple bond between the two carbons, a case that may never occur in nature.

Finally the most obvious flaw of MineMS2 is the inherent exponential nature of the mining process, which makes it less scalable than the method proposed by Hooft, Wandy, Barrett, et al. 2016: we have however shown that MineMS2 could be run in a few minutes on datasets of realistic size (e.g. 700 spectra). To further increase the size of the input set of spectra, the number of peaks by spectrum may be reduced ($maxFrag$ argument of MineMS2), but this may result in missing some important patterns in spectra with a high number of peaks.

12.2.2 Extensions of MineMS2

Mining both fragments and losses

Although the mined patterns are currently limited to losses, the software may be extended to also include information about the fragments. This would be of high value for molecules which are characterized by a single fragment (e.g. lipid families). In the current version, MineMS2 mines patterns consisting of fragments only if there is a loss between them (i.e., if the formula of the first fragment is included in the formula of the second fragment). Although the inclusion of additional fragment-based patterns seems difficult, it is possible to integrate them into the graphs by discretizing the masses, and considering them as vertex labels. However, such an approach would probably be computationally intractable apart from datasets of very limited size. The direct mining of both losses and fragments seems therefore difficult with the proposed methodology.

However, the mining of fragments alone is possible with the methodology by simply modifying the Losses Graphs construction process. An artificial peak of m/z 0 could be added, and the full set of mass difference would be constructed. Then all the negative mass differences would be kept and their absolute value would be discretized using the proposed algorithm. The mass difference originating from the 0 peak would be then discretized separately, and the formula generation would need to take into account the mass of the added adduct ion, as the fragments are charged. This would lead to a set of graphs similar to Losses Graphs where the precursor would be replaced by the m/z 0 peak and the topological order would be inverted however all the edges between fragments ions other than the precursor peak and the 0 peak would be inverted. This set of graphs would have exactly the same properties than the Losses Graphs and could be mined using the same algorithms. While the combination of the two sets of patterns is far from trivial, this approach is, to my opinion, the most realistic to mine simultaneously fragments and losses. Moreover the combined set of both type of patterns may still be represented by a Hass Diagram, and the approach of Hass Diagram reduction described in Section 12.1 would therefore be applicable.

Another possible extension of the MineMS2 software is the development of a visualization module, which would offer a more interactive exploration of the set of patterns. In particular, targeted queries would be available, such as the isolation of patterns with a specified loss, or the finding of the biggest pattern within a spectrum. Pattern visualization would be very useful for biologists, since the task of MS/MS analysis and *de*

novo identification currently requires a large amount of manual work, and this module would therefore considerably increase the value of the proposed methodology. Some exploration features have already been implemented in C++ in the R package, and the addition of a web interface would be useful.

12.2.3 Interpretability of MineMS2 derived patterns

MineMS2 has some strong advantages over alternative MS/MS exploration approaches, due to the pattern mining strategy: in particular, patterns are more interpretable than the single similarity measure computed by GNPS (M. Wang et al. 2016), and more related to the physical fragmentation process than the discrete distribution mined by MS2LDA (Hooft, Wandy, Barrett, et al. 2016).

While the pattern mining and the structure prediction objectives have been addressed separately up to now, an interesting approach to combine both strategies has been proposed in MESSAR, association rule mining is used to relate spectral features (fragments and losses) to substructures. The proposed methodology, however, does not take into account the fact that losses may be consecutive. A similar approach could potentially be used with MineMS2, by considering each pattern as an item and each Losses Graph as an itemset, thus finding fragmentation patterns corresponding to known substructures.

12.2.4 Potential coupling of MineMS2 to *in silico* fragmentation methods

As stated in introduction, MineMS2 was developed to work on a graph representation from the mass spectrum. Since graphs are also used in *in silico* fragmentation models, pattern mining could be included in such approaches. The main *in silico* fragmentation methods (Ruttkies et al. 2016; Böcker and Dührkop 2016; Tsugawa, Kind, et al. 2016) currently test a vast number of possible fragmentations for each compound, and then select the best scoring one according to some specified criterion (see section 9.2). By using MineMS2, the detection of meaningful patterns common to several spectra would suggest common fragmentation subtrees: therefore, a constraint of equality between the subtrees may be included into the scoring process. This would vastly differ from the usual in-silico fragmentation methods, which always consider all the spectra independently, and therefore ignore the fact that spectra acquired in a single acquisition will probably share a higher degree of similarity.

Chapter 13

Conclusion

In this thesis, we have described the two innovative contributions we developed and implemented for the high-throughput processing and annotation of metabolomics data from high-resolution mass spectrometry.

In Part II, we proposed the first freely available workflow for the processing of FIA-HRMS data, based on an approximation of the physical processes affecting an the EIC signal (Delabrière et al. 2017). Features detection and quantification were shown to be as accurate and much faster than the manual processing by chemical experts, and to outperform reference algorithms for LC-MS data such as centWave. The resulting software, named proFIA, is available as an R package on Bioconductor (https://doi.org/10.18129/B9.bioc.proFIA), and as a Galaxy module within the Workflow4Metabolomics online platform. proFIA features several new algorithms including an indicator of matrix effect, and an estimation of the sample peak, which should also be helpful for experimenters when optimizing the analytical setup.

To our knowledge, matrix effect is the main limiting factor of FIA-MS, however, while a lot of possible factors have been proposed (see section 5.1.2), its effect on a full biological sample has not been evaluated. The raw matrix effect indicator proposed by proFIA is a first step in this direction and could potentially be used to evaluate the matrix effect in function of the characteristics of the molecules. This could have a great impact on the analytical workflow at many level. For example it would potentially allow to use FIA-MS in targeted experiments if the targeted compounds are known to ionize correctly. Similarly while the initial discovery of a biomarker in LC-MS is better because of its higher sensitivity, its validation could potentially be performed in FIA-MS if its ionization is known to be sufficient. A better understanding of matrix

effect could also lead to improvement in the DI-MS technique, the other other widely used high-throughput approach.

More generally, proFIA by mimicking the physical process allows a better insight into the quality of the analytical protocol by providing a quick classification of the observed signals (See figure 6.10). This kind of approach, which allows a quick evaluation of an experimental setup could potentially be also used in LC-MS with different metrics. While in FIA the desired peak shape is an asymmetric peak, in LC-MS the desired peak shape is a Gaussian, and generally the peak picking approach has been tailored to detect Gaussian peaks. However non Gaussian peak exhibiting tailing or fronting may arise from various physico-chemical factors, or from the internal processing of the mass spectrometer (Wahab et al. 2017) . These peaks are harder to quantify and generally more difficult to reach, therefore a classification of the peak shapes in LC-MS acquisition could potentially allow an optimization of the acquisition protocol to optimize signal shape and increase the peak picking software efficiency.

In Part III, we developed a new approach to extract structural similarities within a set of MS/MS spectra. The strategy relies on a new representation of fragmentation spectra as graphs which does not require prior knowledge of the elemental formula from the precursor, as well as a dedicated Frequent Subgraph Mining suite of algorithms to efficiently extract the set of closed frequent subgraphs. Input high-quality spectra may be generated by the MS2process module that we developed. The mining workflow itself, implemented as the MineMS2 R package, was shown to successfully mine two large collections of spectra (DIA experiment and in-house database). As the set of frequent patterns is large, a method is currently being developed to summarize this set. As soon as this final step is validated, the manuscript and the package will be submitted.

The proposed pattern mining approaches has multiple potential applications. First the concept of patterns, while having some inherent limitations discussed previously, has a high potential of coupling with machine learning approaches. A set of patterns could be used to define a set of features upstream of a machine learning approach. Such features would potentially be more interpretable than the usual features or kernel used, as they would be extracted directly from similar data. Moreover in the case of MineMS2, the proximity with the chemical fragmentation makes them more interpretable. Such interpretable features are really important, especially to allow a better use of these computational tool in the metabolomics community.

While the two parts address distinct data analysis questions, the underlying strategy was similar: building a model mimicking the physical process as much as possible,

and providing new interpretable informations, such as the sample peak in FIA or the fragmentation patterns in MS2 annotation. Furthermore, both proFIA and MineMS2 packages have been successfully combined to automatically process the **LemmDB** dataset, which consists of MS/MS spectra from standards acquired with an FIA-HRMS protocol.

This PhD therefore provides solutions to the community on two major challenges for high-throughput metabolomics: the preprocessing and the annotation.

References

Aberg, K. Magnus et al. (2009). "Feature detection and alignment of hyphenated chromatographic–mass spectrometric data Extraction of pure ion chromatograms using Kalman tracking". In: *Journal of Chromatography A*. DOI: 10.1016/j.chroma.2008.03.033.

Agrawal, Rakesh and Ramakrishnan Srikant (1994). "Fast Algorithms for Mining Association Rules in Large Databases". In: *Proceedings of the 20th International Conference on Very Large Data Bases*. VLDB '94. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., pp. 487–499.

Allen, Felicity, Russ Greiner, and David Wishart (2014). "Competitive fragmentation modeling of ESI-MS/MS spectra for putative metabolite identification". In: *Metabolomics* 11.1, pp. 98–110. DOI: 10.1007/s11306-014-0676-4.

Allen, Jess et al. (2003). "High-throughput classification of yeast mutants for functional genomics using metabolic footprinting". In: *Nature Biotechnlogy*. DOI: 10.1038/nbt823.

Alonso, Arnald, Sara Marsal, and Antonio Julià (2015). "Analytical Methods in Untargeted Metabolomics: State of the Art in 2015". In: *Frontiers in Bioengineering and Biotechnology* 3, p. 23. DOI: 10.3389/fbioe.2015.00023.

Anderle, Markus et al. (2004). "Quantifying reproducibility for differential proteomics: noise analysis for protein liquid chromatography-mass spectrometry of human serum". In: *Bioinformatics*. DOI: 10.1093/bioinformatics/bth446.

Andreev, Victor P. et al. (2003). "A Universal Denoising and Peak Picking Algorithm for LC-MS Based on Matched Filtration in the Chromatographic Time Domain". In: *Analytica Chemistry*. DOI: 10.1021/ac0301806.

Armitage, Emily G. and Coral Barbas (2014). "Metabolomics in cancer biomarker discovery: Current trends and future perspectives". In: *Journal of Pharmaceutical and Biomedical Analysis* 87. Review Papers on Pharmaceutical and Biomedical Analysis 2013, pp. 1–11. DOI: https://doi.org/10.1016/j.jpba.2013.08.041.

Armitage, Emily Grace et al. (2015). "Missing value imputation strategies for metabolomics data". In: *Electrophoresis* 36.24, pp. 3050–3060. DOI: 10.1002/elps.201500352.

Asai, Tatsuya et al. (2002). "Efficient Substructure Discovery from Large Semi-structured Data". In: *Proceedings of the 2002 SIAM International Conference on Data Mining*, pp. 158–174. DOI: 10.1137/1.9781611972726.10.

Ayed, Rihab et al. (2016). *A list of FSM Algorithms and available Implementations in Centralized Graph Transaction Databases*. Technical Report. Université Claude Bernard Lyon.

Banerjee, Shibdas and Shyamalava Mazumdar (2012). "Electrospray Ionization Mass Spectrometry: A Technique to Access the Information beyond the Molecular Weight of the Analyte". In: *International Journal of Analytical Chemistry*. DOI: 10.1155/2012/282574.

Beckmann, Manfred, David P. Enot, et al. (2007). "Representation, Comparison, and Interpretation of Metabolome Fingerprint Data for Total Composition Analysis and Quality Trait Investigation in Potato Cultivars". In: *Journal of agricultural and food chemistry*. DOI: 10.1021/jf0701842.

Beckmann, Manfred, David Parker, et al. (2008). "High-throughput, nontargeted metabolite fingerprinting using nominal mass flow injection electrospray mass spectrometry". In: *Nature Protocols* 3, pp. 486–.

Benton, H. Paul et al. (2015). "Autonomous Metabolomics for Rapid Metabolite Identification in Global Profiling". In: *Analytical Chemistry* 87.2, pp. 884–891. DOI: 10.1021/ac5025649.

Berthod, Alain (1991). "Mathematical Series for Signal Modeling Using Exponentially Modified Functions". In: *Analytical Chemistry* 63, pp. 1879–1884.

Bhuiyan, M. A. and M. Al Hasan (2015). "An Iterative MapReduce Based Frequent Subgraph Mining Algorithm". In: *IEEE Transactions on Knowledge and Data Engineering* 27.3, pp. 608–620. DOI: 10.1109/TKDE.2014.2345408.

Blei, David M, Andrew Y Ng, and Michael I Jordan (2003). "Latent dirichlet allocation". In: *Journal of machine Learning research* 3.Jan, pp. 993–1022.

Böcker, Sebastian and Kai Dührkop (2016). "Fragmentation trees reloaded". In: *Journal of Chemoinformatics* 8.5. DOI: 10.1186/s13321-016-0116-8.

Böcker, Sebastian, Matthias C. Letzel, et al. (2009). "SIRIUS: decomposing isotope patterns for metabolite identification." In: *Bioinformatics* 25.2, pp. 218–224. DOI: 10.1093/bioinformatics/btn603.

Böcker, Sebastian and Zsuzsanna Lipták (2005). "Efficient Mass Decomposition". In: *ACM Symposium on Applied Computing*. DOI: 10.1145/1066677.1066715.

Böcker, Sebastian and Florian Rasche (2008). "Towards de novo identification of metabolites by analyzing tandem mass spectra". In: *Bioinformatics*. DOI: 10.1093/bioinformatics/btn270.

Bogaert, B van den, H F M Boelens, and H C Smlt (1993). "Quantification of chromatographic data using a matched filter: robustness towards noise model errors". In: *Analytica Chemica Acta* 274, pp. 87–97. DOI: 10.1016/0003-2670(93)80608-N.

Boudah, S. et al. (2014). "Annotation of the human serum metabolome by coupling three liquid chromatography methods to high-resolution mass spectrometry". In: *Journal of Chromatography B*. DOI: 10.1016/j.jchromb.2014.04.025.

Boya P., Cristopher A. et al. (2017). "Imaging mass spectrometry and MS/MS molecular networking reveals chemical interactions among cuticular bacteria and pathogenic fungi associated with fungus-growing ants". In: *Scientific Reports* 7.1, pp. 5604–. DOI: 10.1038/s41598-017-05515-6.

Breiman, Leo (2001). "Random Forests". In: *Machine Learning* 45.1, pp. 5–32. DOI: 10.1023/A:1010933404324.

Broadhurst, David I. and Douglas B. Kell (2006). "Statistical strategies for avoiding false discoveries in metabolomics and related experiments". In: *Metabolomics* 2.4, pp. 171–196. DOI: 10.1007/s11306-006-0037-z.

Brooks, Stephen H and John G Dorsey (1990). "Moment analysis for evaluation of flow-injection manifolds". In: *Anaytical Chemistry* 229, pp. 35–46. DOI: 10.1016/S0003-2670(00)85107-7.

Brouard, Céline et al. (2016). "Fast metabolite identification with Input Output Kernel Regression". In: *Bioinformatics* 32.12, pp. i28–i36. DOI: 10.1093/bioinformatics/btw246.

Buescher, Joerg M et al. (2015). "A roadmap for interpreting (13)C metabolite labeling patterns from cells". In: *Current opinion in biotechnology* 34, pp. 189–201. DOI: 10.1016/j.copbio.2015.02.003.

Cameron, A. E. and D. F. Eggers Jr. (1948). "An Ion "Velocitron"". In: *Review of Scientific Instruments* 19 (605).

Caspi, Ron et al. (2018). "The MetaCyc database of metabolic pathways and enzymes". In: *Nucleic Acids Research* 46.D1, pp. D633–D639. DOI: 10.1093/nar/gkx935.

Castaldi, Peter J., Issa J. Dahabreh, and John P.A. Ioannidis (2011). "An empirical assessment of validation practices for molecular classifiers". In: *Briefings in Bioinformatics* 12.3, pp. 189–202. DOI: 10.1093/bib/bbq073.

Cech, Nadja B. and Christie G. Enke (2001). "Partical implacations of some recent studies in electrospray ionization fundamentals". In: *Mass Spectrometry Reviews*. DOI: 10.1002/mas.10008.

Chi, Yun, Yirong Yang, and R. R. Muntz (2004). "HybridTreeMiner: an efficient algorithm for mining frequent rooted trees and free trees using canonical forms". In: *Proceedings. 16th International Conference on Scientific and Statistical Database Management, 2004.* Pp. 11–20. DOI: 10.1109/SSDM.2004.1311189.

Cleveland, William S. (1979). "Robust Locally Weighted Regression and Smoothing Scatterplots". In: *Journal of the American Statistical Association* 74 (368).

Conley, Christopher J. et al. (2014). "Massifquant: open-source Kalman filter-based XC-MS isotope trace feature detection". In: *Bioinformatics*. DOI: 10.1093/bioinformatics/btu359.

Cordella, L. P. et al. (2001). "An improved algorithm for matching large graphs". In: *In: 3rd IAPR-TC15 Workshop on Graph-based Representations in Pattern Recognition, Cuen*, pp. 149–159.

Cuevas, Erik, Daniel Zaldivar, and Raul Rojas (2005). *Kalman filter for vision tracking*. Technical Report B 05-12. Freie Universität Berlin, Institut für Informatik.

Danielsson, Rolf, Dan Bylund, and Karin E. Markides (2002). "Matched filtering with background suppression for improved quality of base peak chromatograms and mass spectra in liquid chromatography–mass spectrometry". In: *Analytica Chimica Acta* 454, pp. 167–184. DOI: 10.1016/S0003-2670(01)01574-4.

Darga, Paul T. et al. (2004). "Exploiting Structure in Symmetry Detection for CNF". In: *Proceedings of the 41st Annual Design Automation Conference*. DAC '04. San Diego, CA, USA: ACM, pp. 530–534. DOI: 10.1145/996566.996712.

Daubechies, Ingrid (1992). *Ten Lectures on Wavelets*.

Delabrière, Alexis et al. (2017). "proFIA: a data preprocessing workflow for flow injection analysis coupled to high resolution mass spectrometry". In: *Bioinformatics* 33.23, pp. 3767–3775. DOI: 10.1093/bioinformatics/btx458.

Dias, Sérgio M. and Newton J. Vieira (2015). "Concept lattices reduction: Definition, analysis and classification". In: *Expert Systems with Applications* 42.20, pp. 7084–7097. DOI: 10.1016/j.eswa.2015.04.044.

Draper, John et al. (2009). "Metabolite signal identification in accurate mass metabolomics data with MZedDB, an interactive m/z annotation tool utilising predicted ionisation behaviour 'rules'". In: *BMC bioinformatics*. DOI: 10.1186/1471-2105-10-227.

Draper, J. et al. (2013). "Flow infusion electrospray ionisation mass spectrometry for high throughput, non-targeted metabolite fingerprinting: a review". English. In: *Metabolomics* 9.1, pp. 4–29. DOI: 10.1007/s11306-012-0449-x.

Du, P., W.A. Kibbe, and S.M. Lin (2006). "Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching". In: *Bioinformatics* 22.17, pp. 2059–2065. DOI: 10.1093/bioinformatics/btl355.

Dunn, Warwick B et al. (2011). "Procedures for large-scale metabolic profiling of serum and plasma using gas chromatography and liquid chromatography coupled to mass spectrometry". In: *Nature Protocols* 6, pp. 1060–1083. DOI: 10.1038/nprot.2011.335.

Dunn, Warwick B. et al. (2012). "Mass appeal: metabolite identification in mass spectrometry-focused untargeted metabolomics". In: *Metabolomics*. DOI: 10.1007/s11306-012-0434-4.

El Islem Karabadji, Nour, Sabeur Aridhi, and Hassina Seridi (2016). "A Closed Frequent Subgraph Mining Algorithm in Unique Edge Label Graphs". In: *Machine Learning and Data Mining in Pattern Recognition*. Ed. by Petra Perner. Cham: Springer International Publishing, pp. 43–57.

Eli, Grushka, Myers Marcus N., and Giddings J. Calvin (1970). "Moments Analysis for the Discernment of Overlapping Chromatographic Peaks". In: *Analytical Chemistry* 42.1, pp. 21–26. DOI: 10.1021/ac60283a015.

Enke, Christie G. (1997). "A Predictive Model for Matrix and Analyte Effects in Electrospray Ionization of Singly-Charged Ionic Analytes". In: *Analytical Chemistry* 69, pp. 4885–4893. DOI: 10.1021/ac970095w.

Enot, D.P. et al. (2008). "Preprocessing, classification modeling and feature selection using flow injection electrospray mass spectrometry metabolite fingerprint data". In: *Nature Protocols* 3.3, pp. 446–470. DOI: 10.1038/nprot.2007.511.

Eppstein, David (1995). "Subgraph Isomorphism in Planar Graphs and Related Problems". In: *Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA '95. San Francisco, California, USA: Society for Industrial and Applied Mathematics, pp. 632–640.

Fahy, Eoin et al. (2007). "LIPID MAPS online tools for lipid research". In: *Nucleic Acids Research* 35. DOI: 10.1093/nar/gkm324.

Favé, Gaëlle et al. (2011). "Development and validation of a standardized protocol to monitor human dietary exposure by metabolite fingerprinting of urine samples". In: *Metabolomics*. DOI: 10.1007/s11306-011-0289-0.

Felinger, Attila (1998). "Peak shape analysis". In: *Data Analysis and Signal Processing in Chromatography*. Ed. by B.G.M. Vandeginste and S.C. Rutan. Vol. 21. Elsevier. Chap. Peak shape analysis, pp. 97–119.

Fenaille, Francois et al. (2017). "Data acquisition workflows in liquid chromatography coupled to high resolution mass spectrometry-based metabolomics: Where do we stand?" In: *Journal of Chromatography A*. DOI: 10.1016/j.chroma.2017.10.043.

Foley, Joe P. (1987). "Equations for Chromatographic Peak Modeling and Calculation of Peak Area". In: *Analytical Chemistry* 59, pp. 1984–1987. DOI: 10.1021/ac00142a019.

Frainay, Clément and Fabien Jourdan (2017). "Computational methods to identify metabolic sub-networks based on metabolomic profiles". In: *Briefings in Bioinformatics* 18.1, pp. 43–56. DOI: 10.1093/bib/bbv115.

Frank, Ildiko E. and Silvia Lanteri (1989). "Classification models: Discriminant analysis, SIMCA, CART". In: *Chemometrics and Intelligent Laboratory Systems* 5.3, pp. 247–256. DOI: 10.1016/0169-7439(89)80052-8.

French, William R. et al. (2014). "Wavelet-Based Peak Detection and a New Charge Inference Procedure for MS/MS Implemented in ProteoWizard's msConvert". In: *Journal of proteome research*. DOI: 10.1021/pr500886y.

F.R.S., Lord Rayleigh (1882). "On the equilibrium of liquid conducting masses charged with electricity". In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*.

Fuhrer, Tobias et al. (2011). "High-Throughput, Accurate Mass Metabolome Profiling of Cellular Extracts by Flow Injection Time-of-Flight Mass Spectrometry". In: *Analytical Chemistry*. DOI: 10.1021/ac201267k.

Fukushima, Atsushi et al. (2014). "Metabolomic Characterization of Knockout Mutants in Arabidopsis: Development of a Metabolite Profiling Database for Knockout Mutants in Arabidopsis". In: *Plant Physiology* 165.3, pp. 948–961. DOI: 10.1104/pp.114.240986.

Garey, Michael R. and David S. Johnson (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman & Co.

Gedcke, D. A. (2001). "Dealing with Dead Time Distortion in a Time Digitizer". In: *ORTEC application note*.

Gentleman, R.C. et al. (2004). "Bioconductor: open software development for computational biology and bioinformatics". In: *Genomic Biology* 5, R80. DOI: 10.1186/gb-2004-5-10-r80.

Giacomoni, Franck et al. (2014). "Workflow4Metabolomics: A collaborative research infrastructure for computational metabolomics". In: *Bioinformatics*. DOI: 10.1093/bioinformatics/btu813.

Godzien, Joanna et al. (2015). "Controlling the quality of metabolomics data: new strategies to get the best out of the QC sample". In: *Metabolomics* 11.3, pp. 518–528. DOI: 10.1007/s11306-014-0712-4.

Gromski, Piotr S. et al. (2014). "A comparative investigation of modern feature selection and classification approaches for the analysis of mass spectrometry data". In: *Analytica Chimica Acta*. DOI: 10.1016/j.aca.2014.03.039.

Gross, Jonathan L., Jay Yellen, and Ping Zhang (2013). *Handbook of Graph Theory*. Ed. by Jonathan L. Gross, Jay Yellen, and Ping Zhang.

Gross, Jürgen H (2011). *Mass Spectrometry: A textbook (2nd Ed)*. Ed. by Springer editor board. Springer.

Gudes, E., S. E. Shimony, and N. Vanetik (2006). "Discovering Frequent Graph Patterns Using Disjoint Paths". In: *IEEE Transactions on Knowledge and Data Engineering* 18.11, pp. 1441–1456. DOI: 10.1109/TKDE.2006.173.

Guida, Riccardo Di et al. (2016). "Non-targeted UHPLC-MS metabolomic data processing methods: a comparative investigation of normalisation, missing value imputation, transformation and scaling". In: *Metabolomics*. DOI: 10.1007/s11306-016-1030-9.

Guilbault, G. G. and M. Hjelm (1989). "Nomenclature for automated and mechanised analysis (Recommendations 1989)". In: *Pure and Applied Chemistry* 61.9, pp. 1657–1664. DOI: 10.1351/pac198961091657.

Guitton, Yann et al. (2017). "Create, run, share, publish, and reference your LC–MS, FIA–MS, GC–MS, and NMR data analysis workflows with the Workflow4Metabolomics 3.0 Galaxy online infrastructure for metabolomics". In: *The International Journal of Biochemistry and Cell Biology*. DOI: 10.1016/j.biocel.2017.07.002.

Habchi, Baninia et al. (2017). "An innovative chemometric method for processing direct introduction high resolution mass spectrometry metabolomic data: independent component–discriminant analysis (IC–DA)". In: *Metabolomics*. DOI: 10.1007/s11306-017-1179-x.

Han, Jiawei, Jian Pei, and Yiwen Yin (2000). "Mining Frequent Patterns Without Candidate Generation". In: *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*. SIGMOD '00. Dallas, Texas, USA: ACM, pp. 1–12. DOI: 10.1145/342009.335372.

Hanahan, Douglas and Robert A. Weinberg (2011). "Hallmarks of Cancer: The Next Generation". In: *Cell* 144.5, pp. 646–674. DOI: 10.1016/j.cell.2011.02.013.

Hastings, Janna et al. (2016). "ChEBI in 2016: Improved services and an expanding collection of metabolites". In: *Journal of Nucleic Acids Research* 44.D1, p. 1214. DOI: 10.1093/nar/gkv1031.

Hautbergue, Thaïs et al. (2017). "Evidencing 98 secondary metabolites of Penicillium verrucosum using substrate isotopic labeling and high-resolution mass spectrometry". In: *Journal of Chromatography B* 1071. Identification of molecules from non-targeted analysis, pp. 29–43. DOI: 10.1016/j.jchromb.2017.03.011.

Heng-Keang, Lim et al. (2008). "A generic method to detect electrophilic intermediates using isotopic pattern triggered data-dependent high-resolution accurate mass spectrometry". In: *Rapid Communications in Mass Spectrometry* 22.8, pp. 1295–1311. DOI: 10.1002/rcm.3504.

Hido, S. and H. Kawano (2005). "AMIOT: induced ordered tree mining in tree-structured databases". In: *Fifth IEEE International Conference on Data Mining (ICDM'05)*. DOI: 10.1109/ICDM.2005.20.

Holcapek, M. et al. (2004). "Effects of ion-pairing reagents on the electrospray signal suppression of sulphonated dyes and intermediates". In: *Journal of Mass Spectrometry*. DOI: 10.1002/jms.551.

Hooft, J. J. J. van der, Sandosh Padmanabhan, et al. (2016). "Urinary antihypertensive drug metabolite screening using molecular networking coupled to high-resolution mass spectrometry fragmentation". In: *Metabolomics* 12.7, p. 125. DOI: 10.1007/s11306-016-1064-z.

Hooft, J. J. J. van der, Joe Wandy, Michael P. Barrett, et al. (2016). "Topic modeling for untargeted substructure exploration in metabolomics". In: *PNAS* 113.48, pp. 13738–13743. DOI: 10.1073/pnas.1608041113.

Hooft, J. J. J. van der, Joe Wandy, Francesca Young, et al. (2017). "Unsupervised Discovery and Comparison of Structural Families Across Multiple Samples in Untargeted Metabolomics". In: *Anal. Chem.* 89.14, pp. 7569–7577. DOI: 10.1021/acs.analchem.7b01391.

Horai, H., M. Arita, and T. Nishioka (2008). "Comparison of ESI-MS Spectra in MassBank Database". In: *2008 International Conference on BioMedical Engineering and Informatics*. Vol. 2, pp. 853–857. DOI: 10.1109/BMEI.2008.339.

Horai, Hisayuki et al. (2010). "MassBank: a public repository for sharing mass spectral data for life sciences". In: *Journal of mass spectrometry* 45.7, pp. 703–714. DOI: 10.1002/jms.1777.

Hrydziuszko, O. and MR. Viant (2012). "Missing values in mass spectrometry based metabolomics: an undervalued step in the data processing pipeline". In: *Metabolomics* 8, pp. 161–174. DOI: 10.1007/s11306-011-0366-4.

Huan, J., W. Wang, and J. Prins (2003). "Efficient mining of frequent subgraphs in the presence of isomorphism". In: *Third IEEE International Conference on Data Mining*, pp. 549–552. DOI: 10.1109/ICDM.2003.1250974.

Huan, Jun et al. (2004). "SPIN: Mining Maximal Frequent Subgraphs from Graph Databases". In: *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '04. Seattle, WA, USA: ACM, pp. 581–586. DOI: 10.1145/1014052.1014123.

Inokuchi, Akihiro, Takashi Washio, and Hiroshi Motoda (2000). "An Apriori-Based Algorithm for Mining Frequent Substructures from Graph Data". In: *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery*. PKDD '00. London, UK, UK: Springer-Verlag, pp. 13–23.

Jiang, Chuntao, Frans Coenen, and Michele Zito (2013). "A survey of frequent subgraph mining algorithms". In: *The Knowledge Engineering Review* 28.1, pp. 75–105. DOI: 10.1017/S0269888912000331.

John Draper Amanda J. Lloyd, Royston Goodacre and Manfred Beckmann (2013). "Flow infusion electrospray ionisation mass spectrometry for high throughput, non-targeted metabolite fingerprinting: a review". In: *Metabolomics*. DOI: 10.1007/s11306-012-0449-x.

Johnson, Beverly F., Robert E. Malick, and John G. Dorsey (1992). "Reduction of injection variance in flow-injection analysis". In: *Talanta* 39.1, pp. 35–44. DOI: 10.1016/0039-9140(92)80047-H.

Johnson, Caroline H., Julijana Ivanisevic, and Gary Siuzdak (2016). "Metabolomics: beyond biomarkers and towards mechanisms". In: *Nature Reviews Molecular Cell Biology* 17, pp. 451–459. DOI: 10.1038/nrm.2016.25.

Junot, Christophe et al. (2013). "High resolution mass spectrometry based techniques at the crossroads of metabolic pathways". In: *Mass Spectrometry Reviews* 33.6, pp. 471–500. DOI: 10.1002/mas.21401.

Junttila, Tommi and Petteri Kaski (2011). "Conflict Propagation and Component Recursion for Canonical Labeling". In: *Proceedings of the First International ICST Conference on Theory and Practice of Algorithms in (Computer) Systems*. TAPAS'11. Rome, Italy: Springer-Verlag, pp. 151–162.

Kaddurah-Daouk, Rima, Bruce S. Kristal, and Richard M. Weinshilboum (2008). "Metabolomics: A Global Biochemical Approach to Drug Response and Disease". In: *Annual Review of Pharmacology and Toxicology* 48.1. PMID: 18184107, pp. 653–683. DOI: 10.1146/annurev.pharmtox.48.113006.094715.

Kalman, R. E. (1960). "A New Approach to Linear Filtering and Prediction Problems". In: *Journal of Basic Engineering* 82 (1).

Kanehisa, Minoru et al. (2017). "KEGG: new perspectives on genomes, pathways, diseases and drugs". In: *Journal of Nucleid Acids Research* 45.D1, pp. D353–D361. DOI: 10.1093/nar/gkw1092.

Kasper, Piotr T. et al. (2012). "Fragmentation trees for the structural characterisation of metabolites". In: *Rapid communications in mass spectrometry*. DOI: 10.1002/rcm.6340.

Katajamaa, M. and M. Oresic (2005). "Processing methods for differential analysis of LC/MS profile data". In: *BMC Bioinformatics* 6, p. 179. DOI: 10.1186/1471-2105-6-179.

Kebarle, Paul and Liang Tang (1993). "From ions in solution to ions in the gas phase". In: *Anaytical Chemistry* 65.22, pp. 972–986. DOI: 10.1021/ac00070a001.

Kebarle, Paul and Udo H. Verkerk (2009). "Electrospray : from ions in solution to ions in the gas phase, what we know now." In: *Mass Spectrometry Reviews* 28, pp. 898–917. DOI: 10.1002/mas.20247.

Keller, Bernd O. et al. (2008). "Interferences and contaminants encountered in modern mass spectrometry". In: *Anytica Chimica Acta* 627. DOI: 10.1016/j.aca.2008.04.043.

Kenar, Erhan et al. (2014). "Automated Label-free Quantification of Metabolites from Liquid Chromatography–Mass Spectrometry Data". In: *Molecular and Cellular Proteomics*. DOI: 10.1074/mcp.M113.031278.

Kim, Sunghwan et al. (2016). "PubChem Substance and Compound databases". In: *Journal of Nucleic Acids Research* 44.Database issue, pp. D1202–D1213. DOI: 10.1093/nar/gkv951.

Kind, Tobias and Oliver Fiehn (2006). "Metabolomic database annotations via query of elemental compositions: Mass accuracy is insufficient even at less than 1 ppm". In: *BMC Bioinformatics*. DOI: 10.1186/1471-2105-7-234.

— (2007). "Seven Golden Rules for heuristic filtering of molecular formulas obtained by accurate mass spectrometry". In: *BMC Bioinformatics*. DOI: 10.1186/1471-2105-8-105.

— (2010). "Advances in structure elucidation of small molecules using mass spectrometry". In: *Bioanalytical Reviews*. DOI: 10.1007/s12566-010-0015-9.

King, Richard et al. (2000). "Mechanistic Investigation of Ionization Suppression in Electrospray Ionization". In: *Journal of American Society for Mass Spectrometry*. DOI: 10.1016/S1044-0305(00)00163-X.

Kloet, Frans M. van der et al. (2009). "Analytical Error Reduction Using Single Point Calibration for Accurate and Precise Metabolomic Phenotyping". In: *Journal of Proteome Research* 8.11, pp. 5132–5141. DOI: 10.1021/pr900499r.

Kolev, Spas D. (1995). "Mathematical modelling of flow-injection systems". In: *Analytica Chimica Acta* 308, pp. 36–66. DOI: 10.1016/0003-2670(94)00574-6.

— (2008). "Theoretical Basis of Flow Injection Analysis". In: *Advances in flow injection analysis and related techniques*. Ed. by Spas D. Kolev D. barcelo and Ian D. McKelvie. Elsevier. Chap. Theoretical Basis of Flow Injection Analysis, pp. 47–76.

Kucza, Witold (2013). "Flow injection analysis simulations and diffusion coefficient determination by stochastic and deterministic optimization methods". In: *Analytica Chimica Acta* 788, pp. 74–80. DOI: 10.1016/j.aca.2013.06.006.

Kuhl, Carsten et al. (2012). "CAMERA: An Integrated Strategy for Compound Spectra Extraction and Annotation of Liquid Chromatography/Mass Spectrometry Data Sets". In: *Analytical Chemistry* 84.1, pp. 283–289. DOI: 10.1021/ac202450g.

Kultima, Kim et al. (2009). "Development and Evaluation of Normalization Methods for Label-free Relative Quantification of Endogenous Peptides". In: *Molecular and Cellular Proteomics*. DOI: 10.1074/mcp.M800514-MCP200.

Lange, E. et al. (2006). "High-accuracy peak picking of proteomics data". In: *Computational Proteomics*.

Laponogov, Ivan et al. (2018). "ChemDistiller: an engine for metabolite annotation in mass spectrometry". In: *Bioinformatics* 34.12, pp. 2096–2102. DOI: 10.1093/bioinformatics/bty080.

Lawson, Thomas N. et al. (2017). "msPurity: Automated Evaluation of Precursor Ion Purity for Mass Spectrometry-Based Fragmentation in Metabolomics". In: *Analytical Chemistry* 89.4, pp. 2432–2439. DOI: 10.1021/acs.analchem.6b04358.

Lee, Jinsoo et al. (2012). "An In-depth Comparison of Subgraph Isomorphism Algorithms in Graph Databases". In: *Proceedings of the VLDB Endowment* 6.2, pp. 133–144. DOI: 10.14778/2535568.2448946.

Lee, MinJae (2010). "Multiple Inputation and Quantile Regression methods for Biomarker Data subject to Detection Limits". PhD thesis. University of Pittsburgh. DOI: 10.1186/s12874-017-0463-9.

Lei, Zhentian, David V. Huhman, and Lloyd W. Sumner (2011). "Mass Spectrometry Strategies in Metabolomics". In: *Journal of biological chemistry*. DOI: 10.1074/jbc.R111.238691.

Levenspiel, Octave and Kenneth B. Bischoff (1964). "Patterns of flow in chemical process vessels". In: *Advances in chemical engineering (Vol 4.)* Ed. by Thomas B. Drew. Wiley. Chap. Patterns of flow in chemical process vessels, pp. 95–198. DOI: 10.1016/S0065-2377(08)60240-9.

Li, Zhucui et al. (2018). "Comprehensive evaluation of untargeted metabolomics data processing software in feature detection, quantification and discriminating marker selection". In: *Analytica Chimica Acta* 1029, pp. 50–57. DOI: 10.1016/j.aca.2018.05.001.

Libiseller, Gunnar et al. (2015). "IPO: a tool for automated optimization of XCMS parameters". In: *BMC bioinformatics*. DOI: 10.1186/s12859-015-0562-8..

Livera, Alysha M. De et al. (2015). "Statistical Methods for Handling Unwanted Variation in Metabolomics Data". In: *Anal. Chem.* 87.7, pp. 3606–3615. DOI: 10.1021/ac502439y.

Lloyd, Amanda J et al. (2011). "Use of mass spectrometry fingerprinting to identify urinary metabolites after consumption of specific foods". In: *American Journal of Clinical Nutrition*. DOI: 10.3945/ajcn.111.017921.

Madalinski, G. et al. (2008). "Direct introduction of biological samples into a LTQ-orbitrap hybrid mass spectrometer as a tool for fast metabolome analysis". In: *Anal. Chem.* 80.9, pp. 3291–3303. DOI: 10.1021/ac7024915.

Makarov, Alexander (2005). "Electrostatic Axially Harmonic Orbital Trapping: A High-Performance Technique of Mass Analysis". In: *Analytica Chemistry*. DOI: 10.1021/ac991131p.

Margaret, Hartnett and Diamond Dermot (1993). "Neural Network Based Recognition of Flow Injection Patterns". In: *Analyst* 118.118, pp. 347–354. DOI: 10.1039/AN9931800347.

Mark S., Jeansonne and Foley Joe P. (1992). "Improved equations for the calculation of chromatographic figures of merit for ideal and skewed chromatographic peaks". In: *Journal of Chromatography* 594, pp. 1–8. DOI: 10.1016/0021-9673(92)80307-G.

Martin, A. J. P. and R. L. M. Synge (1941). "A new form of chromatogram employing two liquid phases". In: *Chromtography of amino-acids*.

Mathur, Raman and Peter B. O'Connor (2009). "Artifacts in Fourier transform mass spectrometry". In: *Rapid Communications in Mass Spectrometry*. DOI: 10.1002/rcm.3904.

McKay, Brendan D et al. (1981). "Practical graph isomorphism". In:

Mckay, Brendan D. and Adolfo Piperno (2014). "Practical Graph Isomorphism, II". In: *J. Symb. Comput.* 60, pp. 94–112. DOI: 10.1016/j.jsc.2013.09.003.

Melamud, Eugene, Livia Vastag, and Joshua D. Rabinowitz (2010). "Metabolomic Analysis and Visualization Engine for LC–MS Data". In: *Analytical Chemistry*. DOI: 10.1021/ac1021166.

Meyer, David, Friedrich Leisch, and Kurt Hornik (2003). "The support vector machine under test". In: *Neurocomputing* 55.1. Support Vector Machines, pp. 169–186. DOI: doi.org/10.1016/S0925-2312(03)00431-4.

Monteiro, M.S. et al. (2013). "Metabolomics Analysis for Biomarker Discovery: Advances and Challenges". In: *Current Medicinal Chemistry* 20.2, pp. 257–271. DOI: 10.2174/0929867311320020006.

Moré, Jorge J. (1978). "The Levenberg-Levenberg implementation and theory". In: *Numerical analysis*.

Mrzic, Aida et al. (2017). "Automated Recommendation Of Metabolite Substructures From Mass Spectra Using Frequent Pattern Mining". In: *bioRxiv*. DOI: 10.1101/134189.

Murray, Kermit K. et al. (2013). "Definitions of terms relating to mass spectrometry (IUPAC Recommendations 2013)". In: *Pure and applied chemistry*. DOI: 10.1351/PAC-REC-06-04-06.

Mylonas, Roman et al. (2009). "X-Rank: A Robust Algorithm for Small Molecule Identification Using Tandem Mass Spectrometry". In: *Analytical chemistry* 81.18, pp. 7604–7610. DOI: 10.1021/ac900954d.

Naake, Thomas and Emmanuel Gaquerel (2017). "MetCirc: navigating mass spectral similarity in high-resolution MS/MS metabolomics data". In: *Bioinformatics* 33.15, pp. 2419–2420. DOI: 10.1093/bioinformatics/btx159.

Nagana Gowda, GA and D Raftery (2013). "Biomarker Discovery and Translation in Metabolomics". In: *Current Metabolomics* 1.3, pp. 227–240. DOI: 10.2174/2213235X113019990005.

Nanita, Sergio C. (2013). "Quantitative Mass Spectrometry Independence from Matrix Effects and Detector Saturation Achieved by Flow Injection Analysis with Real-Time Infinite Dilution". In: *Anaytical Chemistry* 85, pp. 11866–11875. DOI: 10.1021/ac402567w.

Newgard, Christopher B. (2017). "Metabolomics and Metabolic Diseases: Where Do We Stand?" In: *Cell Metabolism* 25.1, pp. 43–56. DOI: 10.1016/j.cmet.2016.09.018.

Newgard, Christopher B. et al. (2009). "A Branched-Chain Amino Acid-Related Metabolic Signature that Differentiates Obese and Lean Humans and Contributes to Insulin Resistance". In: *Cell Metabolism* 9.4, pp. 311–326. DOI: 10.1016/j.cmet.2009.02.002.

Nijssen, Siegfried and Joost N. Kok (2004). "A Quickstart in Frequent Structure Mining Can Make a Difference". In: *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '04. Seattle, WA, USA: ACM, pp. 647–652. DOI: 10.1145/1014052.1014134.

Nikolskiy, Igor et al. (2013). "An Untargeted Metabolomic Workflow to Improve Structural Characterization of Metabolites". In: *Analytical Chemistry* 85.16, pp. 7713–7719. DOI: 10.1021/ac400751j.

Oliver, Stephen G. et al. (1998). "Systematic functional analysis of the yeast genome". In: *Trends in Biotechnology* 16.9, pp. 373–378. DOI: 10.1016/S0167-7799(98)01214-1.

Page, Jason S. et al. (2007). "Ionization and Transmission Efficiency in an Electrospray Ionization–Mass Spectrometry Interface". In: *American Society for Mass Spectrometry*. DOI: 10.1016/j.jasms.2007.05.018.

Pence, Harry E. and Antony Williams (2010). "ChemSpider: An Online Chemical Information Resource". In: *Journal of Chemical Education* 87.11, pp. 1123–1124. DOI: 10.1021/ed100697w.

Peng, J. y. et al. (2008). "An Efficient Algorithm for Detecting Closed Frequent Subgraphs in Biological Networks". In: *2008 International Conference on BioMedical Engineering and Informatics*. Vol. 1, pp. 677–681. DOI: 10.1109/BMEI.2008.187.

Pesyna, Gail M. et al. (1976). "Probability based matching system using a large collection of reference mass spectra". In: *Analytical chemistry* 48.9, pp. 1362–1368. DOI: 10.1021/ac50003a026.

Pinto, Rui Climaco, Johan Trygg, and Johan Gottfries (2013). "Advantages of orthogonal inspection in chemometrics". In: *Journal of Chemometrics* 26.6, pp. 231–235. DOI: 10.1002/cem.2441.

Pluskal, Tomas et al. (2010). "MZmine 2: Modular framework for processing, visualizing, and analyzing mass spectrometry- based molecular profile data". In: *BMC Bioinformatics*. DOI: 10.1186/1471-2105-11-395.

Prosser, Gareth A, Gerald Larrouy-Maumus, and Luiz Pedro S de Carvalho (2014). "Metabolomic strategies for the identification of new enzyme functions and metabolic pathways". In: *EMBO Reports* 15.6, pp. 657–669. DOI: 10.15252/embr.201338283.

Rasche, Florian et al. (2010). "Computing Fragmentation Trees from Tandem Mass Spectrometry Data". In: *analytical chemistry*. DOI: 10.1021/ac101825k.

Rauf, Imran et al. (2012). *Finding Maximum Colorful Subtrees in Practice*. In: *Research in computational molecular biology*. DOI: 10.1089/cmb.2012.0083.

Robotti, Elisa, Marcello Manfredi, and Emilio Marengo (2014). "Biomarkers Discovery through Multivariate Statistical Methods: A Review of Recently Developed Methods and Applications in Proteomics". In: *Journal of Proteomics and Bioinformatics*. DOI: 10.4172/jpb.S3-003.

Rolin, D (2013). *Metabolomics Coming of Age with its Technological Diversity*. Advances in Botanical Research. Vol. 67. 693 pp.

Roux, Aurelie et al. (2012). "Annotation of the Human Adult Urinary Metabolome and Metabolite Identification Using Ultra High Performance Liquid Chromatography Coupled to a Linear Quadrupole Ion Trap-Orbitrap Mass Spectrometer". In: *Analytical Chemistry* 84.15, pp. 6429–6437. DOI: 10.1021/ac300829f.

Rusilowicz, Martin et al. (2015). "A batch correction method for liquid chromatography-mass spectrometry data that does not depend on quality control samples". In: *Metabolomics* 12.3, pp. 56–67. DOI: 10.1007/s11306-016-0972-2.

Ruttkies, Christoph et al. (2016). "MetFrag relaunched: incorporating strategies beyond in silico fragmentation". In: *Journal of chemoinformatics* 8.1, pp. 3–. DOI: 10.1186/s13321-016-0115-9.

Ruzicka, Jaromir and Elo Harald Hansen (1988). *Flow Injection Analysis, 2nd Edition*. Ed. by J. D. Winerfordner. Wiley.

Savtisky, Abraham and Marcel J. E. Golay (1964). "Smoothing and Differentiation of Data by Simplified Least Squares Procedures". In: *Analytical Chemistry*. DOI: 10.1021/ac60214a047.

Scalbert, Augustin et al. (2014). "The food metabolome: a window over dietary exposure". In: *The American Journal of Clinical Nutrition* 99.6, pp. 1286–1308. DOI: 10.3945/ajcn.113.076133.

Scheubert, Kerstin et al. (2017). "Significance estimation for large scale metabolomics annotations by spectral matching". In: *Nature Communications* 8.1, pp. 1494–. DOI: 10.1038/s41467-017-01318-5.

Schymanski, Emma L. and Steffen Neumann (2013). "The Critical Assessment of Small Molecule Identification (CASMI): Challenges and Solutions". In: *Metabolites* 3, pp. 517–538. DOI: 10.3390/metabo3030517.

Schymanski, Emma L., Christoph Ruttkies, et al. (2017). "Critical Assessment of Small Molecule Identification 2016: automated methods". In: *Journal of Chemoinformatics* 9.1, p. 22. DOI: 10.1186/s13321-017-0207-1.

Scott, Ian M. et al. (2010). "Enhancement of Plant Metabolite Fingerprinting by Machine Learning". In: *Bioinformatics*. DOI: 10.1104/pp.109.150524.

Sévin, Daniel C. et al. (2016). "Nontargeted in vitro metabolomics for high-throughput identification of novel enzymes in Escherichia coli". In: *Nature Methods*. DOI: 10.1038/nmeth.4103.

Shah, Jasmit S. et al. (2017). "Distribution based nearest neighbor imputation for truncated high dimensional data with applications to pre-clinical and clinical metabolomics studies". In: *BMC Bioinformatics*. DOI: 10.1186/s12859-017-1547-6.

Shamir, Ron and Dekel Tsur (1999). "Faster Subtree Isomorphism". In: *Journal of the ACM* 33.2, pp. 267–280. DOI: 10.1006/jagm.1999.1044.

Shang, Haichuan et al. (2008). "Taming Verification Hardness: An Efficient Algorithm for Testing Subgraph Isomorphism". In: *Proc. VLDB Endow.* 1.1, pp. 364–375. DOI: 10.14778/1453856.1453899.

Shen, Huibin, Kai Dührkop Sebastian Böcker, and Juho Rousu (2014). "Metabolite identification through multiple kernel learning on fragmentation trees". In: *Bioinformatics*. DOI: 10.1093/bioinformatics/btu275.

Smedsgaard, Jorn and Jens C. Frisvad (1995). "Using direct electrospray mass spectrometry in taxonomy and secondary metabolite profiling of crude fungal extracts". In: *Journal of Microbiological Methods*.

Smit, H.C. and P.J.H. Scheeren (1988). "Characterization of responses to injection of microliter samples and to peak-shaped input signals for inductively-coupled plasma/atomic emission spectrometry". In: *Analytica Chimica Acta* 215, pp. 143–153.

Smith, C.A. et al. (2006). "XCMS: processing mass spectrometry data for metabolite profiling using nonlinear peak alignment, matching, and identification". In: *Analytical Chemistry* 78.3, pp. 779–787. DOI: 10.1021/ac051437y.

Smith, Rob et al. (2014). "Proteomics, lipidomics, metabolomics: a mass spectrometry tutorial from a computer scientist's point of view". In: *BMC Bioinformatics*.

Sokolovv, Steve, Joel Karnofsky, and Phil Gustafson (1978). *The Finnigan Library Search Program*. Application Report. Finnigan Instruments.

Southam, Andrew D. et al. (2007). "Dynamic Range and Mass Accuracy of Wide-Scan Direct Infusion Nanoelectrospray Fourier Transform Ion Cyclotron Resonance Mass Spectrometry-Based Metabolomics Increased by the Spectral Stitching Method". In: *Analytical Chemistry*. DOI: 10.1021/ac062446p.

Southam, Andrew D et al. (2017). "A complete workflow for high-resolution spectral-stitching nanoelectrospray direct-infusion mass-spectrometry-based metabolomics and lipidomics". In: *Nature Protocol*. DOI: 10.1038/nprot.2016.156.

Stein, Stephen E. and Donald R. Scott (1994). "Optimization and Testing of Mass Spectral Library Search Algorithms for Compound Identification". In: *Journal of American Society for Mass Spectrometry*. DOI: 10.1016/1044-0305(94)87009-8.

Stekhoven, DJ. and P. Bühlmann (2011). "MissForest–non-parametric missing value imputation for mixed-type data." In: *Bioinformatics*. DOI: 10.1093/bioinformatics/btr597.

Stravs, Michael A. et al. (2012). "Automatic recalibration and processing of tandem mass spectra using formula annotation". In: *Journal of Mass Spectrometry* 48.1, pp. 89–99. DOI: 10.1002/jms.3131.

Sumner, Lloyd W et al. (2007). "Proposed minimum reporting standards for chemical analysis Chemical Analysis Working Group (CAWG) Metabolomics Standards Initiative (MSI)". In: *Metabolomics* 3.3, pp. 211–221. DOI: 10.1007/s11306-007-0082-2.

Tautenhahn, R., C. Bottcher, and S. Neumann (2008). "Highly sensitive feature detection for high resolution LC/MS". In: *BMC Bioinformatics* 9.1, p. 504. DOI: 10.1186/1471-2105-9-504.

Tengstrand, Erik, Johan Lindberg, and K. Magnus Abergberg (2014). "TracMass 2–Modular Suite of Tools for Processing Chromatography-Full Scan Mass Spectrometry Data". In: *Analytical Chemistre*. DOI: 10.1021/ac403905h.

Termier, Alexandre et al. (2007). "DIGDAG, a First Algorithm to Mine Closed Frequent Embedded Sub-DAGs". In: *MLG*.

Thomas, L. T., S. R. Valluri, and K. Karlapalem (2006). "MARGIN: Maximal Frequent Subgraph Mining". In: *Sixth International Conference on Data Mining (ICDM'06)*, pp. 1097–1101. DOI: 10.1109/ICDM.2006.102.

Titzmann, Thorsten et al. (2010). "Improved peak analysis of signals based on counting systems: Illustrated for proton-transfer-reaction time-of-flight mass spectrometry". In: *International Journal of Mass Spectrometry*. DOI: 10.1016/j.ijms.2010.07.009.

Treutler, Hendrik and Steffen Neumann (2016). "Prediction, Detection, and Validation of Isotope Clusters in Mass Spectrometry Data". In: *Metabolites*. DOI: 10.3390/metabo6040037.

Treviño, Victor et al. (2014). "GridMass: a fast two-dimensional feature detection method for LC/MS". In: *Journal of Mass-Spectrometry*. DOI: 10.1002/jms.3512.

Trivedi, Drupad K., Katherine A. Hollywood, and Royston Goodacre (2017). "Metabolomics for the masses: The future of metabolomics in a personalized world". In: *New Horizons in Translational Medicine* 3.6, pp. 294–305. DOI: 10.1016/j.nhtm.2017.06.001.

Trotzmuller, Martin et al. (2010). "Characteristics and origins of common chemical noise ions in negative ESI LC-MS". In: *Journal of Mass Spectrometry*. DOI: 10.1002/jms.1924.

Trufelli, Helga et al. (2011). "An overview of matrix effects in Liquid Chromatography-Mass Spectrometry". In: *Mass Spectrometry Reviews*. DOI: 10.1002/mas.20298.

Trygg, Johan and Svante Wold (2002). "Orthogonal projections to latent structures (O-PLS)". In: *Journal of Chemometrics* 16.3, pp. 119–128. DOI: 10.1002/cem.695.

Tsugawa, Hiroshi, Tomas Cajka, et al. (2015). "MS-DIAL: data-independent ms/ms deconvolution for comprehensive metabolome analysis". In: *Nature Methods*. DOI: 10.1038/nmeth.3393.

Tsugawa, Hiroshi, Mitsuhiro Kanazawa, et al. (2014). "MRMPROBS suite for metabolomics using large-scale MRM assays". In: *Bioinformatics*. DOI: 10.1093/bioinformatics/btu203.

Tsugawa, Hiroshi, Tobias Kind, et al. (2016). "Hydrogen rearrangement rules: computational MS/MS fragmentation and structure elucidation using MS-FINDER software". In: *Analytical chemistry* 88.16, pp. 7946–7958. DOI: 10.1021/acs.analchem.6b00770.

Ullmann, J. R. (1976). "An Algorithm for Subgraph Isomorphism". In: *J. ACM* 23.1, pp. 31–42. DOI: 10.1145/321921.321925.

Viant, Mark R et al. (2017). "How close are we to complete annotation of metabolomes?" In: *Current Opinion in Chemical Biology* 36. Omics, pp. 64–69. DOI: 10.1016/j.cbpa.2017.01.001.

Vinaixa, Maria et al. (2016). "Mass spectral databases for LC/MS- and GC/MS-based metabolomics: State of the field and future prospects". In: *Angewandte Communications* 78, pp. 23–35. DOI: 10.1016/j.trac.2015.09.005.

Wahab, M. Farooq, Darshan C. Patel, and Daniel W. Armstrong (2017). "Total peak shape analysis: detection and quantitation of concurrent fronting, tailing, and their effect on asymmetry measurements". In: *Journal of Chromatography A* 1509, pp. 163–170. DOI: 10.1016/j.chroma.2017.06.031.

Wang, Mingxun et al. (2016). "Sharing and community curation of mass spectrometry data with Global Natural Products Social Molecular Networking". In: *Nature Biotechnology* 34, pp. 828–. DOI: 10.1038/nbt.3597.

Wang, San-Yuan, Ching-Hua Kuo, and Yufeng J. Tseng (2013). "Batch Normalizer: A Fast Total Abundance Regression Calibration Method to Simultaneously Adjust Batch and Injection Order Effects in Liquid Chromatography/Time-of-Flight Mass Spectrometry-Based Metabolomics Data and Comparison with Current Calibration Methods". In: *Analytical Chemistry* 85.2, pp. 1037–1046. DOI: 10.1021/ac302877x.

Wang, Zeneng et al. (2011). "Gut flora metabolism of phosphatidylcholine promotes cardiovascular disease". In: *Nature* 472, pp. 57–63. DOI: 10.1038/nature09922.

Ward, Jane L. et al. (2010). "The metabolic transition during disease following infection of Arabidopsis thaliana by Pseudomonas syringae pv. tomato". In: *The Plant Journal*. DOI: 10.1111/j.1365-313X.2010.04254.x.

Wee, Andrew et al. (2008). "A continuous wavelet transform algorithm for peak detection". In: *Electrophoresis*. DOI: 10.1002/elps.200800096.

Wei, Runmin et al. (2018). "Missing Value Imputation Approach for Mass Spectrometry-based Metabolomics Data". In: *Scientific Reports* 8.1. DOI: 10.1038/s41598-017-19120-0.

Wentzell, Peter D., Michael R. Bowdridge, et al. (1992). "Random walk simulation of flow injection analysis. Evaluation of dispersion profiles". In: *Analytica Chimica Acta*. DOI: 10.1016/0003-2670(93)85113-X.

Wentzell, Peter D. and Anthony C. Tarasuk (2014). "Characterization of heteroscedastic measurement noise in the absence of replicates". In: *Analytica Chimica Acta* 847, pp. 16–28. DOI: 10.1016/j.aca.2014.08.007.

Wishart, David S. (2016). "Emerging applications of metabolomics in drug discovery and precision medicine". In: *Nature Reviews Drug Discovery* 15, pp. 473–484. DOI: 10.1038/nrd.2016.32.

Wishart, David S et al. (2018). "HMDB 4.0: the human metabolome database for 2018". In: *Nucleic Acids Research* 46.D1, pp. 608–617. DOI: 10.1093/nar/gkx1089.

Wold, Herman (1966). "Estimation of Principal Components and Related Models by Iterative Least squares". In: *Multivariate Analysis*. New York: Academic Press, pp. 391–420.

Wold, Svante, Michael Sjöström, and Lennart Eriksson (2001). "PLS-regression: a basic tool of chemometrics". In: *Chemometrics and Intelligent Laboratory Systems* 58.2. PLS Methods, pp. 109–130. DOI: 10.1016/S0169-7439(01)00155-1.

Wolf, Sebastian et al. (2010). "In silico fragmentation for computer assisted identification of metabolite mass spectra". In: *BMC Bioinformatics* 11.1, p. 148. DOI: 10.1186/1471-2105-11-148.

Wörlein, Marc et al. (2005). "A Quantitative Comparison of the Subgraph Miners Mofa, Gspan, FFSM, and Gaston". In: *Proceedings of the 9th European Conference on Principles and Practice of Knowledge Discovery in Databases*. PKDD'05. Porto, Portugal: Springer-Verlag, pp. 392–403.

Worley, Bradley and Robert Powers (2013). "Multivariate Analysis in Metabolomics". In: *Current Metabolomics* 1.1, pp. 92–107. DOI: 10.2174/2213235X11301010092.

Yan, Xifeng and Jiawei Han (2002). "gSpan: graph-based substructure pattern mining". In: *2002 IEEE International Conference on Data Mining, 2002. Proceedings.* Pp. 721–724. DOI: 10.1109/ICDM.2002.1184038.

— (2003). "CloseGraph: Mining Closed Frequent Graph Patterns". In: *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '03. Washington, D.C.: ACM, pp. 286–295. DOI: 10.1145/956750.956784.

Yan, Xifeng, X. Jasmine Zhou, and Jiawei Han (2005). "Mining Closed Relational Graphs with Connectivity Constraints". In: *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*. KDD '05. Chicago, Illinois, USA: ACM, pp. 324–333. DOI: 10.1145/1081870.1081908.

Yang, Li et al. (2009). "Screening phosphatidylcholine biomarkers in mouse liver extracts from a hypercholesterolemia study using ESI-MS and chemometrics". In: *Analytical and Bioanalytical Chemistry* 393 (2).

Yang, Ming, Tomoyoshi Soga, and Patrick J Pollard (2013). "Oncometabolites: linking altered metabolism with cancer". In: *The Journal of Clinical Investigation* 123.9, pp. 3652–3658. DOI: 10.1172/JCI67228.

Yu, Tianwei and Dean P. Jones (2014). "Improving peak detection in high-resolution LC/MS metabolomics data using preexisting knowledge and machine learning approach". In: *Bioinformatics* 30 (24), pp. 2941–2948. DOI: 10.1093/bioinformatics/btu430.

Yu, Tianwei, Youngja Park, et al. (2009). "apLCMS-adaptive processing of high-resolution LC/MS data". In: *Bioinformatics*. DOI: 10.1093/bioinformatics/btp291.

Yu, Tianwei and Hesen Peng (2010). "Quantification and deconvolution of asymmetric LC-MS peaks using the bi-Gaussian mixture model and statistical model selection". In: *BMC Bioinformatics* 11.1, p. 559. DOI: 10.1186/1471-2105-11-559.

Zaki, M. J. (2005). "Efficiently mining frequent trees in a forest: algorithms and applications". In: *IEEE Transactions on Knowledge and Data Engineering* 17.8, pp. 1021–1035. DOI: 10.1109/TKDE.2005.125.

Zaki, Mohammed J. (2002). "Efficiently Mining Frequent Trees in a Forest". In: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data*

Mining. KDD '02. Edmonton, Alberta, Canada: ACM, pp. 71–80. DOI: 10.1145/775047.775058.

Zaki, Mohammed J. (2004). "Efficiently Mining Frequent Embedded Unordered Trees". In: *Fundamental Informatica* 66.1-2, pp. 33–52.

Zaki, Mohammed J. and Ching-Jui Hsiao (2002). "CHARM: An Efficient Algorithm for Closed Itemset Mining". In: *Proceedings of the 2002 SIAM International Conference on Data Mining*, pp. 457–473. DOI: 10.1137/1.9781611972726.27.

Zamboni, Nicola, Alan Saghatelian, and GaryÂ J. Patti (2015). "Defining the Metabolome: Size, Flux, and Regulation". In: *Molecular Cell* 58.4, pp. 699–706. DOI: 10.1016/j.molcel.2015.04.021.

Zhang, Aihua et al. (2012). "Modern analytical techniques in metabolomics analysis". In: *Analyst* 137 (2), pp. 293–300. DOI: 10.1039/C1AN15605E.

Zhang, Guo-Fang et al. (2011). "Metabolomics, Pathway Regulation, and Pathway Discovery". In: *Journal of Biological Chemistry* 286 (27), pp. 23631–23635. DOI: 10.1074/jbc.R110.171405.

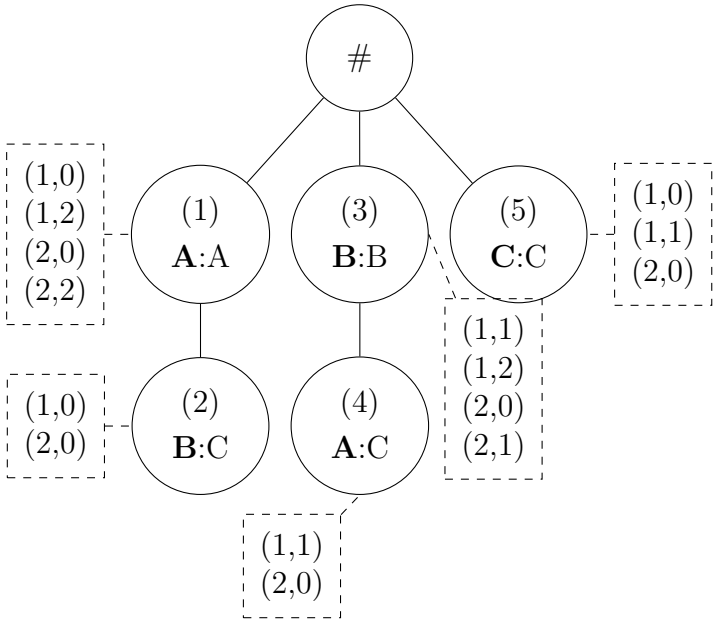
Zhang, Wenchao et al. (2014). "MET-COFEA: A Liquid Chromatography/Mass Spectrometry Data Processing Platform for Metabolite Compound Feature Extraction and Annotation". In: *Analytical Chemistry* 86.13. PMID: 24856452, pp. 6245–6253. DOI: 10.1021/ac501162k.

Appendices

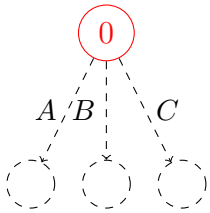
Appendix A

k-LDFM subtree enumeration example

Here we describe the generation of all subtrees from the k-Path tree from Figure 11.2 using Algorithm 7.



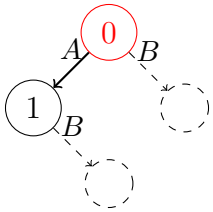
This tree is similar to the k-Path Treefrom Figure 11.2c. A unique identifier has been added in parenthesis for each node v. The bold letter indicates the label of the corresponding edge $l(v)$, and the second label is $h(v)$. $O(v)$ are shown in dashed boxes. For clarity, the growth of each pattern starts from the root (in contrast to Algorithm 7), and the computation of the support is also detailed.



$Exts = ((0, A, 1),$
$(0, B, 3), (0, C, 5))$
$S = \emptyset$

Initial state (1)

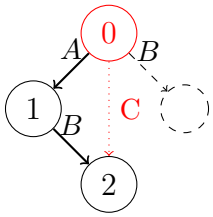
Initially, only the root is present on the graph and all extensions are possible. Here, the extensions are showed in dashed lines for visualization purpose, and the root node is in red.



$Exts = ((1, B, 2), (0, B, 3))$
$S = \{A\}$
$O = \{(1, 0), (1, 2),$
$(2, 0), (2, 2)\}$

Extension $(0, A, 1)$ **(2)**

Step 2: After the first extension $(0, A, 1)$ is added, the extension $(1, B, 2)$ is added by looking at the possible labels of the successors in T. Extension $(0, C, 5)$ becomes impossible because it would lead to the same node as extension $(1, B, 2)$, as readily detected by checking that $h_T(2) = C$. The occurrences are directly extracted from T.



$Exts = ((0, B, X))$
$S = \{A, C\}$
$O = \{(1, 0), (2, 0)\}$

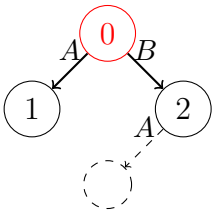
Extension $(1, B, 2)$ **(3)**

The addition of $(1, B, 2)$ does not result in a new extension as there is no extension from (2) in T. A C edge is is directly added from the root of the pattern because $h_T(2) = C$. The new occurrence set in computed by intersecting the occurrences of the parent patterns and the occurrences of $O_T(2)$.

$Exts = ()$
$S = \{A, C\}$

Test of $(0, B, 3)$ **and backtrack (4)**

The support of the graph obtained by extending **(3)** by $(0, B, 3)$ is $\{(1, 0), (2, 0)\} \cap \{(1, 1), (1, 2), (2, 0), (2, 1)\} = \{(2, 0)\}$, therefore it is not frequent and the algorithm backtracks to **(2)**.



$Exts = ((2, A, 4))$

$S = \{A, B, C\}$

Extension $(0, B, 3)$ **(5)**

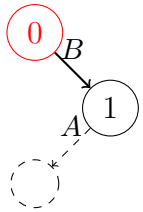
Because we add an extension which is superior in Left-Most Depth-First Order, the extension $(1, A, 1)$ is removed. $h_T(1)$ is also added to S. Extension $(2, A, 4)$ is added from the path tree. The intersection of the occurrence sets is performed.

$Exts = ()$

$S = \{A, B, C\}$

Test of $(2, A, 4)$ **and backtrack (6)**

Because $h_T(4) = C$ and C is in S, the algorithm backtracks to **(1)**, as there is no more extension in **(2)**.

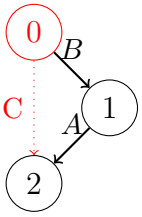


$Exts = ((1, A, 4))$

$S = \{A, B\}$

Extension $(0, B, 3)$ **(7)**

Because we add an extension which is superior to $(0, A, 1)$ in the Left-Most Depth First order, the extension $(0, A, 1)$ is removed as $h_T(3) = A$, A is added to S. Extension $(1, A, 4)$ is added from T.



$Exts = ()$

$S = \{A, B, C\}$

Extension $(1, A, 4)$ **(8)**

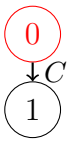
Because $h_T(4) = C$, C is added S. There is no new extension to be added from T.

$Exts = ()$

$S = \{A, B, C\}$

Backtrack to (1)

As there is no more extension in **(7) and (8)**, the algorithm backtracks to **(1)**.



$Exts = ()$

$S = \{A, B, C\}$

Extension $(0, C, 5)$**(1)**

As the extension $(0, C, 5)$ is added, $(0, A, 1)$ and $(0, B, 3)$ extensions are removed because they are lower in the Depth-First Left-Most order. As a result, A and B are added to the set of forbidden values.

After this step, the algorithm ends, as all the possible spanning subtrees have been visited.

Synthèse

Dans cette thése sont décrites deux contributions distinctes que nous avons développées et mises en œuvre pour le traitement à haut débit et l'annotation des données métabolomiques de la spectrométrie de masse à haute résolution. Un état de l'art global introduisant la métabolomique et justifiant les travaux effectués dans cette thèse est dréssé dans la section 2. Cette thése est donc divisèe and deux parties inépendantes.

Dans la partie II, nous avons proposé le premier workflow librement disponible pour le traitement des données haut débit d'Analyses par injection en Flux Continues (FIA), une technique de métabolomique haut-deébit. Cet algorithme est basé sur une étude de l'état de l'art de la détection de pics en métabolomiques effectuées en section 4. Pour cela nous avons proposé un nouveau modéle basé sur les processus physiques affectant l'intensité mesuré pour chaque ion. Trois composants ainsi que des moéles calculables pour chacun d'entre eux ont éte extraits dans le chapitre 5. Les composants les plus important sont les suivants : un pic résultant du gradient de concentration induit par le système de FIA noté P, l'effet matrice qui est exprimé en fonction du pic d'injection en vert. La somme des ces deux composants est le quantité d'ion qui passe dans le spectromètre de masse. Néanmoins de par la mesure de cette intensité par le spectrométre de masses, un bruit hétéroscedastique s'y ajoute. Ce modéle est étudi'e plus en détail dans le chapitre 6. Suite á des experiences sur des données simulées on constate notamment que les paramètres du modéles sont trop difficiles á estimer quand le signal est trop affecté par l'effet matrice. Un workflow tenant en compte cette spécificité est a donc été développé et est décrit dans la figure. Dans une premiére parties des bandes de points de masse proches et consecutifs sont détectés. Certaines bandes, suffisamment intenses et pécsentant un pic bien visible sont ensuite utilisées pour estimer P. Cette estimation est ensuite utilisée pour filtrer les signaux et estimer les limites de l'injection pour chaque ion. Cette méthode permet notamment d'obtenir des indicateurs de qualités des signaux interprétables chimiquement. Cette méthode est à notre connaissance la premiére à permettre d'extraire des métriques mesurant l'effet mátrice. Cet algorithme de détection de pics ainsi que des algorithmes d'alignement des signaux et d'imputation des données manquantes permettant un prétraitement complet des données de FIA-HRMS. Ils ont été packagé dans un paquet R, textbfproFIA.

Ce workflow a été évalué sur des données réelles dans le chapitre 7. Pour cela proFIA a été comparé à la détection de pics manuelles effectuée par un expert, ainsi qu'à XCMS(C. Smith et al. 2006) une méthode de référence. Il a été démontré sur

plusieurs appareils que proFIA était plus sensible et plus reproductible dans les deux cas. De plus proFIA avait des performances très proche de la détection manuelle par un expert qui peut être considéré le gold standard. Ces résultats illustrent l'intérêt de la méthodologie proposée. Les perspectives de ce travail sonc discutés dans la partie 8, on peut notamment noter l'utilisation des métriques d'effet matrices pour mieux comprendre les classes de molécules affectées.

La deuxième partie du doctorat a porté sur à l'annotation structurale des métabolites des spectres MS/MS, qui constitue un enjeu majeur en métabolomique. Un état de l'art de cette problèmatique est donné dans la section 9.4. Nous proposons une méthode d'extraction de motifs de fragmentation d'une collection de spectres basées sur une modélisation des spectres sous formes de graphes. Pour cela dans le chapitre 10 nous proposons une nouvelle représentation des spectres de fragmentation sous forme de graphe qui ne nécessite pas de connaitres la composition moléculaire de l'ion parent pour être construite. Nous détaillons le processus de construction de ces graphes et leurs propriétés tout au long du chapitre 4. L'une des propriétés les plus importantes données dans ce chapitre le fait qu'il est possible d'obtenir un algorithme de complexité linéaire pour résoudre le problème d'isomoprhisme de graphes sur ces graphs particuliers.

Dans le chapitre 11, un algorithme de Frequent Subgraph Mining adaptés à ces graphes est proposé. Cet algorithme repose sur une constatation tirés d'un état de l'art sur les algorithmes de FSM dressés dans la section 9.6, les problèmes de FSM étant très demandeur en calcul, il faut réduire l'espace des sous graphe recherchés autant que possible. Pour cela dans le chapitre 11 on montre qu'il est possible de se limiter à la génération de sous arbres fréquents plutôt que de sous graphes. L'exhaustivité de l'algorithme de génération des sous graphes fréquent est prouvée. Cet algorithme a été implémenté dans un package R/C++ mineMS2, et il a été testé sur deux jeux de données biologiques. Sur un jeu de donées de plus de 600 spectres l'agorithme tourne en moins de 5 minutes, contrairement aux algorithme de FSM plus généraux qui ne terminent pas en 2 heures. Ceci est dû à la topologie spécifique des graphes minés qu'ils ne prennent pas en compte. Les limites et les perspectives de cette méthodologie sont discutées dans la section 12, notamment le grand nombre de motifs de fragmentations extraits.

Cette thèse présentedeux contributions originales au traitement de données metabolomiques, toutes deux diponibles à la communauté sous forme de package R (mineMS2). Les deux méthodes proposées sont de plus toute deux dérivées du processes physique, permettant une plus grand interprétabilité.

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre : Nouvelles approches pour le traitement et l'annotation des données de métabolomique haut débit obtenues par spectrométrie de masse haute-résolution

Mots-clés : Analyse de Graphes, Metabolomique, Détection de sous graphes fréquents, Traitement du signal

La métabolomique est une approche de phénotypage présentant des perspectives prometteuses pour le diagnostic et le suivi de plusieurs pathologies. La technique d'observation la plus utilisée en métabolomique est la spectrométrie de masse (MS). Des dévloppements technologiques récents ont considérablement accru la taille et la complexité des données. Cette thèse s'est concentrée sur deux verrous du traitement de ces données, l'extraction de pics des données brutes et l'annotation des spectres.
La première partie de la thèse a portée sur le développement d'un nouvel algorithme de détection de pics pour des données d'analyse par injection en flot continue (Flow Injection Analysis ou FIA), une technique ahut-débit. Un modèle dérivé de la physique de l'instrument de mesure prenant en compte la saturation de l'appareil a été proposé. Ce modèle inclut notamment un pic commun à tous les métabolites et un phénomène de saturation spécifique pour chaque ions. Ce modèle a permis de créer une workflow qui estime ce pic commun sur des signaux peu bruités, puis l'utilise dans un filtre adapté sur tous les signaux. Son efficacité sur des données réelles a été étudié et il a été montré que proFIA était supérieur aux algorithmes existant, avait une bonne reproductibilité et était très proche des mesures manuelles effectuées par un expert sur plusieurs types d'appareils.
La seconde partie de cette thèse a portée sur le développement d'un outil de détection des similarités structurales d'un ensemble de spectre de fragmentation. Pour ce faire une nouvelle représentation sous forme de graphe a été proposée qui ne nécessite pas de connaître la composition atomique du métabolite. Ces graphes sont de plus une représentation naturelle des spectres MS/MS Certaines propriétés de ces graphes ont ensuite permis de créer un algorithme efficace de détection des sous graphes fréquents (FSM) basé sur la génération d'arbres couvrants de graphes. Cet outil a été testé sur deux jeux données différents et a prouvé sa vitesse et son interprétabilité comparé aux algorithmes de l'état de l'art.
Ces deux algorithmes ont été implémentés dans des package R, proFIA et mineMS2 disponibles à la communauté.

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Title : New approaches for the processing and the annotation of high-throughput metabolomics data obtained by High-Resolution Mass Spectrometry

Keywords : Signal processing, Frequent Subgraph Mining, Graph Mining, Metabolomics

Metabolomics is a phenotyping approach with promising prospects for the diagnosis and monitoring of several diseases. The most widely used observation technique in metabolomics is mass spectrometry (MS). Recent technological developments have significantly increased the size and complexity of data. This thesis focused on two bottlenecks in the processing of these data, the extraction of peaks from raw data and the annotation of MS/MS spectra.
The first part of the thesis focused on the development of a new peak detection algorithm for Flow Injection Analysis (FIA) data, an high-throughtput metabolomics technique. A model derived from the physics of the mass spectrometer taking into account the saturation of the instrument has been proposed. This model includes a peak common to all metabolites and a specific saturation phenomenon for each ion. This model has made it possible to create a workflow that estimates the common peak on well-behaved signals, then uses it to perform matched filtration on all signals. Its effectiveness on real data has been studied and it has been shown that proFIA is superior to existing algorithms, has good reproducibility and is very close to manual measurements made by an expert on several types of devices.
The second part of this thesis focused on the development of a tool for detecting the structural similarities of a set of fragmentation spectra. To do this, a new graphical representation has been proposed, which does not require the metabolite formula. The graphs are also a natural representation of MS/MS spectra. Some properties of these graphs have then made it possible to create an efficient algorithm for detecting frequent subgraphs (FSM) based on the generation of trees covering graphs. This tool has been tested on two different data sets and has proven its speed and interpretability compared to state-of-the-art algorithms.
These two algorithms have been implemented in R, proFIA and mineMS2 packages available to the community.