PROBABILISTIC METHODS FOR LIQUID CHROMATOGRAPHY MASS SPECTROMETRY DATA PRE-PROCESSING

JOE WANDY

SUBMITTED IN FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF
Doctor of Philosophy

SCHOOL OF COMPUTING SCIENCE

COLLEGE OF SCIENCE AND ENGINEERING
UNIVERSITY OF GLASGOW

AUGUST 2016

Abstract

In recent years, the large-scale, untargeted studies of compounds that serve as workers in the cell (proteins) and the by-products of essential life-sustaining chemical processes (metabolites) have provided insights into a wide array of fields, such as medical diagnostics, drug discovery, personalised medicine and many others. Measurements in such studies are routinely performed using liquid chromatography mass spectrometry (LC-MS) instruments. From these measurements, we obtain a set of peaks having mass-to-charge, retention time (RT) and intensity values. Before further analysis is possible, the raw LC-MS data has to be processed in a data pre-preprocessing pipeline. In the alignment step of the pipeline, peaks from multiple LC-MS measurements have to be matched. In the identification step, the identity of compounds that generate the observed peaks have to be assigned. Using tandem mass spectrometry, fragmentation peaks characteristic to a compound can be obtained and used to help establish the identity of the compound. Alignment and identification are challenging because the true identities of the entire set of compounds in the sample are unknown, and a single compound can produce many observed peaks, each with a potential drift in its retention time value. However, observed peaks are not independent — there exists structural dependencies among the observed peaks as multiple peaks are related through being attributed to the same underlying compound.

The aim of this thesis is to introduce methods to group these related peaks and to use these groupings to improve alignment and assist in identification during data pre-processing. Firstly, we introduce a generative model to group related peaks by their retention time. This information is used to influence direct-matching alignment, bringing related peak groups closer during matching. Investigations using benchmark datasets reveals that an improved alignment performance is obtained from this approach (Chapter 4). In Chapter 5, we expand the grouping process to consider mass information as well, resulting in a model that performs the grouping of related peaks by their explainable mass relationships, RT and intensity values. Through a second-stage process that matches related peak groups, peak alignment is produced. Experiments on benchmark metabolomics datasets show that an improved alignment performance is obtained, while uncertainties in matched peaksets can also be extracted

from the method. Next, we improve upon the two-stage method described before and introduce a model that performs the clustering of related peaks within and across multiple LC-MS runs at once, allowing for matched peaksets and their respective uncertainties to be naturally extracted from the model (Chapter 6). Finally, we look at fragmentation peaks used for identification and introduce a topic model to group related fragmentation features. These groups of related fragmentation features potentially correspond to substructures shared by metabolites and can be used to assist data interpretation during identification (Chapter 7). This final section corresponds to a work in progress and points to many interesting avenues for future research.

Acknowledgements

This thesis would not have existed without the support of everybody whom I have come in contact over the past years. In particular, I would like to express my gratitude to my supervisor, Simon Rogers, for his encouragements and insightful discussions. I would also like to thank Alice Miller for her guidance. I have also benefited from collaborations with others — in particular, Ronan Daly, Justin jjvd. Hooft, Karl Burgess, Yoann Gloaguen, Naomi Rankin, alongside other staff at Glasgow Polyomics, such as Pawel, Mani, Gavin, Graham, Julian, Galaxy David, Stefan, etc.

I would also like to show my appreciation to everyone from the IDI group, in particular those I have shared my office space with, in no particular order they are: Faiz, Shimin, Colin, Tommi, Daryl, Lorna, George, Xiaoyu, Antoine.

And finally I would like to thank my family for their continuous support and encouragement.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Liquid chromatography combined with mass spectrometry (LC-MS) has emerged as one of the most popular methods of measurements in the untargeted study of proteins (proteomics) and metabolites (metabolomics). Proteins and metabolites serve as crucial building blocks in the body and play a vital role in the cellular maintenance of any organism. Metabolomics in particular is regarded as the -omics that is the closest to the phenotype: changes to the physical traits of an organism is often expressed in the metabolome. Understanding and characterising the proteome and metabolome provide important insights into the working of any biological system.

Before the raw LC-MS data can be used for further analysis, it has to be processed in a data pre-processing pipeline. This starts from the initial step of peak detection, where the observed peaks having m/z, retention time (RT) and intensity values are extracted from the raw data. The two important steps that follow after peak detection are the alignment and identification steps. In most studies, multiple samples are obtained and measured (producing biological replicates) or alternatively a sample is run through the LC-MS instruments multiple times (producing technical replicates). Alignment refers to the matching of these peaks across multiple LC-MS runs. In identification, we seek to associate the information on which compounds generate the observed peaks. Fragmentation data, where parent peaks are processed through a second-stage mass spectrometry, provides an additional information as to the identity of a compound in the form of patterns of fragment peaks that are characteristic to the compound.

In many cases, LC-MS data pre-processing is challenging. The lack of knowledge in the complete composition of compounds in a sample means that we do not know for certain which compounds are present in the sample. Compounds ionise differently during mass spectrometry, while a single compound can produce multiple observed peaks, making data interpretation difficult as there is no one-to-one correspondence between the observed peaks and the compounds that generate them. While the m/z information of a peak is generally

preserved across runs, retention time drift means the observed RT values can vary among peaks produced on different instruments or even peaks produced on the same instrument but measured at a different time period. This makes alignment difficult. Identification using fragmentation data is also hampered by the limited coverage of spectral databases to compare the observed fragmentation pattern against.

However, peaks generated from the same underlying compound are not independent. They are structurally related in a chemical manner e.g. through being the ionisaton product peaks of the same compound. We reason that this structural dependencies can be used to improve alignment. In a similar manner, fragmentation spectra, which provides the characteristic fingerprints of compounds, also contains structural information where a subset of fragment peaks may correspond to a shared chemical substructure in a class of compounds. In this thesis, we show that through generative modelling, the structural dependencies of these peaks can be revealed and exploited to improve or enhance the alignment and identification steps. Moreover through generative modelling, alignment uncertainties can also be quantified, allowing the user to control the level of uncertainty they desire from matched peaksets.

1.1 Thesis Statement

Untargeted liquid chromatography mass spectrometry data pre-processing is a challenging task that is often subject to errors and inaccuracies. Much of this can be attributed to the complexity of the LC-MS data itself and also to the lack of knowledge as to which compounds are present in the sample. However, the structural dependencies in the observed peak data means that through generative modelling, we can explain the relationships between peaks, allowing us to produce groups of related peaks that can be used to improve or enhance the alignment and identification steps of LC-MS data pre-processing.

1.2 List of Contributing Papers

The work described in this thesis has led to the following publication:

1. Wandy, J., Daly, R., Breitling, R., Rogers, S. (2015). ”Incorporating peak grouping information for alignment of multiple liquid chromatography-mass spectrometry datasets”. *Bioinformatics*, 31(12), 1999-2006.

Additionally the following manuscripts are still under review:

1. Wandy, J., van der Hooft, J. J., Rogers, S. (2016). "Ionization Product Clustering to Improve Peak Alignment in LC-MS-based Metabolomics" submitted to *Bioinformatics*.

2. van der Hooft, J. J., Wandy, J., Barrett, M., Burgess, K. V., Rogers, S. (2016). "Topic Modeling for Untargeted Substructure Exploration in Metabolomics" submitted to *Proceedings of the National Academy of Sciences* (PNAS).

Chapter 4 of this thesis is based on the first published paper, Chapters 5 and 7 are based on the two manuscripts that are under review. Finally, the author also contributed to the following work but it is not a part of this thesis:

1. Daly, R., Rogers, S., Wandy, J., Jankevics, A., Burgess, K. E., Breitling, R. (2014). MetAssign: probabilistic annotation of metabolites from LCMS data using a Bayesian clustering approach. *Bioinformatics*, 30(19), 2764-2771.

1.3 Overview of Thesis and Research Contributions

The contributions of this thesis are:

- A method that combines direct-matching and related peak grouping information to improve alignment.

- A generative model (**PrecursorCluster**) that groups related peaks in the same LC-MS run by their ionisation product (IP) relationships, producing IP clusters. This is described alongside methods that use the resulting IP clusters to produce a better alignment.

- A generative model (**HDP-Align**) that groups related peaks in the same and across LC-MS runs in a flexible manner. From this model, we can extract alignment and furthermore, it allows for the probabilities of matching of certain peaksets to be quantified.

- A generative model (**MS2LDA**) that groups related fragmentation features in tandem mass spectrometry data. From this model, we can extract patterns of fragmentation features that potentially correspond to substructures shared by metabolites. A visualisation module is also created to assist in the exploration of the results.

The remainder of this thesis is structured as follows:

- **Chapter 2** discusses the background literature that this thesis is built upon. In particular, the chapter explains the nature of the LC-MS data and the necessary pre-processing steps before the data can be used for further analysis, including the challenges faced in the data pre-processing steps.

- **Chapter 3** introduces probabilistic modelling, with a particular focus on the construction of mixture models and other related generative models that are used in the rest of the thesis.

- **Chapter 4** presents an approach that combines matching and clustering information to produce a better alignment result.

- **Chapter 5** presents the PrecursorCluster model to group related ionisation product peaks into IP clusters. This chapter also introduces ways these IP clusters can be used to produce an improved alignment performance.

- **Chapter 6** presents the HDP-Align model to perform the clustering of related ionisation product peaks within and across multiple runs.

- **Chapter 7** presents the MS2LDA model to capture the structural dependencies of peaks in fragmentation data. It also introduces a visualisation module to aid in the analysis of the results from the model.

- **Chapter 8** presents a summary of the work and contributions. It also highlight the avenues for future research based on the work done so far, and finally it concludes this thesis.

Chapter 2

Computational Mass Spectrometry Analysis

2.1 Introduction

The three major types of macromolecules that are fundamentally essential to all life on Earth: deoxyribonucleic acid (DNA), ribonucleic acid (RNA) and proteins. The central dogma of molecular biology states that *DNA is transcribed into RNA, which is translated into proteins*. Since its initial proposal, the central dogma model has been challenged and expanded to acknowledge other factors that can influence the transcription and translation processes. For instance, the reverse flow of information from RNA to DNA is possible but was not in the initial model. Nevertheless, the central dogma is broadly useful to explain how genetic information can flow in a biological system, starting from DNA to RNA to proteins.

DNA is the basic storage unit of genetic information. In a rather simplified view, the flow of information in a biological system begins from the double-helix strands of the DNA as the starting point. A DNA strand consisting of a series of linked nucleotides subunits. Each nucleotide is a molecule composed of a sugar molecule (deoxyribose), a phoshoric acid and a nitrogenous base. The base in DNA can be either adenine (A), thymine (T), guanine (G) or cytosine (C), and together they form the four well-known 'alphabets' of the DNA. Bases are complementary in their pairing through hydrogen bonds, such that A pairs only with T, and G with C. It is this pairing that produces the double helix structure of the DNA.

Regions of the DNA that code for specific proteins are called genes, however DNA is not the direct template for protein synthesis. Rather, DNA is *transcribed* into RNA. The same information is encoded in RNA as its originating DNA strand, but with the crucial difference that the subunits (nucleotides) of RNA has ribose as the sugar molecule and uracil substituted in place of thymine as one of the bases. In this manner, the four alphabets of RNA are adenine

(A), uracil (U), guanine (G) and cytosine (C).

After the transcription process, a class of RNA molecules known as the messenger RNA (mRNA) serves as the template for protein synthesis. Compared to the relatively inert DNA, mRNA is biochemically active and allows for genetic information to be transferred to outside the nucleus. The ribosome, a part of the translational apparatus of the cell, then reads mRNA and *translates* it into proteins. A sequence of three RNA nucleotides, terms a codon, codes for a particular amino acid, which is the building block of proteins. Proteins serve critical roles in an organism by participating in nearly all cellular processes: performing cellular maintenance, catalysing chemical reactions and carrying other functions essential to life. Proteins also serve as the biochemical machineries involved in carrying out DNA replication and the transcription and translation processes themselves to produce more proteins.

In total, there are 20 different types of amino acids used as the building blocks of proteins (Table 2.1). By allowing multiple codons to encode for the same amino acid, redundancies are built to deal with transcription errors. For instance both 'AAT' and 'AAC' codons correspond to the asparagine amino acid. An amino acid consists of a central carbon atom surrounded by an amine group (-NH_2), a carboxylic group (-COOH) and a side chain specific to the amino acid. Through the loss of water molecule, amino acids can be chained to each other through peptide bonds. A short chain of amino acid residues form a peptide, and in a longer chain, they fold into a fixed structure to form a protein. The function of a protein is directly determined by its three-dimensional structure. As each amino acid can be described by a unique letter drawn from a set of 20 chemical alphabets in Table 2.1, a protein can be succinctly described by a string of its peptides.

Apart from proteins, numerous other chemical reactions essential for sustaining life also happen inside a cell, including crucially, the breaking of organic compounds into energy and the production of other cellular building blocks involved in the transcription and translation processes. Together these chemical reactions comprise the *metabolism* of an organism. In catabolic reactions, large organic molecules within a cell are broken into energy and smaller molecules. These serve as the input to anabolic reactions, producing the basic building blocks of a cell such as proteins and nucleic acids. Both anabolic and catabolic reactions are usually catalysed by enzymes, and together these two reactions comprise the metabolism of an organism. *Metabolites* are small molecules (usually defined as less than 1000 Da) involved during or produced as the by-products of metabolism. Through the help of various enzymes, metabolites are transformed from one form to another in a series of chemical reactions as part of the metabolic pathways. Some examples of common metabolites are the various amino acids, fatty acids, vitamins, carbohydrates and many others. The overall set of metabolites that can be found within an organism is collectively called the *metabolome*.

As illustrated in Figure 2.1, each sub-field of computational biology focuses on the entities

Amino Acids	RNA Codons	Amino Acids	RNA Codons
Isoleucine (I)	AUU, AUC, AUA	Serine (S)	UCU, UCC, UCA, UCG, AGU, AGC
Leucine (L)	CUU, CUC, CUA, CUG, UUA, UUG	Tyrosine (Y)	UAU, UAC
Valine (V)	GUU, GUC, GUA, GUG	TrypUophan (W)	UGG
Phenylalanine (F)	UUU, UUC	Glutamine (Q)	CAA, CAG
Methionine (M)	AUG	Asparagine (N)	AAU, AAC
Cysteine (C)	UGU, UGC	Histidine (H)	CAU, CAC
Alanine (A)	GCU, GCC, GCA, GCG	Glutamic acid (E)	GAA, GAG
Glycine (G)	GGU, GGC, GGA, GGG	Aspartic acid (D)	GAU, GAC
Proline (P)	CCU, CCC, CCA, CCG	Lysine (K)	AAA, AAG
Threonine (T)	ACU, ACC, ACA, ACG	Arginine (R)	CGU, CGC, CGA, CGG, AGA, AGG

Table 2.1: The 20 amino acids and the RNA codons that encode them.

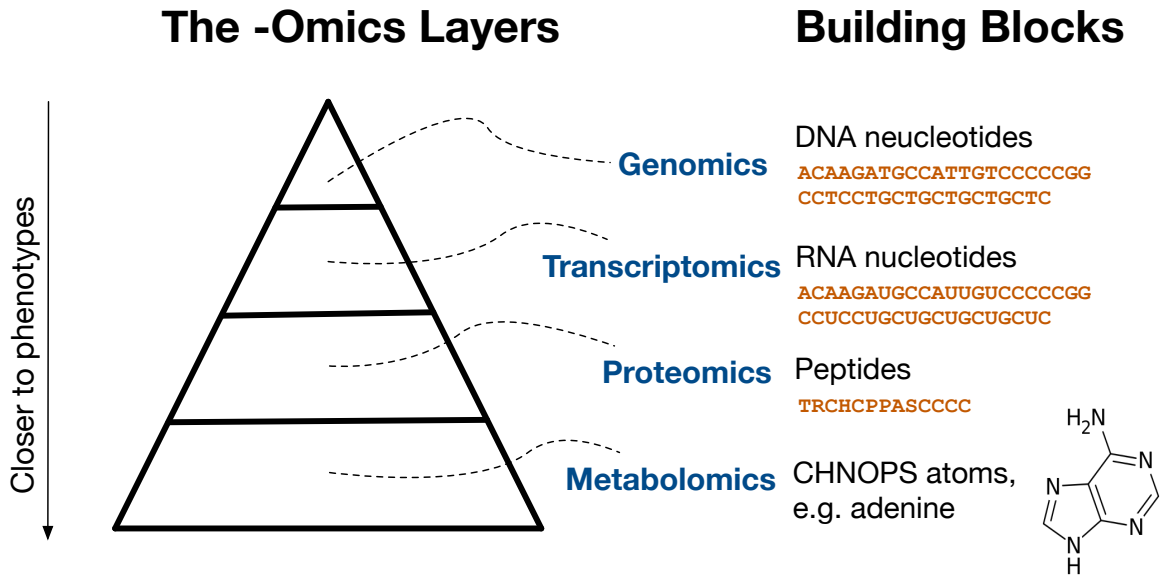


Figure 2.1: The layers of -omics and their building blocks.

and processes involved in a stage of the central dogma. Genomics is concerned with the large-scale study of the entire DNA in the organism (the genome) and how the genes encoded in the genome interact with each other. Transcriptomics focuses on understanding the complete set of mRNA (the transcriptome), particularly those that correspond to protein-encoding genes and measurements on their abundance in the sample. Proteins and their large-scale identifications and quantifications are studied in proteomics. Metabolomics studies the metabolome on a large scale, usually for the purpose of identifying and quantifying the differences of metabolite compositions in a particular organism or tissue under various experimental or physiological conditions.

Moving through the successive -omics layers in Figure 2.1 and getting closer the phenotype introduce greater complexity due to the increased number of ways to putting the building blocks of each -omics layer together. The building blocks of the genome are the nucleotides of the DNA, while in the transcriptome, the building blocks are the nitrogenous bases that comprise the RNA. There are only four possible alphabets in the genome and transcriptome. In proteomics, the object of interest, proteins, is a chain of amino acid residues. There are 20 possible alphabets of amino acids residues listen in Table 2.1. The small molecules in metabolomics have atoms as their building blocks, with the elements **C**arbon, **H**ydrogen, **N**itrogen, **O**xygen, **P**hospor and **S**ulphur (CHNOPS) that can be arranged in many chemically-plausible configuration. Furthermore, unlike the genome that is relatively static, the proteome and metabolome of an organism are also considerably more dynamic. The expression of proteins and metabolites are governed by various complex, interacting factors. In a process called post-translational modification [3], proteins can be chemically modified after synthesis in a way that completely alters its structure and folding stability, e.g. through phosphorylation (the addition of a phoshate group) or methylation (the addition of a methyl group). Metabolites expression can also change in response to the cellular sytems cellular [4] or environmental factor [5]. As a result, the knowledge of the DNA sequence alone is not sufficient to predict the proteins and metabolites that may be expressed in an organism. However, the metabolome is considered closest to the physically observed properties (phenotypes) of that organism [6], so changes to phenotype are often most readily observed in the metabolome. Studying the metabolome therefore provides us with an instantaneous 'snapshot' of the chemical activities that occur in the cell, leading to an understanding of how cellular processes behaves and possibly an explanation of how certain phenotypes are expressed.

2.2 Measurement Technologies

Sequencing technologies, in particular next-generation sequencing (NGS) machines such as Illumina and Ion Torrent, have been instrumental in revolutionising genomics by making possible the high-throughput and rapid sequencing of the entire DNA sequence from a sample [7]. Transcriptome relies on DNA micro-array technologies and more recently, have been increasingly performed by NGS sequencing as well. Nuclear magnetic resonance (NMR) spectroscopy and mass spectrometry (MS) are the two widely used measurement technologies for proteomics and metabolomics. Before we can understand the principle behind NMR spectroscopy and mass spectrometry, we need to take a detour and talk about atoms.

Atoms are the small building blocks of matter. An atom has a nucleus at the centre, which consists of positively charged protons and neutrons with no charge. Electrons, having negative charge, are bound to the neucleus through electromagnetic force. The overall charge of the atom is therefore determined by the number of electrons and protons that it has. The atom is called a positive ion when there are more protons than electron, otherwise it is a negative ion. Two or more atoms held via chemical bonds comprise a compound. The molecular mass of a compound is the sum of the molecular mass of its elements, measured in Dalton (Da), where one Da is $\frac{1}{12}$ of the molecular mass of the carbon element (^{12}C). Elements in nature occur as isotopes. Isotopes are naturally occurring elements that have the same number of protons (same atomic number) but different number of neutrons (different molecular masses). Each elements has many isotope species, for instance carbon has two isotopes: ^{12}C with molecular mass 12.000000 at 98.890% abundance in nature, and ^{13}C with molecular mass 13.003355 and 1.110% abundance.

NMR spectroscopy operates on the principle of measuring the energy absorption of certain nuclei as radio frequency is applied. The nucleus of an atom possesses an angular moment, called spin. A nucleus with a spin of 1/2 develops a magnetic field, and when placed in an external magnetic field, a nuclei can either align itself with the external field (a lower energy state) or against the external field (a higher energy state). In NMR spectroscopy, initially most nuclei will be in their ground state of being in alignment with the external magnetic field, but when radio waves are applied, the nuclei in the lower energy state can absorb the energy and move to the higher energy state (their spin flip). When the radio waves are removed, the energised nuclei relaxes back to the lower energy state. The fluctuation of the magnetic field during relaxation is called 'resonance' and can be measured in the form of a current in the magnetic coil around the sample, resulting in peaks in an NMR spectrum. Many isotopes naturally occurring in an organic compound, e.g. ^{1}H and ^{13}C, have a spin of 1/2 and can therefore be measured by NMR spectroscopy. From NMR measurements, signals in the time-domain is obtained, and through Fourier transform, this signal is converted from the time domain to the frequency domain. Before data analysis is possible, the

resulting NMR spectra are processed in a data pre-processing pipeline. This includes steps like baseline correction, noise filtering, alignment, and compound identification [8]. For identification, spectra are annotated through comparisons to databases that contain reference spectra either developed in-house or publicly available (e.g. BioMagResBank [9], Madison Metabolomics Consortium Database [10], and many others). Identification is one of the greatest challenges in NMR analysis [11], although in recent years, several methods such as BATMAN [12] and IQNMR [13] have been introduced that aim to automate this process.

As an alternative to NMR spectroscopy, mass spectrometry operates by ionising compounds in the sample, producing charged ions that are separated by their mass-to-charge (m/z) ratio. During mass spectrometry, the compounds to be analysed (metabolites or peptide fragment) are introduced into the ionisation source of the MS, and depending on the ionisation mode used, these compoundd produce positively or negatively charged ions. They travel through the mass analyser and arrive at the detector at a different rate due to each ion having different mass-to-charge (m/z) ratios. The detector measures the ions that arrive and produce signals in form of a mass spectrum, showing the relative abundance of detected ions at different m/z ratios. MS instruments can be ranked by the ascending order of their resolving powers of their mass analyser: **(1)** time-of-flight MS, **(2)** quadropole MS, and lastly **(3)** Fourier transform ion-cyclotron MS. A higher resolving power corresponds to a better ability of the instrument to detect small differences in mass-to-charge (m/z) ratios. Having a higher resolving power is generally very useful when trying to identify which metabolites are present in the sample. Modern high-precision MS instruments have very accurate resolving power, with accuracy up to several parts-per-million. The difference between the observed mass-to-charge value to the exact-mass-to-charge value of a compound is the mass accuracy of a mass spectrometry instrument, measured in parts-per-million, i.e. mass accuracy $= 1e6 * \frac{\text{(observed m/z}-\text{exact m/z)}}{\text{exact m/z}}$.

The main advantage of NMR spectroscopy over MS is that its spectra is very high reproducibility since the same compound structure always produces peaks at the same locations in the spectra. Absolute quantification of the abundance of the compounds is possible in NMR as the signal intensity in NMR spectra is directly proportional to the concentration of protons in the nucleus of the compounds. In MS, often only the relative abundance (with respect to some reference compounds of known concentration) can be obtained. However, while the resulting spectra from NMR provides information on the structure of the metabolite, certain regions in the spectra can also be crowded with many overlapping metabolite signals [11], potentially hindering identification. NMR also has a lower sensitivity than mass spectrometry, which limits the number of metabolites that can be detected from NMR spectra. For more detailed comparisons of NMR vs. MS, the reader is directed to [11]. As it stands, the two approaches are often seen as complementary rather than competitive.

In direct injection mass spectrometry, the sample is introduced into the MS at a constant flow. However the ionisation capacity of MS is limited, and in what is called the ion su-

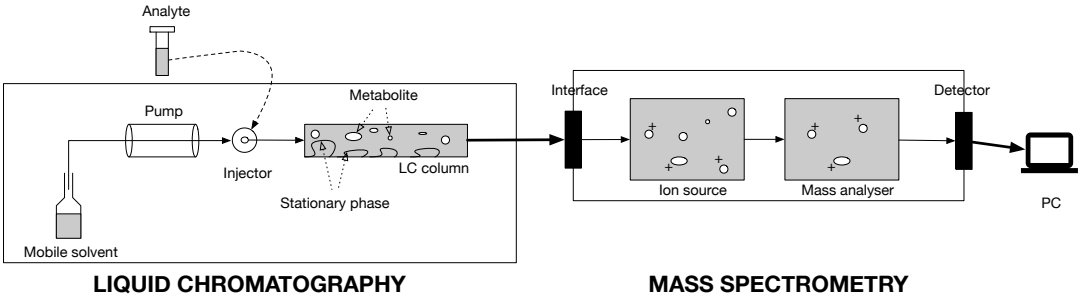


Figure 2.2: A typical LC-MS set-up. High performance liquid chromatography instruments are used to separate metabolites (by their chemical properties) in the sample before they are gradually introduced into the mass spectrometer.

pression effect, compounds can compete for charges during ionisation — resulting in certain compounds not being ionised and detected in the mass spectra [14]. Separating compounds as they gradually elute from the chromatographic column at a different *retention time* (RT) into the MS is often preferred. Additionally, from chromatographic separation, the retention time of observed peak reflects the underlying biochemistry of the compounds and can serve as an additional information to deduce their identities [15]. Particularly in large-scale untargeted studies, MS is often coupled to a chromatographic separation technology such as liquid chromatography (LC), forming the combined set-up of LC-MS (Figure 2.2).

As illustrated in Figure 2.2, during liquid chromatography, the solvent containing the analytes (metabolites) is introduced and pumped into the stationary phase that is part of the chromatographic column. Metabolites elutes at different time through their interactions with the capillary in the column, based on their biochemical properties (e.g. their hydrophobicity, polarity, molecular shapes etc). In the LC-MS set-up, metabolites that elute from liquid chromatography are then vaporised and ionised inside the mass spectrometer. Ionisation in an LC-MS setup is usually performed via electrospray ionisation (ESI). In ESI, the sample analyte is dissolved into a solvent and sprayed through an electrospray (a highly charged needle) creating charged droplets. As the charged droplets travel through the vacuum of the MS, they evaporate, creating charged electric fields on the surfaces. In the strong electric field of the MS, ions on the surface of the droplets have enough energy to separate, generating charged molecular ions and their corresponding fragment ions. The generated ions are separated by the mass analyser inside the MS instrument according to their m/z (mass-to-charge) ratios and the detected signal abundance for a particular m/z value. As ESI requires a continuous supply of dissolved analytes, it can be directly coupled to LC, so often it is the preferred method of ionisation in LC-MS.

2.3 LC-MS Analysis in Metabolomics

The raw data produced from an LC-MS set-up is a collection of mass spectra from each scan over a range of elution time. Each MS measurement of compounds that elute at the same or similar retention time is called a scan. A mass spectrum in each scan is the two dimensional representation of m/z values of charged ions to signal intensities (Figure 2.3C). The sum the signal intensities across all mass spectra, called the total ion chromatogram or TIC (Figure 2.3D) shows how compounds elute over time over all m/z values. The TIC plot can be too crowded, so given a specific m/z range to inspect, the extracted ion chromatogram (EIC) plot shows the total signal in that m/z range vs. RT (Figure 2.3E). The m/z range for inspection in the EIC is usually selected based on the prior knowledge of what signal a compound is supposed to produce in the spectra.

As shown in Figure 2.3B, the raw LC-MS data can also be seen as a 3D image containing peaks that can be characterised by a set of vector of m/z, retention time and intensity. This raw LC-MS data is noisy, so pre-processing has to take place before analysis can be performed and biological conclusion drawn. Generally, the main steps of LC-MS data pre-processing takes the form of a sequential pipeline shown in Figure 2.4. Note that Figure 2.4 illustrates an exemplar pipeline. In practice, many variations of this exemplar pipeline exists. For instance, the gap filling and the peak grouping steps can be omitted, the noise filtering step can be performed before peak alignment, no visualisation is produced from the output of identification, etc. The following sections explain in details the key steps of the LC-MS data processing pipeline in Figure 2.4.

2.3.1 Raw Data Importing & Peak Detection

The LC-MS data pre-processing pipeline starts with the raw data importing of vendor-proprietary format into an open XML-based format, such as mzXML [16] or mzML format [17]. Peak detection is applied to the imported LC-MS data to produce peaks. Each peak feature is characterised by its m/z, RT and intensity values. The CentWave algorithm [18] from XCMS is one of the more widely used peak detection method in metabolomics. It is particularly suitable for modern metabolomics data that are generated from instruments having a high mass accuracy. CentWave extracts regions of interest from the data. Chromatographic analysis of the EIC from each region of interest is performed using continuous wavelet transform is used to detect candidate chromatographic peaks. For each candidate peak, once its chromatographic peak boundaries have been identified, the centroid m/z value of a peak feature is defined as the weighted mean of the m/z values within the boundaries. Similarly, the intensity of a peak feature is defined as the maximal intensity value in the chromatographic peak boundaries. The signal-to-noise ratio of each candidate peak is calculated

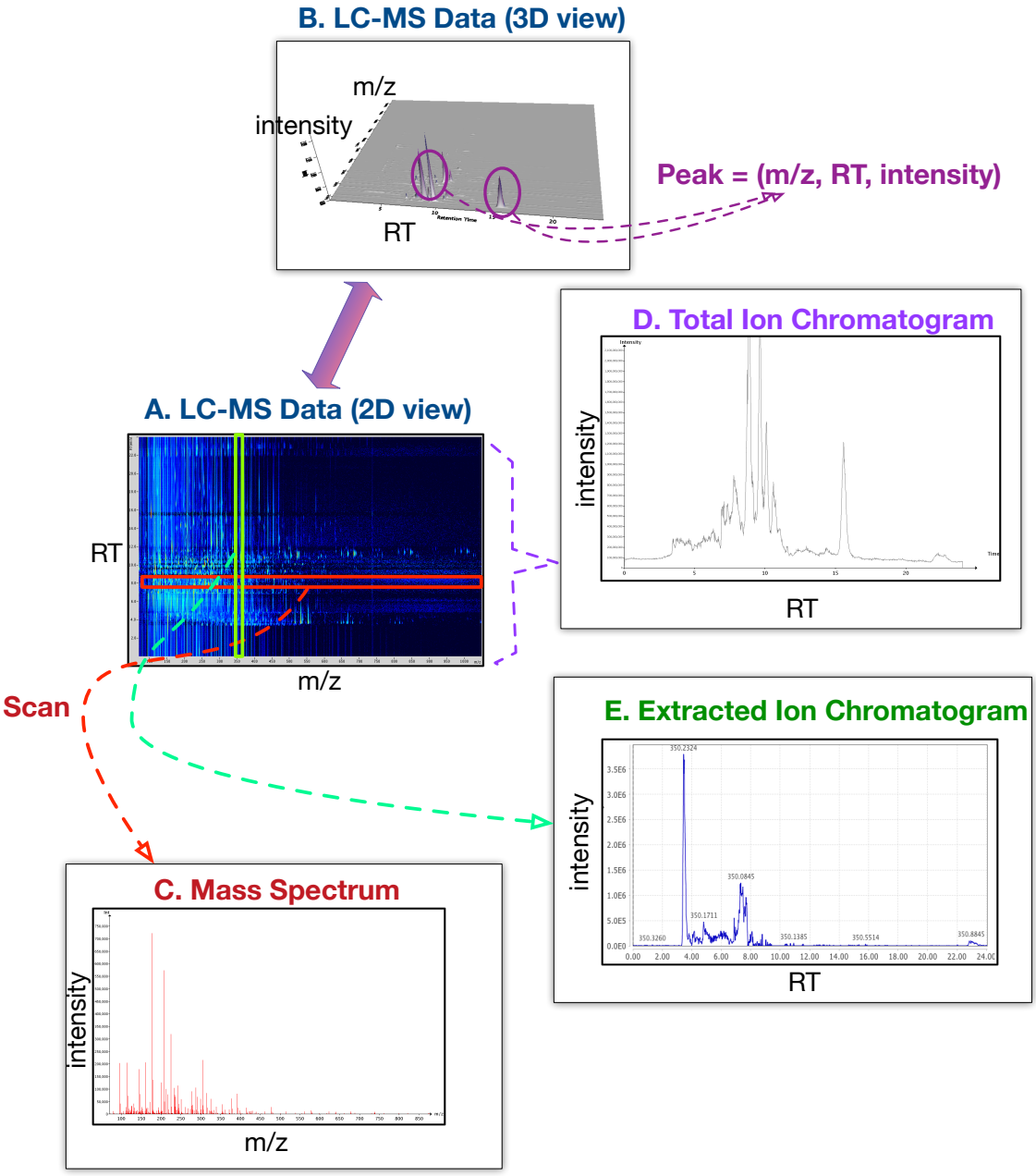


Figure 2.3: The resulting raw data (ion chromatograms) produced from an LC-MS experiment. We can view the data as a 2D profile seen from the top **(A)** or a 3D profile **(B)**. A peak in the data is thus characterised by its intensity value on the m/z and retention time axes. From a scan, a slice of the data on the m/z axis is the mass spectrum **(C)**. A collection of mass spectra is produced over the whole range of retention time. Summing over all scans produce the total ion chromatogram (TIC) **(D)**, while plotting the intensity values vs. RT for a particular m/z range produces the extracted ion chromatogram (EIC) **(E)**.

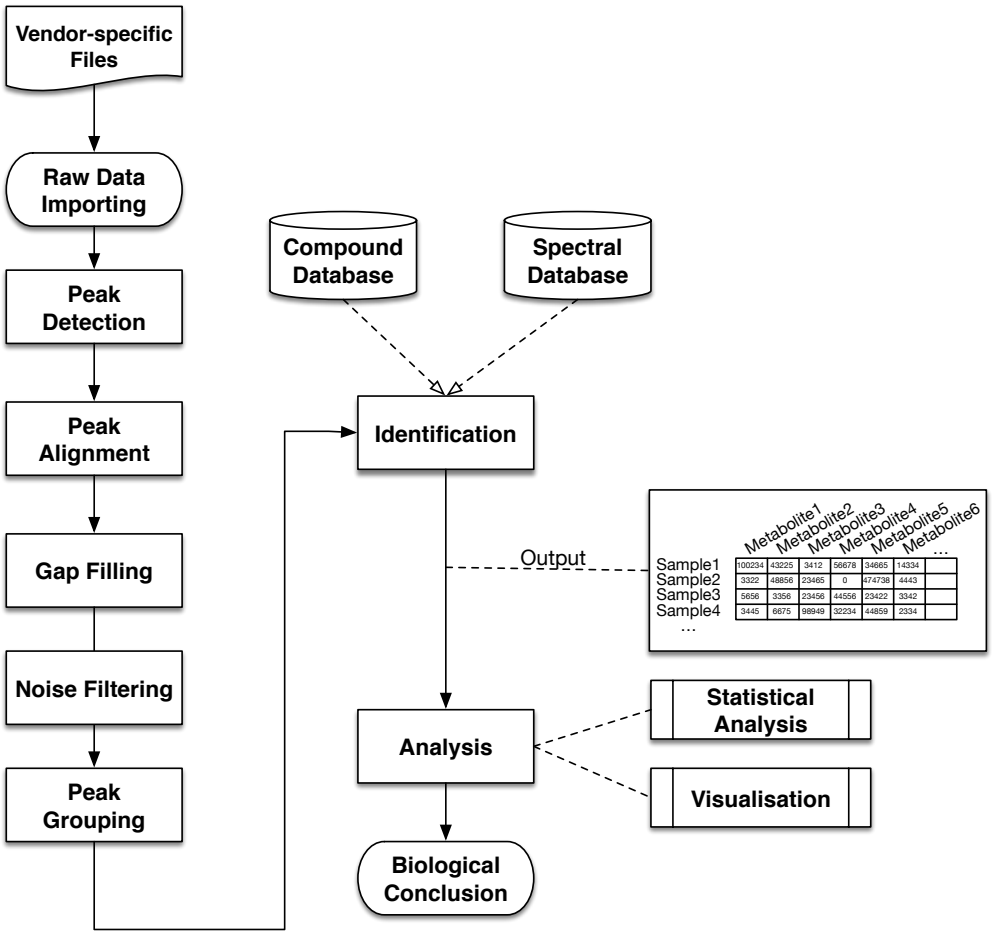


Figure 2.4: An exemplar pre-processing pipeline of LC-MS metabolomics data.

and if it is lower than the thresholded defined by the user, the candidate peak is rejected. As an alternative peak detection method, the MZmine 2 [19] software suite is also widely used.

A survey of the many different approaches for peak detections can be found in [20, 21, 8], however it is important to note that most peak detection methods are sensitive to the choice of parameters [18], with a method potentially producing different results when its parameters are varied. For instance, CentWave requires as user-defined parameters the mass deviation in parts-per-million (which is usually set based on the mass accuracy of instrument), the minimum width of the chromatographic peak and a signal-to-noise threshold. Setting a width that is too narrow or a signal-to-noise threshold that is too high can potentially lead to peaks that should be detected instead marked as missing.

2.3.2 Peak Alignment

Following peak detection, peak alignment is performed to match peaks that are the same across samples. An alignment method takes as input multiple lists of peaks — one from each LC-MS run — and produces as output a list of *aligned peaksets*. Each aligned peakset is a set of peaks coming from different runs that are considered to be *correspondent* and have to be matched. Alignment is necessary because experiments in biology usually involve the comparison of multiple samples. Samples can be produced as either biological or technical replicates. Biological replicates are obtained from the same organism studied under varying conditions and exposed to different factors (e.g. treatment or no treatment). Biological replicates are necessary to determine entities that are differentially expressed across samples. In contrast, technical replicates are obtained from the same sample analysed multiple times. Technical replicates are necessary to account for the variability and measurement errors throughout the experiment. In this manner, each replicate, whether biological or technical, is measured through the LC-MS instrument. This produces an LC-MS run for each replicate.

An initial approach towards alignment of multiple LC-MS runs would be to spike a known amount of internal standards into each sample before running them through the LC-MS instruments. Standards are compounds of known concentration that produce peaks at well-defined m/z and RT values. The peaks generated from these standards can be used as 'landmark' peaks to linearly shift the retention time in each sample, usually against a reference sample. Alternatively, stable-isotope labelling experiments exploit the fact that atoms have isotopes, which when measured in mass spectrometry, produce a distinct pattern of peaks that follow the binomial distribution. This information can be used to aid peak alignment and identification. In a labelling experiment, two samples are prepared: one from cells that grow in a normal medium and another from cells that grow in isotopic reagents. The two samples are combined and measured as a single LC-MS run. A metabolite from the normal medium

and its corresponding isotopic counterpart have the same chemical formula and structure and hence will appear at close retention time, however the distinctive pattern of peaks produced from the isotopic metabolite makes it possible to trace the peaks back to the metabolite that produce them [22]. This makes alignment easier. However, labelled experiments consume expensive reagents, are more difficult to prepare and harder to compare across laboratories and to various mass spectral databases online for identification. Consequently, it is common for large-scale untargeted LC-MS experiments, where the identities of the metabolites of interest are not known in advance, to be performed label-free without relying on such labelling information. This is called label-free experiments. To be comparable, the results from these label-free experiments need to be aligned, using peak alignment methods.

Broadly speaking, the main challenge in the peak alignment stage of label-free experiments is the poor reproducibility of retention time, with potentially large non-linear shifts and distortions across LC-MS runs produced from different analytical platforms or even the same platform over time [23]. There is often a large amount of variations in the retention times across the replicates. Retention time variation could be due to instrument-specific factors (the condition of the chromatographic column itself. including flow rate variations, gradient slope and temperature [24]) or experiment-specific factors (e.g. instrument malfunctions or columns that need be replaced mid-experiment). Both factors are difficult to control, even in a careful experimental setting. Consequently, most alignment methods correct for those shifts and distortions by finding a mapping function f that maps peaks from one run to another. Depending on how they find f, alignment methods can be divided into two broad categories: **(1)** warping-based methods and **(2)** direct-matching methods.

Warping-based Alignment Methods

Warping-based methods seek to model the RT drifts between runs. In the past, many warping-based methods operate by aligning the whole ion chromatograms (profile data) directly before peak detection. Since this alignment step is performed before peak detection, warping-based methods that operate on profile data do not depend on the correctness of detected peaks. In this manner, the profile data being aligned is reduced to a simpler form by using the total ion chromatograms (TIC) as a representation of the entire data — frequently ignoring the rich information present in the m/z dimension of LC-MS data. As a consequence, warping-based methods that rely on profile information alone might not perform well for the alignments of the typical LC-MS data produced from complex mixtures – frequently having a lot of peaks of different m/z values co-eluting at similar retention times.

Many warping-based methods that operate on profile data are based on dynamic programming. In dynamic programming, all possible local solutions are evaluated but computed only for each sub-problem. In theory, this allows for an optimal global solution to be obtained ef-

ficiently. In practice, exact dynamic programming solutions are often intractable when a large number of runs need to be aligned at once due to their high time complexity when aligning multiple profile data simultaneously. As such, many of these methods aligns runs in a hierarchical pairwise manner. Some examples of well-known warping-based methods that operate on profile data are highlighted below:

1. **Dynamic Time Warping (DTW)** [25] performs a pairwise alignment of runs using the RT information only. The TICs being aligned are first discretised along the RT axes. Finding the alignment path is accomplished by setting up an alignment matrix and obtaining the best warping path that minimises the global distance in the alignment matrix. Three weight factors that computes the penalty for matches, expansion and compression are defined. The optimal warping path is obtained by applying dynamic programming principle and tabulating intermediate results in the alignment matrix (in a manner similar to global sequence alignment for DNA sequences). The best warping path can then be read by backtracking from the final entry of the alignment matrix to the start.

2. **Correlation Optimised Wrapping (COW)** [26] operates in a manner similar to DTW by using the discretised TICs. COW divides the RT axes of replicates into segments. Each segment boundary can change within some user-specified slack parameter. COW then produces an alignment by finding the path across segments that has the highest sum of correlations. An alignment matrix is set up, and different segment boundaries can be shifted to maximise the global correlations between the two replicates being aligned using dynamic programming. In [24], COW is combined with a component detection algorithm (CODA [27]) that removes noisy signal and background noise from the mass chromatograms, aligning only regions containing high-quality information.

3. **Parametric Time Warping (PTW)** [28] produces pairwise alignment by using a second degree polynomial for mapping time between chromatograms. Coefficients of the polynomial are optimised by minimising the sum of squared residuals between the reference and aligned chromatograms. PTW performs much faster than COW. However, the quadratic polynomial model proposed in PTW, while simpler to describe, might not be sufficient to capture the complexity in non-linear retention time drifts across LC-MS data [23]. Semi-parametric Time Warping (STW) extends upon PTW and uses a series of B-splines as the mapping function. Optimising the warping coefficients in STW is done iteratively.

4. **Continuous Profile Mode (CPM)** [29] aligns multiple LC-MS data in a time series using a hidden Markov model-based approach. Each observed chromatogram profile is

considered to be a time series of noisy signals sampled from a canonical latent profile. Parameters of the model are trained using the Expectation-Maximisation algorithm. The actual alignment of observed profiles to the latent profile is done using Viterbi algorithm. Compared to previous pair-wise methods such as DTW, CPM alignment is more robust since it aligns multiple LC-MS data simultaneously.

Since untargeted metabolomic experiments often produce a large number of runs, all of which need to be aligned as correctly as possible, most of the recent advances in warping-based methods are based on aligning peaks — a reduced representation of the raw LC-MS data obtained as the outcome of the peak detection step. Operating on peaks makes it easier to incorporate mass, intensities and other structural information that can potentially help improve the alignment result. By extracting a smaller set of features from complex LC-MS raw data, often it is easier and faster to align many runs at once. To deal with the non-linear nature of retention time shifts in LC-MS data, a approach is to attempt to fit a regression curve on the peaks — usually using all the features observed across run or by selecting a certain subsets of all peaks. Some examples of well-known warping-based methods that operate on peaks are highlighted below:

1. **XCMS** [30] XCMS is one of the oldest tool used in metabolomics for processing mass spectrometry data and metabolite profiling. Alignment is XCMS is performed in two stages: peak matching and retention time correction. During the peak matching stage, the m/z axis is divided into discrete fixed-width overlapping bins. The alignment algorithm constructs a Gaussian kernel density estimation of the peaks inside each bin. This results in groups of peaks ('meta-peaks') that are close in their masses. Groups that do not contain enough peaks across samples are discarded. Next, during the retention time correction stage, well-behaved groups are selected as landmark peaks. The median retention time of each group is calculated, and the deviation from the median for each peak is used to train a local regression model. The resulting regression is used to correct for peak deviations.

2. **OpenMS** [31] OpenMS alignment works by first selecting a replicate that has the highest number of features. This replicate is used as the reference replicate, against which all other replicates are aligned against (in a star-like manner). The actual alignment process is divided into following two phases: superposition and consensus. During the superposition phase, the alignment algorithm tries to find the parameter for an affine transformation that maximises the number of features mapped from the reference replicate to the other replicates. An object recognition algorithm, called pose clustering, is used for this purpose. Additional information – such as m/z, RT and intensity dimension – is considered during the clustering process. The subsequent consensus phase

then produces the actual alignment between matching features across replicates, using nearest-neighbour criteria.

3. **MZmine's RANSAC Aligner** [19] The RANSAC aligner is an alignment method developed part of the MZmine 2 software suite, used for the processing of metabolomics data. Random Sample Consensus (RANSAC) works by constructing a local regression model that maps retention time from one replicate to another. Once retention time correction has been performed, the actual matching of peaks across runs are performed greedily (using the older Join Aligner in MZmine 2). RANSAC Aligner is an iterative, non-deterministic algorithm, so there can be variations in the final alignment results. This non-determinism comes from the random sampling in the construction of the candidate model using the RANSAC algorithm[32].

Direct-matching Alignment Methods

Direct matching methods, which skip the warping step and seek to establish the correspondence of peaks across runs directly, can be preferred due to their simplicity, while still offering good performance [33]. Most direct matching methods consist of two stages: computing feature similarity and using this similarity to match peaks across runs. A wide range of feature similarity measures have been proposed to compare the m/z and RT values of two peaks, including normalised weighted absolute difference [19], cosine similarity [34], Euclidean distance [35], and Mahalanobis distance [36]. Once similarity has been computed, feature matching can be established through either a greedy or combinatorial matching method. Direct matching approaches therefore require that the peak detection step has already been completed, and the correctness of aligned peaksets depend on the output of the peak detection step. In fact, *all* steps that operate on peaks are similarly dependent on the correctness of the preceding peak detection step. In the presence of chemical and technical noises in the raw LC-MS data, relying on detected peak might serve to provide informative features rather than operating on the entire profile data [14].

Many approaches have been proposed for direct matching of peaks. Greedy direct-matching methods work by making a locally optimal choice at each step, in the hope that this will lead to an acceptable matching solution in the end. RTAlign in MSFACTs [37] merges all runs and greedily groups features into aligned peaksets within a user-defined RT tolerance. Join Aligner [19] in MZmine 2 merges successive runs to a master peaklist by matching features greedily according to their similarity scores within user-defined m/z and RT windows. Similarly, MassUntangler [35] performs nearest-distance matching of features, followed by various intermediate filtering and conflict-resolutions steps. Recent advances in direct matching methods have also posed the matching task as a combinatorial optimisation problem. Simultaneous Multiple Alignment (SIMA) [36] uses the Gale-Shapley algorithm to find a stable

matching in the bipartite graph produced by joining peaks (nodes) from one run with peaks from another run that are within certain m/z and RT tolerances. [38] explores the application of the classical Hungarian algorithm to find the maximum weighted bipartite matching. BIPACE [34] establishes correspondence by finding the maximal cliques in the graph. SMFM [39] uses dynamic programming to compute a maximum bipartite matching under a relaxed bijective mapping assumption for time mapping.

As the output of direct-matching methods is the list of aligned peaksets itself, this class of methods can be used as an independent alignment method or as a second-stage process that follows a warping-based method. Once RT drift have been corrected in warping-based methods, it is often easier to establish the actual correspondence of peaks. Seen differently, if a good correspondence between peaks can be established, finding a warping function that maps the retention time from one run to another also becomes easier. In this manner, both approaches to alignment — whether warping-based or direct-matching — complements each other. It is worth noting, however, that the final goal of alignment is not correcting retention time but establishing the matching of correspondent peaks across runs. In this manner, direct-matching methods directly addresses the core of the alignment problem, i.e. establishing the correspondence between peaks from different LC-MS runs rather than correcting for retention time.

Direct-matching methods can also be categorised depending on whether they require a user-defined reference run to be specified. When such reference is necessary, the full alignment of multiple runs is constructed through successive merging of pairwise runs towards the reference run (e.g. MZmine2's Join aligner in [19]. Alternatively, methods that do not require a reference run can either operate in a hierarchical fashion – where the final multiple alignment results are constructed in a greedy manner by merging of successive pairwise results following a guide tree (e.g. SIMA [36]) – or by pooling features across runs and grouping similar peaks in the combined input simultaneously (e.g. the *group()* function of XCMS in [30]).

Limitations of Current Alignment Methods

Several limitations exist in the alignment methods surveyed. In particular, many warping-based methods make the implicit assumption that the elution order of peaks are preserved across runs. This is not always the case as peaks are known to produce RT values that shift in a non-linear manner across runs [23], resulting in different elution orders of correspondent peaks in different runs. Warping-based methods, which operates on the profile data (i.e. the total ion chromatograms or the extracted ion chromatograms, see Figure 2.3), distorts the signal which may be undesirable when computing peak intensities later on. Direct-matching methods do not make such a strong assumption on the elution order of peaks, but it is often

assumed that a peak in one run can always be matched to another peak from a different run. In practice, a single peak from one run can have several potential matching peaks in another run, while having no matches in another run. It has been suggested that peaks are related due to being generated from the same compound [40], but this fact is not exploited during most direct-matching methods that performs matching on the basis of individual peak features alone. Many methods also require for a reference LC-MS run to be defined, and for the alignment of the remaining runs to be performed successively with respect to that reference run. To our knowledge, no studies have been done on the effect of the selection of this reference run or the effect of changing alignment orders.

Another challenge of alignment lies in the lack of comparative evaluations and benchmarking of the resulting alignment. Evaluation of alignment quality through manual visual inspection of superimposed profile images and some selected chromatograms is problematic and is not a systematic approach towards performance evaluation. While straightforward, the visual inspection of alignment quality is tedious and do not work for evaluation of a large number of aligned peaksets produced by the alignment of a large number of samples. It is also often subjective and might suffer from dissimilar interpretations across different experiments and datasets. In recent years, a number of works [33, 35, 36, 1] have been proposed that use precision and recall, two widely used measures in information retrieval, to evaluate alignment performance. In general, performance evaluation of alignment methods is difficult due to the lack of gold standard and evaluation criteria for benchmarking [21, 41]. Relatively few works, such as [33], exists that provide a comprehensive ground truth for evaluation. In fact, despite the many alignment methods that exist, most methods remain unevaluated, evaluated against a small number of alternatives or evaluated based on highly subjective criteria [42]. For instance, 48 alignment methods developed from 2001 to 2012 were surveyed in [41], and a majority (60%) were found to include no comparative evaluations to other methods at all. The lack of comparative evaluations make it difficult for the end-user to select which alignment method to use.

2.3.3 Gap Filling & Noise Filtering

From alignment, certain peaks might be missing in an aligned peakset. The gap filling step aims to recover this missing signal from the raw data. A peak may be missing as it was not detected in the peak detection step (due to having a low intensity or a poor chromatographic peak shapes). As another possibility, gap filling can also be performed before alignment, following the peak detection step, to recover lost signals [43]. Once gap filling is done, noise filtering is performed. Filtering can be performed based many criteria, e.g. using a threshold on the intensity to remove low-intensity peaks that are likely to be noise.

2.3.4 Peak Grouping

In the peak grouping stage, the sets of peaks that are chemically-related to each other are grouped. During ionisation in mass spectrometry, a single metabolite alone can produce multiple peaks (e.g. isotopic peaks, adduct peaks and fragment peaks) that are all chemically-related to each other. Following [2], we call this set of peaks the *ionisation product* (IP) peaks of the compound. In particular, the presence of naturally occurring isotopes (e.g. ^{13}C) means a single compound can produce a pattern of peaks with m/z and intensity that follow the isotopic distributions of the atomic elements of the compound [44]. Similarly, the formation of adducts (the addition of a molecule ion to another) means that within a mass spectrum, certain adduct peaks, generated from the same compound, can be explained by the set of adduct transformations [45]. As they co-elute from the column, these IP peaks are expected to have similar chromatographic peak shapes, and therefore they share similar RT values. In [46], an analogous concept of 'derivative peaks' is defined to be the set of peaks that elute at the same retention time, show a strong correlation between their chromatographic peak shapes, have mass differences that can be explained by known chemical relationships and have intensity values that can be correlated across different runs. The set of peaks produced from the same compound can therefore be grouped into a cluster of related peaks.

As discussed in Section 2.3.2, grouping information is also generally not used during alignment. In identification, peak groups can be used as a data filtering procedure, although whether the grouping information is used might vary from one pipeline to another. In particular, assuming that each observed peak corresponds to a single compound can produce many matches, with potentially a large number of false positives, when querying for matching mass in large public compound databases (such as KEGG or PubChem). As such, a filtering procedure can be used to reduce the number of possible matches. CAMERA [47] performs the annotations of ionisation product species on groups of peaks, based on constructing a similarity graph and detecting highly-connected subgraphs in the graph. IP peaks are annotated on the subgraphs based on how their masses can be explained by a set of user-defined chemical rules. In [2], IP peaks are grouped along the RT dimension using a sliding window and along the m/z dimension using k-means clustering. The grouping induced by these methods are used as a form of data filtering by discarding peaks that are deemed irrelevant. Following the idea of derivative peaks in [46], the mzMatch software suite [48] detects IP peaks based on a greedy clustering scheme. Peaks having the largest intensity are clustered to others sharing chromatographic peak shape correlations above a certain user-defined threshold. This is repeated until all peaks are processed. In [49], the same idea is exploited in the form of a mixture model to cluster peaks based on their chromatographic peak shape correlations.

2.3.5 Peak Identification

In a general sense, peak identification refers to the process of annotating a label that tells us which peaks are associated to which metabolite. As shown in Figure 2.4, the output from the identification step is a matrix where each row in the matrix corresponds to a biological or technical sample, each column a metabolite, and entries in the matrix are the intensity of the detected metabolite in each sample. Untargeted identification is challenging in untargeted metabolomic studies due to the vast number of metabolites present in sample and the diversity in elements that comprise a metabolite. Unlike the genome that has four nucleotide bases as its sole alphabets, or proteins with twenty one amino acids as their building blocks, metabolites are harder to characterise structurally, the basic building blocks of a metabolite are atoms (commonly CHNOPS) that can be arranged in a variety of configurations in a single molecule alone (Figure 2.1).

The term 'identification' can be overloaded with many different meanings, e.g. is it the definite annotation of a compound identity to a peak or is it the assignment of some putative labels to the peak? The Chemical Working Group of the Metabolomics Standards Initiative proposed four levels of identifications for the reporting of metabolite identifications [50] that have been accepted to a varying degree by the community. In this scheme, the most confident Level 1 identification is obtained through the comparison of the observed peaks against those generated from a set of chemical standards (a solution containing compounds of known concentration). A putative Level 2 identification is obtained from comparison against publicly available spectral libraries. Level 3 identification seeks to confirm the chemical class of the compounds, while a Level 4 of no identification is assigned unknown compounds. In its most basic form, both Level 1 and 2 identifications are performed by taking the neutral masses of observed peaks and matching them against the list of masses from a database of compounds, which may range in size from just a few hundreds of metabolites to as large as tens of thousands of compounds or more. The database for matching may be constructed for the standard compounds or the public database. Having a high mass accuracy is therefore crucial for identification as it reduces the size of possible alternatives that can be matched.

In untargeted metabolomics, the lack of knowledge in the composition of metabolites in the sample means that, apart from the small number of metabolites confidently identified as the authentic standard compounds, the putative annotations of metabolite identities that are assigned to a peak might be the result of incorrect matching against the compound database. This leads to false positive identifications and consequently incorrect biological conclusions. Creating a larger authentic standard to facilitate more confident identifications is constrained by time and cost and can never be comprehensive enough to include all metabolites of interest in an untargeted study. Another challenge of identification is even at the very high mass accuracy of 1 ppm, the number of possible formulae matched by accurate mass is still

too large to allow for definite metabolite identifications [51]. Identification is particularly difficult for metabolites present in low abundance in the samples. Relying on mass alone for untargeted identification is also problematic as different metabolites may produce peaks having the same measured m/z values, and as in the case of isomers, the same precursor mass can therefore be matched to multiple possible formulae. Retention time drift, a main challenge in alignment, means RT values vary across different chromatography platforms and laboratories and cannot be easily used as a characteristic identifying information in a public compound database. Incomplete knowledge on the metabolites expected to be present in the sample, coupled with the complexity of the sample being analysed itself, means identification is challenging [52], with more metabolites being putatively identified (Level 2) than very confidently identified (Level 1), but the majority of metabolites can only be identified based on their class (Level 3) or not at all (Level 4). Even for the putatively identified metabolites, their manual verification is a laborous and time-consuming process, often serving as the primary bottleneck in large-scale untargeted metabolomic studies [52, 53]. In particular, false positives from identification is a major concern in the data pre-processing step.

To reduce false positives, additional information can be incorporated into the identification process. In particular, identification can also be performed on the basis of a group of ionisation product peaks, rather than on individual peaks alone, although this is often not exploited in many tools. As discussed before, tools such as CAMERA [47] can produce a group of IP peaks. From this group, the precursor mass that corresponds to the molecular ion mass of the compound can be deduced. This can be used for matching against a compound database, allowing for a set of peaks to be identified rather than individual peaks alone. Other sources of information that can help identification include using the predicted RT of a compound [54, 15, 55], but matching the predicted RT values against the observed RT data that contain drifts might be challenging too. Probabilistic methods that use prior information of a known set of formulae to annotate peaks by explainable transformations have also been proposed [56, 57], but they often have difficulties scaling up to large-scale experiments for practical use.

Fragmentation through tandem MS or MS^n instruments is another way to provide further information to aid identification. As suggested by its name, tandem MS requires two MS analysers operating in tandem. Ions resulting from the initial fragmentation of metabolites in the first MS analyser are selected for further fragmentation in the second MS analyser. The ions selected for the first MS analyser stage are called the precursor ions. In data-dependent acquisition (DDA), precursor ions within some small m/z windows are selected based on some predetermined rules (such as fragmenting the top few most intense precursor peaks in each scan). As a result, typically a small percentage, e.g. less than a fifth of all precursor peaks in the full-scan mode data are selected for MS-MS fragmentation. Peaks that are generated from the fragmentation of the precursor ions in the second MS stage are

called product ions. Fragmentation spectra of product ions are often used as the unique 'fingerprint' identifiers of the structural composition of the precursor ions. An alternative to DDA is the data-independent acquisition (DIA), where no selection of precursor ions needs to be specified as all peaks within a defined m/z range are fragmented. DIA results in a more complex fragmentation spectra due to multiple metabolites being fragmented together in the same m/z window, and require sophisticated analysis strategy to deconvolve the signals from the noise.

A fragmentation spectrum of interest can be identified through matching against **(1)** a database of public reference spectra or **(2)** a database of theoretical spectra generated in an in-silico manner [58]. Examples of public databases are KEGG [59], Massbank [60] and ChemSpider [61]. Frequently, a combination of matching against a public database and in-silico theoretical spectra is used to ensure the largest coverage of compounds during matching. The actual matching process is often established in a greedy manner, heuristically through agreement against a set of well-validated fragmentation rules or combinatorially by minimising a cost/distance function. In the combinatorial case, heuristic rules are still applied to reduce the exponentially-growing search space to allow matching to run in acceptable time. However, fragmentation cannot be used in all cases as not all metabolomics experiments include fragmentation as part of their data acquisition process — due to cost or other resource constraints. Publicly available databases have a limited coverage in the number of submitted spectra. Often spectra in public databases are contributed from a wide variety of instruments, further limiting potential matches as matching is often possible only for spectra generated on similar platforms. Large variance in the mass accuracy and characteristics of submitted spectral library entries further limit potential matches as a query match can only be made against spectra generated from similar platforms and mass accuracies. Unwanted spectral peaks (due to e.g. the presence of contaminants and noise in the sample) present in the database may also lead to incorrect spectral matching. Fragmentation and its challenges are further discussed in Chapter 7.

2.3.6 Analysis

The last step in preprocessing of LC-MS data is the normalisation and visualisation of data. Normalisation is essential for removing any possible variation and systematic bias to allow for comparisons of differential levels of expressions of metabolites across samples. Statistical analysis is performed with visualizations in order to draw useful inferences from data – a step that is crucial in confirming or rejecting biological hypotheses. At this stage, the data is normalised to correct for systematic variations before statistical analysis. Spiked-in compounds that do not occur naturally are used for this purpose. Since the spiked-in compounds are expected to have equal concentration in all samples, they can be used to normalise

peak areas in samples. Statistical analysis, such as t-test, ANOVA and principal component analysis, can then be performed on the normalised peaks across samples. The goal of statistical analysis is to answer biological hypothesis posed by life-science researchers. During the analysis, it is common to place the result obtained from metabolomic studies on the larger biological context by mapping them onto some biological pathways ([62, 63]) or in relation to other -omics studies ([64, 65]).

While targeted metabolomics focuses on a handful of specific metabolites, untargeted studies (such as in [66] and [54]) attempt to perform a global analysis of metabolites in the samples under study. Understanding the metabolome in an untargeted study is a challenging task due to the complex interactions of metabolites in the metabolome. Identification of specific metabolites are frequently not the final goal in untargeted metabolomics, rather it is the discovery of metabolites or groups of metabolites that are differentially expressed or correlated to the expression of specific physical traits being studied. Of particular interest is the detection of metabolites that act as disease biomarkers. The presence or absence of such metabolites can provide an indication to the corresponding presence or absence of disease in the organism [67]. Differences caused by genetic variations are also highly visible as changes in the metabolite composition of an organism. These could be quantified through differential analysis that compares the expression levels (abudance) of metabolites across samples. The resulting differential analysis provides biologists with a better understanding of the metabolic pathways in the cell and how they respond to perturbations. Differential analysis also underpins many practical applications of systems biology, such as nutritional research [68], drug discovery [69] and even in an integrative approach that combines genomics and metabolomics to obtain a more comprehensive picture of living organisms [65]. Visualisation of the identified metabolites can also be performed by mapping metabolites to well-known pathways from databases such as KEGG [70] or MetaCyc [71]. Identified metabolites at this stage can also be integrated with the reconstructed metabolic information from other -omics [72] to allow for a rapid generation of biological hypotheses.

2.3.7 Mass Spectrometry Analysis in Proteomics

LC-MS analysis in proteomics proceeds largely in the same manner as to the data preprocessing pipeline in Figure 2.4. However, the key difference between proteomics and metabolomics lies in sample preparation. In the mass spectrometry analysis of proteins, the samples to be analysed come either in the form of tissues or as body fluids, such as urine, plasma and serum, with each different type of sample demand an appropriate sample handling protocol. Next, cells extracted from the sample are broken down, allowing proteins to be isolated from other constituent parts of the cell, for instance the DNA, lipids and other metabolites that are present. The purified proteins are then separated. Traditional 2-D

gel electrophoresis method allows proteins to be separated according to their size (molecular mass) in one axis and according to their isoelectric points (the pH where the molecule carries no electrical charges) on another. Because 2D-GE approach is tedious and time-consuming, liquid chromatograph mass spectrometry has gotten more popular as the preferred separation technology as it enables the large-scale high-throughput separation of thousands of proteins in a single chromatographic run. Enzymes that can cut the peptide bonds, such as trypsin, are then used to digest proteins into shorter peptide fragments. Using certain enzymes, the cleavage of the peptide bonds happen at specific and predictable spots, allowing well-defined and easily identifiable peptide fragments to emerge. For instance by using trypsin as the digestion enzyme, the cleavage of the protein happens after each arginine or lysine amino acid is encountered, unless a proline amino acid comes next.

Identification of peptide sequences in proteomics largely proceeds in the same manner as metabolomics. Different set of tools and public databases are queried for matching. In particular, the problem of peptide identification from fragmentation data is referred to as peptide mass fingerprinting [73]. As proteins are cleaved into peptides that are unique, the resulting fragmentation spectra are also expected to be unique to a protein. The theoretical peptide spectra can then matched against a reference spectra library. In practice, the resulting fragmentation spectra are not entirely unique and multiple hits can be returned from the spectra library, particularly in the case of libraries that have a large number of records. The fact that the peptide sequence of a protein is known and digestion enzyme produces cuts at predictable spots means identification through a comparison to a *de novo* peptide sequences is possible in proteomics. Additionally, it is also more common in proteomics than metabolomics for an initial separation process, called pre-factionation, to be performed on the digested peptides using liquid chromatography. This divides the entire sample into multiple *fractions* of compounds that elute at different retention time, which can then be ran separately through the LC-MS instrument for mass fragmentation analysis in a manner similar to metabolomics analysis. Certain fractions can be selected for further analysis, leading to a simpler set of data to deal with.

2.4 Conclusion

Data processing has major impact on the outcome of quantitative label-free LC-MS analysis [74]. Even the choice of the software tools itself, with differing implementation details, affect the outcome. In particular, label-free experiments pose many challenges when analysing many LC-MS runs. Since large-scale untargeted metabolomics study can generate a huge number of samples (see [66, 54]), having a reliable and accurate peak alignment step during data pre-processing is important. Peaks that are improperly aligned can lead to false

positives, and especially for untargeted label-free metabolomic experiments, the presence of even relatively small errors in any steps preceding the identification stage (including alignment) can result in significant differences to the final analysis and biological conclusions. Errors or uncertainties inadvertently produced in any sub-step before identification would be carried forward forward in the pipeline. Improper pre-processing steps can also introduce variabilities that obscure important biological variations of metabolites themselves.

Software tools that deals with LC-MS data in proteomics and metabolomics usually operate in a modular and serial manner, where successive transformations occur to the raw LC-MS data as it goes through the data pre-processing pipeline. However, it is important to note that despite the apparently serial pre-processing manner shown in Figure 2.4, the actual pipeline workflow employed by the user is often iterative. For example, it is often the case that certain low intensity metabolites are expected to be present in the identification result, but are found to be missing. This requires the user to revisit each step of the pipeline, experiment with the numerous user-defined parameters and threshold values used for the peak detection, alignment, gap filling, noise filtering and identification step to troubleshoot this issue. Each step of the exemplar pipeline in Figure 2.4 is therefore dependent on the steps that come before it. However, at the moment, each step in the pipeline exists independently and information from one step is not used to improve the performance of the subsequent steps in the pipeline.

This chapter has provided the necessary background knowledge to understand the basic principles of mass-spectrometry-based analysis as applied to large-scale untargeted biological studies, but it is far from complete. A particular emphasis is given to the application of mass spectrometry techniques in the field of metabolomics. For further readings on mass spectrometry as an analytical platform, the reader is directed to more comprehensive textbooks such as [75] and [76]. For literature surveys on the different steps that comprise an LC-MS data processing pipeline, the reader is directed to [21, 14, 77, 8] for metabolomics and [73, 78, 14] for proteomics.

Chapter 3

Probabilistic Modelling

3.1 Introduction

As described in Chapter 2, the raw data produced from liquid chromatography mass spectrometry (LC-MS) measurements has to be processed through a data pre-processing pipeline before further analysis. From the peak detection step, we obtain points on the ion chromatograms having mass-to-charge (m/z), retention time (RT) and intensity values. We call each point a *peak*. The nature of LC-MS measurements means that a compound being analysed generates multiple peaks. At the heart of this thesis is the grouping of these peaks that are structurally or chemically related, and using the grouping to improve other steps (such as the alignment and identification steps) in the pipeline. The problem of finding these groups of related peaks can be approached as an unsupervised learning problem. In the unsupervised learning approach, broadly speaking our task is to separate peaks into *clusters*, where members of the cluster are related through sharing some commonalities, e.g. from being the ionisation products of the same compound or from sharing chemical substructures.

Numerous methods exist to perform data clustering in an unsupervised manner [79, 80]. In probabilistic modelling, one way to do this is to try and explain the generative process that produces the observed data. This results in a generative model. Peaks generated from the same underlying cause in the model can then be assigned to the same cluster. Modelling the data in this manner has some advantages in comparison to other distance-based clustering methods, such as e.g. hierarchical clustering that has also been applied to peak data [81, 82]. A generative model provides more than just clustering. It is often easier to extract from a generative model a hint as to *why* the observed data points are clustered, and this insight can be very useful in certain applications. Additionally, through specifying the appropriate likelihood functions, generative modelling also provides a flexible way of specifying how data points should be clustered, while prior assumptions can be incorporated into the model in a principled manner.

Generative modelling has been applied to LC-MS data. In [49], mixture model clustering is used to cluster LC-MS peaks in the same run by their chromatographic profiles. Mixture models are the building blocks of more complex generative models. In mixture models, it is assumed that the observed data can be explained by the presence of some latent variables. These variables are 'latent' as they are not directly observed, rather their presence is inferred from the observed data. The assumption made in [49] is that ionisation product peaks that are related share similar chromatographic profiles. Given N peaks in the data, the method computes the pairwise Pearson correlation values for the chromatographic profiles of all peaks, resulting in an N-by-N matrix of Pearson correlation values. The likelihood of an entry in this matrix is described by a mixture of two components: an exponential-type distribution to describe the correlation values of peaks in the same cluster and a Gaussian distribution to describe the correlation values of peaks in different clusters.

Along a similar line, mixture model clustering is also used in MetAssign [57] to perform the probabilistic annotations of ionisation product types and formulae to peak data. It is assumed in MetAssign that a prior knowledge of the form of known formulae is provided. Theoretical peaks are then generated using the provided formulae. The likelihood of an observed peak is computed based on how well the observed m/z, RT and intensity values fit the theoretical peaks. Other applications of probabilistic modelling on mass spectrometry data include modelling the assignment of formulae to peaks [56, 83], modelling the fragmentation events of tandem mass spectrometry data, where the separation is performed using liquid chromatography (CMF-ESI, [84]) or gas chromatography (CFM-EI, [85]). Machine learning techniques in general have also been applied to mass spectrometry data, e.g. for the predictions of retention time [54, 15, 55] and the characteristic fingerprints of compounds from fragmentation data [86, 87].

3.2 Mixture Model Clustering

As an example of the thinking process behind generative modelling, we see that during liquid chromatography, metabolites are separated by their chemical properties. From mass spectrometry, ionisation product peaks are produced from the same metabolites. These peaks will co-elute and have similar chromatographic profiles, including broadly similar RT values. A group of observed peaks having similar retention time (RT) values can be therefore be modelled as being generated by the same metabolite, and in this case, although the metabolite is not directly observed, its presence can be inferred based on the observed data. Peaks that are related to the same compound can therefore be clustered according to their RT values. Let our LC-MS run be represented as $\mathbf{y} = \{y_1, y_2, ..., y_n\}$ where each y_n is the RT value of a peak.

A principled way to model a generative process is through Bayesian inference. Suppose θ is the parameter of interest to the generative process that produces the data $\mathbf{y}$. In Bayesian inference, we begin by specifying a prior distribution over the model parameter θ. Through the application of Bayes rule, this prior distribution is updated by the likelihood of seeing the observed data given our prior hypothesis on θ, resulting in a posterior distribution:

$$p(\theta|\mathbf{y}) = \frac{p(\mathbf{y},\theta)}{p(\mathbf{y})} = \frac{p(\mathbf{y}|\theta)p(\theta)}{p(\mathbf{y})} = \frac{p(\mathbf{y}|\theta)p(\theta)}{\int_\theta p(\mathbf{y}|\theta)p(\theta)\ d\theta} \tag{3.1}$$

In eq. (3.1), $p(\mathbf{y},\theta)$ is the joint distribution between the data $\mathbf{y}$ and the model parameter θ. This can be factorised into a product of $p(\mathbf{y}|\theta)$, which is the likelihood of observing the data $\mathbf{y}$ given the model parameter θ, and $p(\theta)$, which is the prior distribution on the model parameter θ. Normalising the joint distribution by the marginal likelihood or evidence $p(\mathbf{y})$ produces the posterior distribution $p(\theta|\mathbf{y})$, which is the probability of model parameter θ given the data. Inferring model parameters given the observed data is usually what we are interested in.

Using the posterior distribution, we can make a prediction on a new peak, y_{new} by averaging over all values of θ. This results in the posterior predictive distribution:

$$p(y_{new}|\mathbf{y}) = \int_\theta p(y_{new}|\theta)p(\theta|\mathbf{y})\ d\theta \tag{3.2}$$

In many cases, the integral in eqs. (3.1) and (3.2) cannot be solved analytically and have to be approximated through maximum likelihood or sampling-based approaches.

We now introduce mixture modelling for this example peak data. Probabilistic mixture model represents each cluster by a probability distribution, with a distribution being a component in the mixture model. Our resulting Gaussian mixture model for the peak data follows from [88]: it starts from having a finite number of components (denoted by K) and is later extended in Section 3.3 to an infinite mixture model, where the number of components is unbounded. The generative process for this finite mixture model can be written as the following. The conditional dependencies of random variables in the finite mixture model is also shown in Figure 3.1A.

$$\begin{aligned} \boldsymbol{\pi}|\boldsymbol{\alpha} &\sim Dir(\boldsymbol{\alpha}) \\ z_{nk} = 1|\pi_k &\sim \boldsymbol{\pi} \\ \mu_k|\mu_0 &\sim \mathcal{N}(\mu_k|\mu_0,\sigma_0^2) \\ y_n|z_{nk}=1,\mu_k &\sim \mathcal{N}(y_n|\mu,\sigma^2) \end{aligned} \tag{3.3}$$

We now explain the model specification in eq. (3.3). First we assume that peaks that are related are generated by the same component in the mixture model. Let the variable $k = 1, ..., K$ index the mixture components. The choice of which probability distribution to

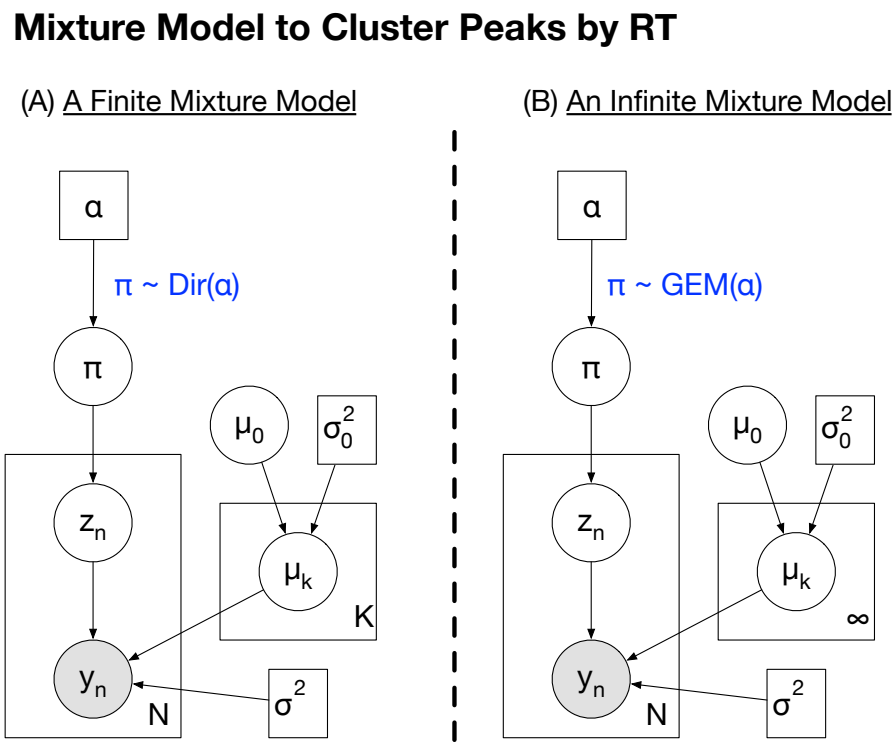


Figure 3.1: Graphical models of (1) a finite mixture model, which is extended into (2) an infinite mixture model, to cluster peaks by their retention time (RT) values. Circles denotes random variables, squares denote fixed parameters, while the shaded node denotes an observed peak's RT.

use as a component is usually determined by the type of observed data. Each observed data point y_n can be considered to a random variable drawn from the generating probability distribution. Assuming that each data point is generated by a univariate Gaussian distribution, we denote by $y_n|\mu, \sigma^2 \sim \mathcal{N}(\mu, \sigma^2)$ that y_n is distributed as a Gaussian distribution having the mean μ and the variance σ^2 (as an alternative parameterisation, precision, i.e. the inverse variance ($\frac{1}{\sigma^2}$) can also be used, with a higher precision meaning a narrower distribution). The probability density function for this univariate Gaussian distribution is given by:

$$\mathcal{N}(y_n|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(y_n-\mu)^2} \tag{3.4}$$

For a single peak, its mixture model likelihood is therefore given by:

$$p(y_n|\boldsymbol{\mu}, \boldsymbol{\pi}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(y_n|\mu_k, \sigma^2) \tag{3.5}$$

where π_k is the mixture proportion (the positive weight for each component) and μ_k is mean for that component. $\boldsymbol{\pi} = \{\pi_1, ..., \pi_k\}$ denotes the vector over all mixture proportions that sums to one ($\sum_{k=1}^{K} \pi_k = 1$).

In this model, each mixture component is set to have an unknown mean μ_k but a known variance σ^2. The choice of setting an unknown μ_k but a fixed variance for σ^2 is motivated by the following reasonable modelling assumptions: **(1)** the retention time drift of observed peaks is broadly similar across the compounds being measured, and **(2)** this parameter can be set by the user based on his knowledge on the characteristic RT drifts of the LC instrument. Each cluster mean μ_k is assumed to be generated independently by a prior Gaussian distribution, parameterised by the mean μ_0 and the variance σ_0^2. Let $\boldsymbol{\mu} = \{\mu_1, ..., \mu_k\}$ be the vector over all component means. This results in:

$$p(\boldsymbol{\mu}|\mu_0, \sigma_0^2) = \prod_{k=1}^{K} \mathcal{N}(\mu_k|\mu_0, \sigma_0^2) \tag{3.6}$$

We also require another random variable z_{nk} to store the assignment of peak n to cluster k, i.e. $z_{nk} = 1$ if peak n is assigned to cluster k and 0 otherwise. Each peak is assumed to be generated independently by exactly one mixture component ($\sum_k z_{nk} = 1$). For a peak, its entire cluster assignments can be stored in a vector $\boldsymbol{z}_n$ of length K, where only k-th entry has a value of 1 (at $z_{nk} = 1$). $\boldsymbol{z}_n$ is assumed to be generated from a multinomial distribution having the parameter vector $\boldsymbol{\pi}$. This multinomial distribution has the probability mass function given by:

$$p(\boldsymbol{z}_n|\boldsymbol{\pi}) = C \prod_{k=1}^{K} \pi_k^{z_{nk}} \tag{3.7}$$

where C is the multinomial coefficient, given by $\frac{(\sum_k z_{nk})!}{\prod_{k=1}^{K} z_{nk}!}$. Since $\boldsymbol{z}_n$ has only one draw from the multinomial, C evaluates to 1 and can be dropped. Now, let $\boldsymbol{Z}$ be the set of all indicator vectors for all peaks. This results in the following likelihood for all the peak assignment vectors:

$$\begin{aligned} p(\boldsymbol{Z}|\boldsymbol{\pi}) &= \prod_{n=1}^{N} \prod_{k=1}^{K} \pi_k^{z_{nk}} \\ &= \prod_{k=1}^{K} \pi_k^{c_k} \end{aligned} \tag{3.8}$$

where $c_k = \sum_n z_{nk}$ is the count of peaks assigned to the k-th cluster. Collectively for all peaks, the joint likelihood of the observed data and the cluster assignments is:

$$p(\boldsymbol{y}, \boldsymbol{Z}|\boldsymbol{\mu}, \boldsymbol{\pi}) = \prod_{n=1}^{N} \prod_{k=1}^{K} [\pi_k \ \mathcal{N}(y_n|\mu_k, \sigma^2)]^{\boldsymbol{z}_{nk}} \tag{3.9}$$

In our generative model, a prior distribution is also placed on $\boldsymbol{\pi}$. Due to its conjugacy to the multinomial distribution, a Dirichlet distribution parameterised by the vector $\boldsymbol{\alpha} =$

$[\alpha_1, \alpha_2, ..., \alpha_k]^T$ is a suitable prior. This results in:

$$\begin{aligned} p(\boldsymbol{\pi}|\boldsymbol{\alpha}) &= \frac{\Gamma(\sum_k \alpha_k)}{\prod_k \Gamma(\alpha_k)} \prod_{k=1}^{K} \pi_k^{\alpha_k - 1} \\ &\propto \prod_{k=1}^{K} \pi_k^{\alpha_k - 1} \end{aligned} \tag{3.10}$$

We can now state the complete joint likelihood of the model. Putting together the individual terms in eqs. (3.6)-3.10) and their respective independence assumptions, we obtain the joint probability distribution of the model parameters and data $p(\boldsymbol{y}, \boldsymbol{Z}, \boldsymbol{\mu}, \boldsymbol{\pi}|\boldsymbol{\alpha}, \mu_0, \sigma_0^2)$, which can be factorised into:

$$p(\boldsymbol{y}, \boldsymbol{Z}, \boldsymbol{\mu}, \boldsymbol{\pi}|\boldsymbol{\alpha}, \mu_0, \sigma_0^2) = p(\boldsymbol{y}|\boldsymbol{Z}, \boldsymbol{\mu}, \boldsymbol{\pi}) p(\boldsymbol{Z}|\boldsymbol{\pi}) p(\boldsymbol{\pi}|\boldsymbol{\alpha}) p(\boldsymbol{\mu}|\mu_0, \sigma_0^2) \tag{3.11}$$

3.2.1 Gibbs Sampling for a Finite Mixture Model

Given the joint distribution in eq. (3.11), we are interested to infer the posterior distribution on the assignments $\boldsymbol{Z}$, the mixture proportions $\boldsymbol{\pi}$ and the cluster means $\boldsymbol{\mu}$. This is given by:

$$p(\boldsymbol{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}|\boldsymbol{y}, \boldsymbol{\alpha}, \mu_0, \sigma_0^2) = \frac{p(\boldsymbol{y}, \boldsymbol{Z}, \boldsymbol{\mu}, \boldsymbol{\pi}|\boldsymbol{\alpha}, \mu_0, \sigma_0^2)}{p(\boldsymbol{y}|\alpha, \mu_0, \sigma_0^2)} \tag{3.12}$$

Substituting eq. (3.11) into the numerator of eq. (3.12) results in the following posterior distribution over the parameters that we want to infer:

$$p(\boldsymbol{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}|\boldsymbol{y}, \alpha, \mu_0, s_0) \propto p(\boldsymbol{y}|\boldsymbol{Z}, \boldsymbol{\mu}, \boldsymbol{\pi}) p(\boldsymbol{Z}|\boldsymbol{\pi}) p(\boldsymbol{\pi}|\boldsymbol{\alpha}) p(\boldsymbol{\mu}|\mu_0, s_0) \tag{3.13}$$

In many cases for the more interesting and complex models, the posterior distribution (such as the one in eq. 3.13) and also the posterior predictive distribution cannot be derived analytically. Various methods, such as the EM algorithm [89], can be used to perform posterior inference in a mixture model, but throughout this thesis, we will use Gibbs sampling, an instance of Markov chain Monte Carlo (MCMC) methods. Gibbs sampling approximates the target posterior distribution by sequentially updating each random variable conditioned on all other random variables in the model. This requires deriving the *conditional distribution* of each random variable that we want to infer. In some cases, obtaining these conditional distributions can be challenging, although the process can be simplified by the independence assumptions of our model (e.g. in assuming that the cluster means are independent) and through the use of the appropriate conjugate prior distributions. Here we describe the steps required to construct a Gibbs sampler for the mixture model defined in eq. (3.3).

As the initial step in our Gibbs sampler, we initialise the cluster means $\mu_1, \mu_2, .., \mu_k$ and the mixture proportion $\boldsymbol{\pi}$ by sampling from their respective prior distributions. Then we sequentially sample for new values of $\boldsymbol{Z}$, $\boldsymbol{\mu}$ and $\boldsymbol{\pi}$ from the conditional distributions listed below.

1. We can update $\boldsymbol{z}_n$, the membership vector for peak n, by updating each of its k-th individual entry, i.e. z_{nk}. Simplifying eq. (3.9) to consider just one n-th peak, we obtain the following after normalisation:

$$P(z_{nk} = 1|\boldsymbol{\pi}, y_n, \mu_k) = \frac{\pi_k \ \mathcal{N}(y_n|\mu_k, \sigma^2)}{\sum_{k=1}^{K} \pi_k \ \mathcal{N}(y_n|\mu_k, \sigma_2)} \tag{3.14}$$

2. As the next step, we also need to update each μ_k conditioned on the membership vectors $\boldsymbol{Z}$ and the hyperparameters μ_0 and σ_0. Consider one k-th cluster, and let $\boldsymbol{x}_k = \{x_1, x_2, ..., x_m\}$ be the set of peaks currently assigned to cluster k. The variable m indexes over the member peaks of cluster k, and there are M_k such peaks. Their joint likelihood is given by $p(\boldsymbol{x}_k|\mu_k)$. As defined in eq. (3.6), we assume that each μ_k is independent given its conjugate prior $\mathcal{N}(\mu_0, \sigma_0^2)$. The posterior distribution on $p(\mu_k|\boldsymbol{x}_k, \mu_0)$ is therefore:

$$\begin{aligned} p(\mu_k|\boldsymbol{x}_k, \mu_0) &\propto p(\boldsymbol{x}_k|\mu_k) \cdot p(\mu_k|\mu_0) \\ &= \prod_{m=1}^{M_k} \mathcal{N}(x_m|\mu_k, \sigma^2) \cdot \mathcal{N}(\mu_k|\mu_0, \sigma_0) \\ &= \prod_{m=1}^{M_k} \frac{1}{\sqrt{2\sigma^2\pi}} exp\left(\frac{-(x_n - \mu_k)^2}{2\sigma^2}\right) \cdot \frac{1}{\sqrt{2\sigma_0^2\pi}} exp\left(\frac{-(\mu_k - \mu_0)^2}{2\sigma_0^2}\right) \end{aligned} \tag{3.15}$$

Since $p(\mu_k|\boldsymbol{x}_k, \mu_0)$ is a product of Gaussians, the posterior is proportional to another Gaussian, parameterised by say $\mathcal{N}(\tilde{\mu}, \tilde{\sigma}^2)$. Equating this with eq. (3.15) results in:

$$exp\left(\frac{-(\mu_k - \tilde{\mu})^2}{2\tilde{\sigma}^2}\right) \propto exp\left(\frac{-\sum_{m=1}^{M}(x_m - \mu_k)^2}{2\sigma^2} + \frac{-(\mu_k - \mu_0)^2}{2\sigma_0^2}\right) \tag{3.16}$$

Simplifying eq. (3.16) and completing the squares, we obtain the following parameters for $\mathcal{N}(\mu_k|\tilde{\mu}, \tilde{\sigma}^2)$:

$$\tilde{\mu} = \tilde{\sigma}^2\left(\frac{\sum_{m=1}^{M} x_m}{\sigma^2} + \frac{\mu_0}{\sigma_0^2}\right), \ \ \tilde{\sigma}^2 = \frac{1}{\frac{M}{\sigma^2} + \frac{1}{\sigma_0^2}} \tag{3.17}$$

3. Finally we also need to update the mixture proportion $\boldsymbol{\pi}$. Putting together the multinomial likelihood and Dirichlet prior in eqs. (3.8) and (3.10), we obtain a conditional

distribution for $\boldsymbol{\pi}$ that is another Dirichlet distribution, parameterised by $[\alpha_1 + c_1, \alpha_2 + c_2, ..., \alpha_k + c_k]^T$. Each entry in this parameter vector is influenced by two values: the pseudo-count contribution from α_k and the actual counts of peaks currently assigned to cluster k from c_k.

$$\begin{aligned} p(\boldsymbol{\pi}|\boldsymbol{\alpha}, \boldsymbol{Z}) = p(\boldsymbol{Z}|\boldsymbol{\pi}) \cdot p(\boldsymbol{\pi}|\boldsymbol{\alpha}) &\propto \prod_{k=1}^{K} \pi_k^{\alpha_k - 1} \cdot \prod_{k=1}^{K} \pi_k^{c_k} \\ &= \prod_{k=1}^{K} \pi_k^{\alpha_k + c_k - 1} \\ &= Dir(\alpha_1 + c_1, \alpha_2 + c_2, ..., \alpha_k + c_k) \end{aligned} \tag{3.18}$$

In Gibbs sampling, each newly updated value is immediately used before sampling for the next value. This sampling of each random variable is repeated until convergence. Often a certain number of initial samples are discarded during the *burn-in* period. Since successive samples are correlated, a certain *thinning* interval is also used to reduce the number of samples used. The resulting samples can now be used to approximate the true posterior distribution of the model. Frequently, the marginal distribution of the random variable of interest is studied. Particularly for our case, often we are interested in the probability of any pair of peaks (or even a set of peaks) to be placed in the same component since, as the subsequent chapters will show, this has a direct application to the problem of peak alignment.

3.2.2 Collapsed Gibbs Sampling for a Finite Mixture Model

As we have chosen conjugate prior distributions on the mixture proportion $\boldsymbol{\pi}$ and also the cluster mean μ_k, it is possible for us to integrate (*collapse*) $\boldsymbol{\pi}$ and μ_k from the model during Gibbs sampling. This results in a collapsed Gibbs sampler (CGS) where we need not sample $\boldsymbol{\pi}$ and μ_k explicitly. Collapsing has also been shown to lead to a better model convergence [89]. It will also help in the next section when we want to extend our finite mixture model (where K the number of components is specified) to an infinite mixture model (where the number of components is unbounded) as we do not need to explicitly sample an infinite-dimensional vector $\boldsymbol{\pi}$.

Specifically in this CGS implementation, we aim to marginalise $\boldsymbol{\pi}$ and μ_k by integrating them out from the conditional probability for z_{nk}, the assignment of peak n to cluster k. Collapsing $\boldsymbol{\pi}$ introduces dependencies among all the $\boldsymbol{z}_n$ random variables, so we introduce another notation $\boldsymbol{Z}^-$ to mean all other $\boldsymbol{z}_n$s except the one for the current n-th peak being sampled upon. Similarly, $\boldsymbol{y}^-$ denotes the RT values for other peaks apart from y_n. The

conditional distribution for z_{nk} in the CGS is given by:

$$p(z_{nk} = 1, \boldsymbol{\pi} | \boldsymbol{Z}^{-}, \boldsymbol{y}, \mu_0, \boldsymbol{\alpha}) \propto p(y_n | \boldsymbol{Z}^{-}, \boldsymbol{y}^{-}, \mu_0) \cdot P(z_{nk} = 1 | \boldsymbol{Z}^{-}, \boldsymbol{\alpha}) \tag{3.19}$$

We consider both terms of eq. (3.19) separately.

1. The first term on the right hand side of eq. (3.19) is the likelihood of y_n to be assigned to cluster k. Here, we no longer need to sample for μ_k as we are integrating over all values of μ_k using the posterior distribution $\mathcal{N}(\mu_k | \tilde{\mu}, \tilde{\sigma}^2)$ defined in eq. (3.16). Instead we can directly compute this likelihood by:

$$\begin{aligned} p(y_n | \boldsymbol{Z}^{-}, \boldsymbol{y}^{-}, \mu_0) &\propto \int p(y_n | \mu_k) \cdot p(\mu_k | \boldsymbol{Z}^{-}, \boldsymbol{y}^{-}, \mu_0) \ d\mu_k \\ &\propto \int \mathcal{N}(y_n | \mu_k, \sigma^2) \cdot \mathcal{N}(\mu_k | \tilde{\mu}, \tilde{\sigma}^2) \ d\mu_k \\ &\propto \mathcal{N}(y_n | \tilde{\mu}, \sigma^2 + \tilde{\sigma}^2) \end{aligned} \tag{3.20}$$

   where $\tilde{\mu}$ and $\tilde{\sigma}$ are defined in eq. (3.17).

   As an alternative parameterisation, we can also rewrite $\tilde{\mu}$ and $\tilde{\sigma}^2$ using precision (inverse variance) $\tau = \frac{1}{\sigma^2}$ and $\tau_0 = \frac{1}{\sigma_0^2}$ to replace the variances. The expression in eq. (3.17) then becomes:

$$\begin{aligned} \tilde{\mu} &= \tilde{\sigma}^2 \left(\frac{\sum_{m=1}^{M} x_m}{\sigma^2} + \frac{\mu_0}{\sigma_0^2} \right) = \frac{\tau \sum_{m=1}^{M} x_m + \mu_0 \tau_0}{M\tau + \tau_0} \\ \tilde{\sigma} &= \frac{1}{\frac{M}{\sigma^2} + \frac{1}{\sigma_0^2}} = \frac{1}{M\tau + \tau_0} \end{aligned} \tag{3.21}$$

   In the Gibbs samplers for the mixture models in the later chapters, this parameterisation using precision is what we will use.

2. The second term on the right hand side of eq. (3.19) is the prior probability of assigning the n peak to cluster k. Again we do not have to sample for $\boldsymbol{\pi}$ as we integrate over all values of $\boldsymbol{\pi}$. Our desired conditional probability is given in eq. (3.22). By definition, $P(z_{nk} = 1 | \boldsymbol{\pi})$ is π while $p(\boldsymbol{\pi} | \boldsymbol{Z}^{-}, \boldsymbol{\alpha})$ is the posterior Dirichlet defined in eq. (3.18). This results in:

$$\begin{aligned} P(z_{nk} = 1 | \boldsymbol{Z}^{-}, \boldsymbol{\alpha}) &= \int P(z_{nk} = 1 | \boldsymbol{\pi}) \cdot p(\boldsymbol{\pi} | \boldsymbol{Z}^{-}, \boldsymbol{\alpha}) \ d\boldsymbol{\pi} \\ &= \frac{c_k + \alpha_k}{\sum_{k=1}^{K} c_k + \alpha_k} \end{aligned} \tag{3.22}$$

   A derivation for the integral in eq. (3.22) can be found in Ch. 24 of [90]. In the result of eq. (3.22), c_k denotes the number of data points (peaks) currently assigned to cluster

k, excluding the n-th peak that is being sampled.

Eq. (3.22) reveals the clustering property of the Dirichlet-multinomial mixture model. The larger a k-th cluster is, the greater the prior probability for the currently sampled peak to be placed into that cluster. This is balanced by the prior hyperparameter α_k. In the absence of any prior knowledge, often α_k is set to be symmetric ($\alpha_k = \frac{\alpha}{K}$), and in this case, small values for α_k will result in fewer, larger clusters as c_k dominates, while large values for α_k will smoothen the prior probabilities, reducing the influence of c_k and producing more uniform clusters. In this manner, α_k plays the role of the pseudo-count that controls how strong the influence of c_k is.

Having derived the terms in the conditional distribution for z_{nk}, we can now describe our CGS for this mixture model. We loop over each peak in a random order and remove the information of that peak from the model. We then resample the assignment of peak n to each of the K clusters using eq. (3.19), normalised to form a probability distribution. Upon sampling a new cluster ($z_{nk} = 1$), we assign peak n to cluster k and add the information of that peak back into the model. This consists of one iteration in our CGS. Each iteration generates a sample, and the collection of samples can be used to approximate our posterior model parameters.

3.3 Dirichlet Process Mixture Model Clustering

One parameter that has to be specified in the model is K, the number of mixture components. However, in many cases, often we do not know the number of components in advance. Determining the number of clusters in general is a challenging problem. In the Bayesian context, selecting K (and also other model parameters) constitute the model selection process. Bayesian non-parametric approach provides a way to perform model selection on the number of mixture components in a principled manner by assuming that there is an infinite set of parameters, but the observed data is generated from a finite subset of that. In this manner, for the non-parametric clustering approach, we do not specify the number of cluster K *a-priori* but instead assume that the data is generated from a mixture of an infinite number of components. The non-parametric clustering model then learns the number of clusters from the data, allowing for the automatic determination of model complexity [91].

Dirichlet Process (DP) [92] is a stochastic process that describes a distribution over probability measures, often used in Bayesian non-parametric mixture model to generate the prior distributions over the mixture components when the number of component is unknown. Here, we focus on providing a brief overview on Dirichlet Process to use in the construction of an infinite mixture model. For more details, the reader is referred to [90, 88, 91, 93]. The DP is

parameterised by a base distribution H and a concentration parameter α, i.e. $DP(H, \alpha)$. A constructive definition for the DP is given through the stick-breaking construction [94]. Let $\boldsymbol{\pi}$ be an infinite-dimensional mixture proportion vector, consisting of the following infinite sequence of entries $\boldsymbol{\pi} = \{\pi_1, \pi_2, ..., \}$. Then $\boldsymbol{\pi} \sim GEM(\alpha)$, where GEM stands for the Griffiths, Engen and McCloskey distribution, if we can generate the k-th entry in $\boldsymbol{\pi}$ with the following stick-breaking process:

$$\begin{aligned} \beta_k &\sim Beta(1, \alpha) \\ \pi_k &= \beta_k \prod_{l=1}^{k-1} (1 - \beta_l) \end{aligned} \tag{3.23}$$

To use eq. (3.23), first we see that expanding the expression for π_k in eq. (3.23) results in the following recursive definition, where each π_k is a product of β_k and $(1 - \sum_{l=1}^{k-1} \pi_l)$.

$$\begin{aligned} \pi_1 &= \beta_1 \\ \pi_2 &= \beta_2(1 - \beta_1) \\ \pi_3 &= \beta_3(1 - \beta_2)(1 - \beta_1) \\ &= \beta_3(1 - \beta_1 - \beta_2(1 - \beta_1)) \\ &= \beta_3(1 - \pi_1 - \pi_2) \\ &... \\ \pi_k &= \beta_k(1 - \sum_{l=1}^{k-1} \pi_l) \end{aligned} \tag{3.24}$$

So to start the stick-breaking process, we begin with a stick of length 1 and generate a random variable $\beta_1 \sim Beta(1, \alpha)$. This random variable β_1 is used to define the position to break the stick initially at $\pi_1 = \beta_1$. We repeat infinitely the step of generating a new $\beta_k \sim Beta(1, \alpha)$ and using it to break the remaining portion of the stick $(1 - \sum_{l=1}^{k-1} \pi_l)$ at π_k. The result of this process is a vector $\boldsymbol{\pi}$ that is Dirichlet-distributed. It can be shown that $\sum_{k=1}^{\infty} \pi_k = 1$ [94]. The stick-breaking process defines an exchangeable distribution: although parts of the sticks are generated in order and each part is conditioned on the previous ones, the resulting joint distribution is independent of the order.

In the mixture setting, $\boldsymbol{\pi}$ generated in this manner can be used as the mixture proportions in an infinite mixture model. Instead of sampling a finite-length vector from the Dirichlet distribution in eq. (3.3), we generate $\boldsymbol{\pi}$ from the stick-breaking process. This lets us formulate the generative process for our data as a mixture model having infinitely-many mixture components using the stick-breaking construction as the prior over $\boldsymbol{\pi}$, resulting in the graphical model shown in Figure 3.1B. This also lets us specify an infinite mixture model in term of samples from the Dirichlet Process. Given $\boldsymbol{\pi}$ generated as before and a base distribution H

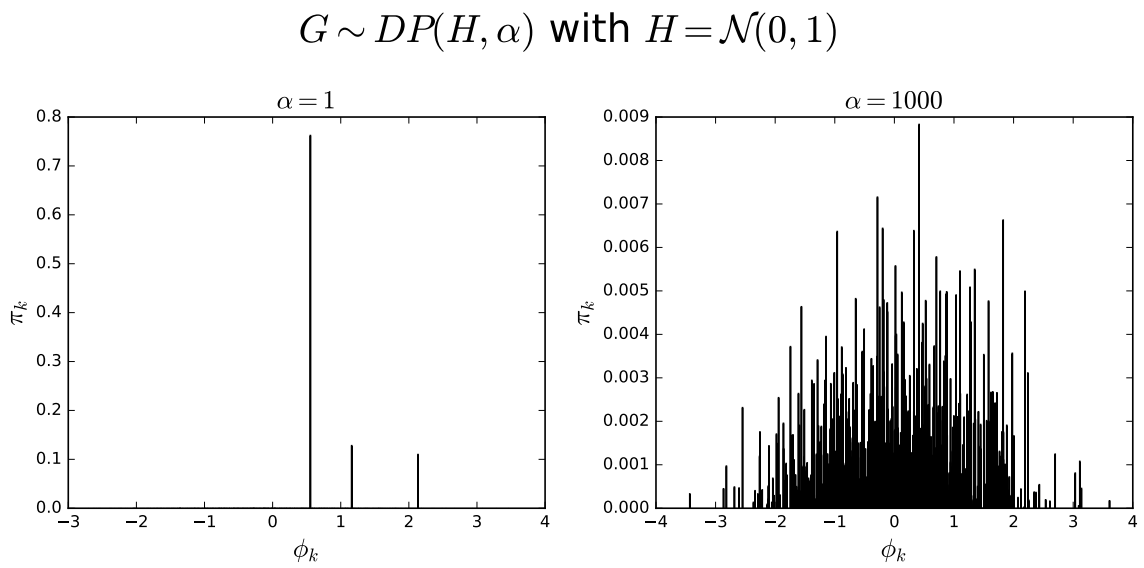


Figure 3.2: Two samples of G, plotted up to 1000 discrete values, generated by a Dirichlet Process with $\alpha = 1$ (left) and $\alpha = 1000$ (right) and a base distribution $\mathcal{N}(0, 1)$. We see that α affects how smooth the resulting discretisation of H is in G.

that we can sample from, let ϕ_k be a value sampled from H (this can be, for instance, the mean μ_k of a mixture component). Then the following G is a discrete distribution that is also an infinite mixture model:

$$G = \sum_{k=1}^{\infty} \pi_k \delta_{\phi_k} \tag{3.25}$$

where δ_{ϕ_k} is the delta function that has its entire probability distribution concentrated at ϕ_k. We say that $G \sim DP(\alpha, H)$ is a sample from the Dirichlet Process parameterised by the concentration parameter α and the base distribution H [93]. In this manner, the DP is a distribution over distributions. A sample from the DP is a discrete probability distribution, even if the base distribution H is continuous. The level of this discretisation is controlled by α. Small α results in a distribution concentrated at fewer discrete values, while large α produces a distribution with support over many discrete values.

An alternative formulation of an infinite mixture model can be given using a DP parameterised by α and the base distribution $H = \mathcal{N}(\mu_0, \sigma_0^2)$. With a fixed variance (σ^2) that represents a user-defined RT drift tolerance, the generative process for our peak RT data becomes:

$$\begin{aligned} G|\alpha, H &\sim DP(\alpha, H) \\ \mu_k|G &\sim G \\ y_n|\mu_k &\sim \mathcal{N}(y|\mu_k, \sigma^2) \end{aligned} \tag{3.26}$$

Figure 3.2 shows two examples of G drawn from a DP with a Gaussian base distribution, $H = \mathcal{N}(0, 1)$ and varying values for α. As noted earlier, a key property of the DP is that

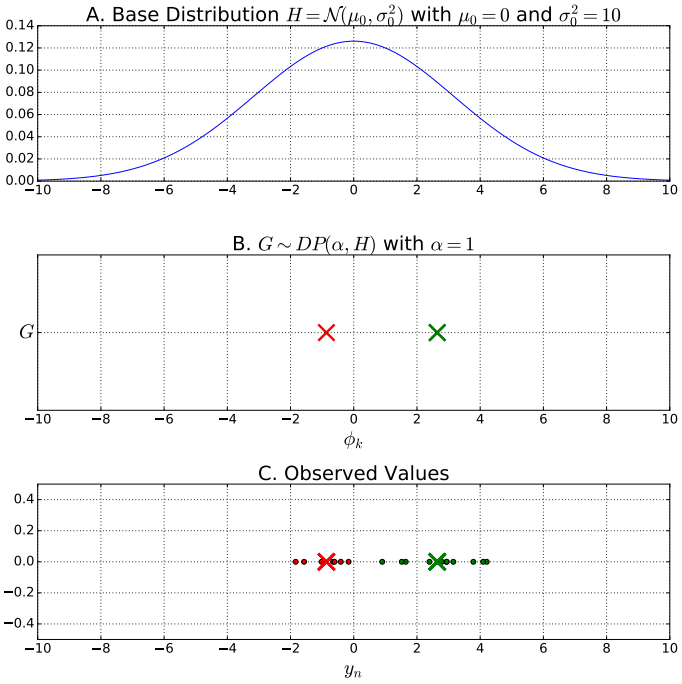


Figure 3.3: An illustration of the generative process for the DP mixture model in eq. (3.26). First, we require a base distribution to sample for discrete values, shown in **(A)**. Given H and some concentration parameter α, we generate a discrete distribution G. In **(B)**, we see that G contains two unique discrete values sampled from H, represented by the red and green crosses. Each discrete value in G ϕ_k is also a cluster mean μ_k. **(C)** The noisy observed values is generated by sampling for μ_k from G, and sampling for y_n conditioned on the μ_k.

G is a discrete distribution despite H being continuous. The level of discretisation of H is controlled by α, with small α resulting in fewer discrete points with larger probabilities in G and large α causing G to have more discrete points. Repeated sampling from G will generate repeated values, which illustrates the usefulness of the DP, in particular for clustering by setting the DP as a prior over the distribution of the mixture components.

We also show in Figure 3.3 how the generative process in eq. (3.26) works. First we sample a distribution G from the DP parameterised by α and a base distribution H. In the example of Figure 3.3A, the base is set to $\mathcal{N}(0, 1)$ and G contains two unique discrete values, denoted by the red and green crosses, each of which is also a cluster component mean μ_k. Every n-th data point is associated with a μ_k, and each μ_k generates the data for its cluster by sampling for values from $\mathcal{N}(\mu_k, \sigma^2)$.

3.3.1 Collapsed Gibbs Sampling for a Dirichlet Process Mixture Model

Having defined the generative process for an infinite mixture model, we are now interested in performing inference on the model parameters. In particular for Gibbs sampling, we require the probability of the current n-th data point that is being sampled to be in a cluster ($P(z_{nk} = 1)$, conditioned on the assignments of other data points ($\boldsymbol{Z}^-$). We show how we derive this by modifying the original collapsed Gibbs sampler as the number of components K goes to infinity. Assuming a symmetric prior on the Dirichlet hyperparameter, $\alpha_k = \frac{\alpha}{K}$, then the conditional probability on the assignment of peak n to cluster k from eq. (3.22) becomes:

$$P(z_{nk} = 1|\boldsymbol{Z}^-, \boldsymbol{\alpha}) = \frac{c_k + \frac{\alpha}{K}}{\alpha + N - 1} \tag{3.27}$$

As before, c_k in eq. (3.27) refers to the count of the number of peaks assigned to cluster k, excluding the current one being sampled. In the denumerator of eq. (3.27), we also see $\alpha + N - 1$ instead of just $\alpha + N$ to exclude the current data point being sampled in this iteration of Gibbs sampling. Then taking the limit of eq. (3.27) as K goes to infinity results in the following conditional probability for $z_{nk} = 1$:

$$P(z_{nk} = 1|\boldsymbol{Z}^-, \boldsymbol{\alpha}) = \begin{cases} \frac{c_k}{\alpha+N-1}, & \text{for existing clusters} \\ \frac{\alpha}{\alpha+N-1}, & \text{for a new cluster} \end{cases} \tag{3.28}$$

The conditional probability in eq. (3.28) is also often formulated as the Chinese Restaurant Process (CRP). In this process, an analogy is proposed based on a Chinese restaurant having an infinite number of tables. Tables correspond to clusters while customers correspond to the observed data points. The CRP is a random process that defines the probability of a customer to sit at a particular table, conditioned on the seating arrangements of other customers. For a non-empty table, this probability is proportional to c_k, the number of other customers seated at a table, while for an empty table, the probability is proportional to α. Under exchangeablity, the joint posterior distribution in the CRP is invariant to the ordering of the items [95]. This means we can assume that the n-th data point is the last customer to arrive in the CRP, and the conditional probability for $P(z_{nk} = 1)$ is thus proportional to eq. (3.28). Coupled with a likelihood for a data point to be assigned to a table, we can use this to modify our conditional probability for Gibbs sampling, resulting in:

$$P(z_{nk} = 1|\boldsymbol{Z}^-, \boldsymbol{\alpha}) \propto \begin{cases} c_k \cdot p(y_n|\boldsymbol{Z}^-, \boldsymbol{y}^-, \mu_0) \\ \alpha \cdot p(y_n|\mu_0) \end{cases} \tag{3.29}$$

The top term in eq. (3.29) corresponds to the probability of being assigned to existing non-empty clusters. As in the finite mixture model case, this prior probability is proportional to c_k, the number of data points (peaks) currently assigned to cluster k excluding the n-th peak that is being sampled, while the likelihood $p(y_n|\boldsymbol{Z}^-, \boldsymbol{y}^-, \mu_0)$ is defined in eq. (3.20) or equivalently in eq. (3.21) when precision is used. The bottom term in eq. (3.29) corresponds to the probability of creating a new cluster with the prior probability proportional to α multiplied by the data likelihood. In this particular case due to conjugacy, we can derive $p(y_n|\mu_0)$, the likelihood of y_n conditioned on the hyperparameter mean μ_0 directly by marginalising over all empty components, resulting in:

$$p(y_n|\mu_0) = \int p(y_n|\mu_k)p(\mu_k|\mu_0) \; d_{\mu_k} \quad = \mathcal{N}(y_n|\mu_0, \sigma^2 + \sigma_0^2) \tag{3.30}$$

In other cases where it is not possible to derive the data likelihood analytically, we can approximate it by sampling for a new μ_k from the prior instead and evaluating y_n under the new cluster mean [88]. This completes the necessary modification to our collapsed Gibbs sampler. The sampling process proceeds largely as before by removing the n-th peak from the model and performing the assignment of that peak to cluster k using eq. (3.29). If an existing k is selected, this is then the same as the finite mixture case. When a new k is selected, we create a new cluster and assign the peak to that cluster. In this manner, the number of mixture components is not fixed in advance *a priori*, but is instead determined based on the observed data and the choice of hyperparameter α.

3.4 Hierarchical Dirichlet Process Mixture Model Clustering

While the DP mixture model allows us to cluster related peaks together, the clustering process within each run is performed separately and independently of the others. However, in some cases it is useful to allow for clustering to be shared across runs. We call the clusters shared in this manner to be the *global clusters*, as opposed to the *local clusters* that are found in each file. In LC-MS data, global clusters can be assumed to correspond to compounds (e.g. metabolites or peptide fragments) that are present in all runs, while local clusters are the noisy realisation of such global clusters in each run. Often, the shared presence of global clusters can be assumed when we have multiple runs generated from the measurements of the same biological sample. In this case, the runs are called technical replicates. In other circumstances when the runs are generated through the measurements of different biological samples, shared compounds might also be found and can therefore be represented as global clusters.

Hierarchical Dirichlet Process (HDP) mixture model is an extension of the DP mixture model that allows for such global clusters to be defined and shared across multiple runs [96]. Within each file, the observed data points are clustered into local clusters, which are assigned to the shared global clusters. In our application, the HDP mixture model is particularly useful for alignment as being able to generatively explain which peaks are generated by which global clusters is the same as being able to match these peaks across runs. This application is shown in Chapter 6 where we introduce a HDP mixture model to simultaneously group peaks within and across runs. From the model, the matching of peaks (alignment) is constructed from the marginal probabilities of peaks of being assigned into the same global cluster.

To understand the HDP mixture model, first we need to define the hierarchical Dirichlet Process. Given a concentration parameter α_0 and a base distribution H, let G_0 to be a distribution sampled from a $DP(\alpha_0, H)$. Then for each file $j = 1, ...J$, we can sample a file-specific distribution G_j from another Dirichlet Process parameterised by α_j, with G_0 as its base distribution. This file-specific DP is denoted as $DP(\alpha_j, G_0)$, resulting in:

$$\begin{aligned} G_0|\alpha_0, H &\sim DP(\alpha_0, H) \\ G_j|\alpha_j, G_0 &\sim DP(\alpha_j, G_0) \end{aligned} \tag{3.31}$$

As a property of the DP, G_0 is a discrete distribution with probabilities that sums to 1 (regardless of whether H is continuous or discrete). This means G_0 has a support over the discrete values $\phi_1, \phi_2, ...$ drawn from its base distribution H. We use G_0 to represent the prior distribution over the global clusters. Each G_j is a prior distribution over the file-specific local clusters, and since G_j is drawn from a DP with G_0 as its base distribution, each G_j is discrete and has a support over the same discrete values as G_0. This is where the sharing property of the HDP comes about. The set of discrete values in G_0, which represent the prior values on the global clusters, are inherited (copied) to be the discrete values in G_j, which represents represent the prior values on the local clusters.

To define a HDP mixture model, we complete the hierarchical prior defined in eq. (3.31) with a data-generating distribution. Continuing with our example of clustering peaks by their RT values, the data now comes in J input files, each corresponding to an LC-MS run. Let $j = 1, ..., J$ indexes over the input files, while $n = 1, ..., N$ indexes over the peaks in a particular j-th file. Within a j-th file, the observed data is $\boldsymbol{y}_j = \{y_{j1}, y_{j2}, ..., y_{jn}\}$. We assume that within a file, the noise on the observed RT values is generated by a Gaussian with mean μ_{jk} and a fixed variance σ^2. As the base distribution, we set $H = \mathcal{N}(\mu_0, \sigma_0^2)$. Together with the prior in eq. (3.31), we obtain the following HDP mixture model that explains how the

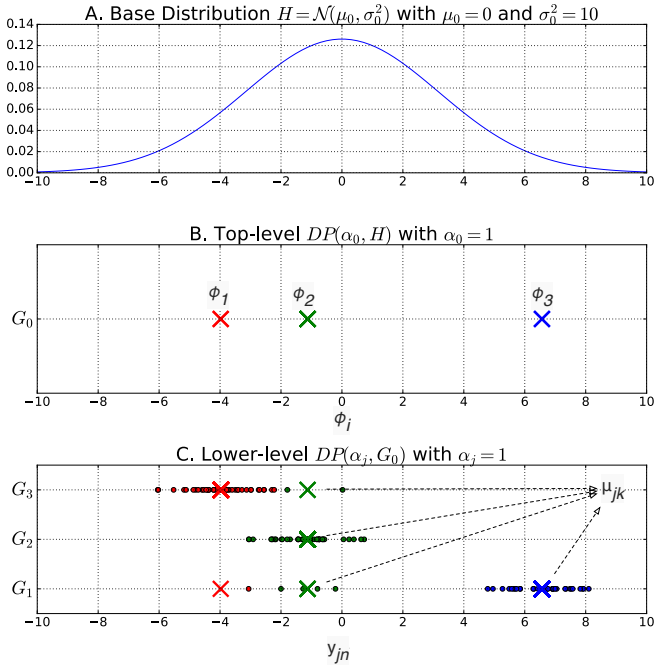


Figure 3.4: An illustration of the generative process for the HDP mixture model defined in eq. (3.32). Similar to the DP mixture, we have a base distribution shown in **(A)**. Given H and some concentration parameter α, we generate a global distribution G_0. In **(B)**, we see that G_0 contains three unique discrete values. We then generate a file-specific distribution G_j by sampling from a DP with G_0 as the base distribution. As a consequence, G_j contains only discrete values copied from G_0. In **(C)**, the noisy observed values within each file is generated by sampling for μ_k from G_j, and sampling for y_n conditioned on the μ_k.

RT values of peaks in multiple files can be generated:

$$\begin{aligned} G_0|\alpha_0, H &\sim DP(\alpha_0, H) \\ G_j|\alpha_j, G_0 &\sim DP(\alpha_j, G_0) \\ \mu_{jk}|G_j &\sim G_j \\ y_{jn}|\mu_{jk} &\sim \mathcal{N}(y_{jn}|\mu_{jk}, \sigma^2) \end{aligned} \tag{3.32}$$

This generative process from the HDP mixture model in eq. (3.32) is also illustrated in Figure 3.4. Note the key difference between HDP mixture (Figure 3.4) and the DP mixture (Figure 3.3), in particular the addition of another level of hierarchy to the HDP mixture model, where within a file j, the discrete values μ_{jk} in G_j is drawn from another DP having G_0 as its base which makes it possible for clustering parameters to be shared.

3.4.1 Gibbs Sampling for a Hierarchical Dirichlet Process Mixture Model

Inference for the HDP mixture model can also be performed via a Gibbs sampling scheme. In the following subsections, we describe the construction of a Gibbs sampler for the HDP mixture model in eq. (3.32). This follows from the *posterior sampling in the Chinese restaurant franchise* approach in Section 5.1 of [**?**]. In addition, we also describe in Chapter 6 the construction of a more elaborate Gibbs sampling scheme for the HDP-Align model.

As a preliminary to the Gibbs sampler, the following indices are defined: $j = 1, ..., J$ indexes over the files, $n = 1, ..., N$ indexes peaks in a j-th file (assume that all files have the same number of peaks), $k = 1, ..., K$ indexes the local clusters in a j-th file and $i = 1, ..., I$ indexes the shared global clusters across all files. Within a j-th file, the observed data is the RT values of peaks, denoted by $\boldsymbol{y}_j = \{y_{j1}, y_{j2}, ..., y_{jn}\}$. At any point in the sampler, the set of local cluster parameters (means) in the j-th file is denoted by $\{\mu_{j1}, \mu_{j2}, ..., \mu_{jk}, \}$, while the set of global cluster parameters across all files is denoted by $\{\phi_1, \phi_2, ..., \phi_i\}$. Note that each local cluster parameter is a copy of a global cluster parameter in a particular file. When the global parameter is updated, all its local copies are updated too.

We also require keeping track of some count variables, namely c_{jk} the number of peaks currently assigned to the k-th local cluster in the j-th file and c_i the number of local clusters from all files currently assigned to the i-th global cluster. Note that both counts should exclude the object being sampled in the current iteration of the Gibbs sampler.

The conditional updates for the Gibbs sampler are given as follows:

1. **Assigning Peaks to Local Clusters**

   Let the indicator variable $z_{jnk} = 1$ denote the assignment of peak n in file j to an existing local cluster k in the same file. Additionally, $z_{jnk^*} = 1$ denotes the assignment of peak n in file j to a new local cluster k^*. For Gibbs sampling, we need to derive the conditional probability of $P(z_{jnk} = 1)$ given the current peak RT value y_{jn} and other model parameters. We denote this by $P(z_{jnk} = 1|y_{jn}, ...)$, where ... refers to other parameters being conditioned upon but not explicitly listed. Similar to the single-file collapsed Gibbs sampling in eq. (3.29), this conditional probability is given by:

$$P(z_{jnk} = 1|y_{jn}, ...) \propto \begin{cases} c_{jk} \cdot p(y_{jn}|z_{jnk} = 1, ...) \\ \alpha_j \cdot p(y_{jn}|z_{jnk^*} = 1, ...) \end{cases} \tag{3.33}$$

   The conditional prior in eq. (3.33) is also known as the Chinese Restaurant Franchise (CRF), which can be seen as an extension of the CRP (described in Section 3.3.1) that accommodates multiple files. In the CRF, a file now corresponds to a restaurant,

each with an infinite number of tables. A new customer arrives at a restaurant and is assigned to an existing table with probability proportional to c_{jk} (the number of other customers already sitting at the table k in file j) or to a new table with probability proportional to α_j (the concentration parameter of the lower-level DP in file j). Across all restaurants, a global menu of dishes is maintained. The first customer who sits at a new table orders a dish from the global menu, which is shared by any subsequent customers who join that table. Existing dishes are served to the table with probability proportional to c_i (the number of tables across the entire franchise already served the i-th dish), while a new dish is created with probability proportional to α_0 (the concentration parameter of the top-level DP).

For the assignment of a data point y_{jn} to a local cluster, a customer in the CRF analogy corresponds to a peak, a table is a local cluster parameter while a dish is a global cluster parameter. We consider the top and bottom terms of eq. (3.33) separately. The top term $p(y_{jn}|\boldsymbol{z}_{jnk} = 1, ...)$ is the probability of assigning the data point to an existing k-th local cluster in file j having the cluster mean μ_{jk}. This is proportional to c_{jk} multiplied by the likelihood $\mathcal{N}(y_{jn}|\mu_{jk}, \sigma^2)$. The bottom term $p(y_{jn}|z_{jnk^*} = 1, ...)$ is the probability of assigning the data point to a new local cluster k^*. This is proportional to α_j, multiplied by the likelihood of y_{jn} under the new local cluster. To evaluate this likelihood, first we generate a new cluster mean μ_{jk^*} by sampling from the top-level Dirichlet Process $DP(\alpha_0, H)$. Let $\phi_1, \phi_2, ...\phi_i$ be the currently existing global cluster parameters shared across files. With probability proportional to c_i, an existing ϕ_i is instantiated as a new local cluster mean μ_{jk^*} in file j. Alternatively, with probability proportional to α_0, a new ϕ_{i^*} is sampled from the base distribution $\mathcal{N}(\mu_0, \sigma_0^2)$ and instantiated as a new μ_{jk^*} in file j.

2. **Assigning Local Clusters to Global Clusters**

As the next step of the Gibbs sampler, we can also sample the assignment of a local cluster k (and its entire member peaks) in file j to a global cluster, allowing for multiple data points to be moved at the same time. This

Let the indicator variable $v_{jki} = 1$ denote the assignment of a local cluster k in file j to an existing global cluster i, and $v_{jki^*} = 1$ denote the assignment of the local cluster to a new global cluster i^*. Furthermore, at any point in the sampler, let $\boldsymbol{x}_{jk} = \{x_{j1}, x_{j2}, ..., x_{jm}\}$ be the RT values of peaks currently assigned to the local cluster k in file j, and there are M_{jk} such peaks.

Similar to eq. (3.33), the conditional prior for the assignment of a local cluster to a

global cluster follows the CRP, resulting in:

$$P(v_{jki}=1|\boldsymbol{x}_{jk},...) \propto \begin{cases} c_i \cdot p(\boldsymbol{x}_{jk}|v_{jki}=1,...) \\ \alpha_0 \cdot p(\boldsymbol{x}_{jk}|v_{jki^*}=1,...) \end{cases} \tag{3.34}$$

In eq. (3.34), $p(\boldsymbol{x}_{jk}|v_{jki}=1,...)$ is given by the likelihood of the member peaks $\boldsymbol{x}_{jk}$ of local cluster k in file j to be placed under a global cluster i with parameter ϕ_i, therefore $p(\boldsymbol{x}_{jk}|v_{jki}=1,...) = \prod_{m=1}^{M_{jk}} \mathcal{N}(x_{jm}|\phi_i,\sigma^2)$ following the assumed independence assumption of x_{jm} conditioned on ϕ_i. Similarly, to evaluate $p(\boldsymbol{x}_{jk}|v_{jki^*}=1,...)$, first we sample for a new ϕ_{i^*} from the base distribution $\mathcal{N}(\mu_0,\sigma_0^2)$ and evaluate the data likelihood of $\boldsymbol{x}_{jk}$ under ϕ_{i^*}.

3. **Updating Other Parameters**

   As the last step of our Gibbs sampler, we update each of global cluster parameter ϕ_i (and also its instantiated copies in each of the j-th file). For a particular i-th global cluster, this depends on the observations currently associated to it via any of the local clusters. We now denote by $\boldsymbol{x}_i$ the set of RT values of peaks currently assigned to the i-th global cluster from across all the files, and there are M_i such peaks. The posterior density for ϕ_i is given by:

$$p(\phi_i|\boldsymbol{x}_i,\mu_0,..) \propto \mathcal{N}(\phi_i|\mu_0,\sigma_0^2) \prod_{m=1}^{M_i} \mathcal{N}(x_i|\phi_i,\sigma^2) \tag{3.35}$$

   As in eq. (3.17) for the single-file DP mixture case, this posterior can be simplified into another Gaussian $\mathcal{N}(\phi_i|\tilde{\mu},\tilde{\sigma}^2)$ parameterised by:

$$\tilde{\mu} = \tilde{\sigma}^2\left(\frac{\sum_{m=1}^{M_i} x_m}{\sigma^2} + \frac{\mu_0}{\sigma_0^2}\right), \quad \tilde{\sigma}^2 = \frac{1}{\frac{M}{\sigma^2}+\frac{1}{\sigma_0^2}} \tag{3.36}$$

   Appendix A in [**?**] also describes how the DP concentration parameters for each α_j and also the α_0 can be updated..

This completes the Gibbs sampler for the HDP mixture model defined in eq. (3.32). We initialise the sampler by putting all peaks in the same file into one local cluster and assign all local clusters into one global cluster. The sampler then operates by repeating the three steps: re-sampling local cluster membership for all RT values following step (1) above, and re-sampling the local cluster to global cluster membership for all local clusters following step (2) above, and finally updating the parameters as in step (3). During the re-sampling steps, we keep track of which existing local clusters are empty in file j via the count variables $\{c_{j1}, c_{j2}, ..., c_{jk}\}$. An empty table cannot have any new member assigned to it and can be

deleted. Upon deleting a local cluster, we also update the global counter c_i that tracks how many local clusters across all files are assigned to the i-th global cluster. A global cluster that is empty (not instantiated in any file) is also deleted.

As the Gibbs sampling procedure above has shown (and also as can be observed from Figure 3.4), we see that within a file-specific G_j, each discrete value for μ_{jk} is copied exactly from the discrete values of ϕ_i from the global G_0. This may not be what we want. In particular, for our application, the top-level G_0 is the prior distribution over global compounds, which can be expected to correspond to metabolites (for metabolomics data) or peptide fragments (for proteomics data). The sample from a file-specific DP, denoted by G_j, is the prior distribution over the realisation of those compounds within a file. It is reasonable to expect the discrete values in G_j to vary from G_0 with some random noise that represents the RT drift in the observed peaks. The addition of noise in this manner results in the 'HDP with random effects' model [97].

Mixture models are used extensively throughout this thesis. In the coming Chapter 4, a DP mixture model is used to group related peaks by their RT. In Chapter 5, this is extended to a mixture model that groups related peaks by taking into account their mass transformations. In Chapter 6, we propose HDP-Align, a HDP mixture model that resembles [97] but incorporates other LC-MS specific information, such as the m/z and RT values, to cluster peaks hierarchically across multiple files. Beyond clustering, the model is used to induce alignment (matching) of peaks from different files by placing peaks by placing them into the same RT and m/z clusters. Finally, in Chapter 7, a topic model based on Latent Dirichlet Allocation (LDA) is proposed to decompose fragmentation spectra into a set of co-occuring fragment peaks, allowing for better hypothesis generations in the identification of metabolites present in the sample. In the following Section 3.5, we provide a brief introduction to the LDA model.

3.5 Latent Dirichet Allocation

We now turn our attention to a different kind of extension to the standard mixture model. In the standard mixture setting, a set of related data points that can be grouped together are said to be explained by a cluster (a probability distribution). For instance, a set of peaks are related through having close RT values. They can be grouped together under one cluster, and following the generative process, we assume that all observations in the same group are produced by a single probability distribution. In this section, we relaxes that assumption and introduces Latent Dirichlet Allocation (LDA) [98], another generative model that allows for related data points in the same group to be produced by a mixture of distributions instead.

The classical application of LDA lies for topic discovery for the text domain, although LDA-

like models have been applied to continuous data [99, 100, 101]. In the text application, data points are the individual words, which can be grouped, forming a document. In a document, certain words tend to co-occur together — for instance, 'bayesian' and 'probability' are two such words – and ignoring word orders, we can represent this pattern of co-occurrences by a multinomial distribution the counts of words in a document. In the standard mixture model construction, we would have a document assigned to a cluster, and all words from the same document generated by sampling from the same multinomial distribution linked to the cluster. LDA relaxes that assumption by allowing for a document to contain words generated by a mixture of different *topics*. A topic in this case still corresponds to a multinomial distribution over the entire vocabulary space.

We now describe the LDA model for text. Let $n = 1, ..., N$ indexes the unique words (vocabularies) and $d = 1, ..., D$ indexes the documents in our collection. Thus w_{dn} refers to the n-th word in the d-th document. We also require the index $k = 1, ..., K$ over the topics, and let z_{dn} to refer to the assignment of w_{dn} to any of the k-th topic. The generative process for LDA is given as follows. For each d-th document, we sample a multinomial $\boldsymbol{\theta}_d$ over the K topics. This document-to-topic distribution provides the mixture proportions of topics that explain the words in the document. Each document is also associated to a $\boldsymbol{z}_d$, the vector of assignment of its words to topics. Then for each word in the document, we sample z_{dn}, the assignment of the n-th word in the d-th document to a particular k-th topic. Given $z_{dn} = k$, we generate the actual word w_{dn} by sampling from the k-th topic-to-word distribution $\boldsymbol{\phi}_{z_{dn}}$ that this word is assigned to through $z_{dn} = k$. A prior Dirichlet distribution parameterised by $\boldsymbol{\alpha}$ is placed on the document-to-topic multinomials, and similarly, another prior Dirichlet parameterised by $\boldsymbol{\beta}$ is placed on the topic-to-word multinomials. The generative model for LDA can be summarised as the following, and is illustrated in Figure 3.5

$$\begin{aligned} w_{dn}|\boldsymbol{\phi}_{z_{dn}} &\sim Multinomial(\boldsymbol{\phi}_{z_{dn}}) \\ z_{dn}|\boldsymbol{\theta}_d &\sim Multinomial(\boldsymbol{\theta}_d) \\ \boldsymbol{\theta}_d|\boldsymbol{\alpha} &\sim Dir(\boldsymbol{\alpha}) \\ \boldsymbol{\phi}_k|\boldsymbol{\beta} &\sim Dir(\boldsymbol{\beta}) \end{aligned} \tag{3.37}$$

To state the joint distribution of the model, we need to define more notations. Let $\boldsymbol{W} = \{\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_d\}$ denote the entire collection of documents and $\boldsymbol{Z}$ denotes the entire set of assignment variables for all documents, $\boldsymbol{Z} = \{\boldsymbol{z}_1, \boldsymbol{z}_2, ..., \boldsymbol{z}_d\}$. We put the multinomial parameter sets for all the document-to-topic distributions into $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_d\}$. Similarly, the multinomial parameter sets for all topic-to-word distributions are put into

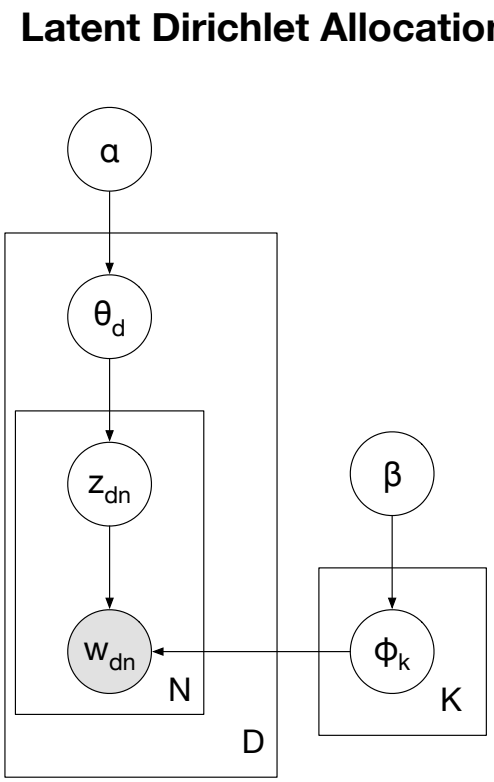


Figure 3.5: Graphical model of the Latent Dirichlet Allocation model. Circles denotes random variables, while the shaded node denotes the observed word value.

$\boldsymbol{\Phi} = \{\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, ..., \boldsymbol{\phi}_k\}$. Then the joint probability distribution is given by:

$$\begin{aligned} p(\boldsymbol{W}, \boldsymbol{Z}, \boldsymbol{\Theta}, \boldsymbol{\Phi} | \boldsymbol{\alpha}, \boldsymbol{\beta}) &= p(\boldsymbol{\Phi}|\boldsymbol{\beta}) \cdot p(\boldsymbol{\Theta}|\boldsymbol{\alpha}) \cdot p(\boldsymbol{Z}|\boldsymbol{\Theta}) \cdot p(\boldsymbol{W}|\boldsymbol{\Phi}, \boldsymbol{Z}) \\ &= \left[\prod_{k=1}^{K} p(\boldsymbol{\phi}_k|\boldsymbol{\beta})\right] \cdot \left[\prod_{d=1}^{D} p(\boldsymbol{\theta}_d|\boldsymbol{\alpha})\right] \cdot \left[\prod_{d=1}^{D}\prod_{n=1}^{N} p(z_{dn}|\boldsymbol{\theta}_d)\right] \cdot \left[\prod_{d=1}^{D}\prod_{n=1}^{N} p(w_{dn}|\boldsymbol{\phi}_{z_d}\right. \end{aligned} \tag{3.38}$$

3.5.1 Collapsed Gibbs Sampling for Latent Dirichlet Allocation

Similar to the mixture model case, inference in LDA can be performed via a collapsed Gibbs sampling scheme. In particular, we are interested in the conditional probability of $P(z_{dn} = k|\boldsymbol{Z}^-, \boldsymbol{W}, \boldsymbol{\alpha}, \boldsymbol{\beta})$, the assignment of word n in document d to topic k given other assignments and model hyperparameters. This is proportional to the joint distribution given in eq. (3.38). For collapsed Gibbs sampling in LDA, we also aim to integrate out the document-to-topic distributions $\boldsymbol{\Theta}$ and the topic-to-word distributions $\boldsymbol{\Phi}$ from the joint distribution in eq. (3.38).

This results in:

$$\begin{aligned} P(z_{dn} = k|\boldsymbol{Z}^{-}, \boldsymbol{W}, \boldsymbol{\alpha}, \boldsymbol{\beta}) &\propto \int_{\boldsymbol{\Theta}} \int_{\boldsymbol{\Phi}} p(\boldsymbol{W}, \boldsymbol{Z}, \boldsymbol{\Theta}, \boldsymbol{\Phi}|\boldsymbol{\alpha}, \boldsymbol{\beta}) \ d\boldsymbol{\Theta} \ d\boldsymbol{\Phi} \\ &\propto \int_{\boldsymbol{\Theta}} \int_{\boldsymbol{\Phi}} p(\boldsymbol{\Phi}|\boldsymbol{\beta}) \cdot p(\boldsymbol{\Theta}|\boldsymbol{\alpha}) \cdot p(\boldsymbol{Z}|\boldsymbol{\Theta}) \cdot p(\boldsymbol{W}|\boldsymbol{\Phi}, \boldsymbol{Z}) \ d\boldsymbol{\Theta} \ d\boldsymbol{\Phi} \\ &\propto \left[\int_{\boldsymbol{\Theta}} p(\boldsymbol{Z}|\boldsymbol{\Theta}) \cdot p(\boldsymbol{\Theta}|\boldsymbol{\alpha}) \ d\boldsymbol{\Theta} \right] \left[\int_{\boldsymbol{\Phi}} p(\boldsymbol{W}|\boldsymbol{\Phi}, \boldsymbol{Z}) \cdot p(\boldsymbol{\Phi}|\boldsymbol{\beta}) \ d\boldsymbol{\Phi} \right] \end{aligned} \tag{3.39}$$

The right hand side of eq. (3.39) can be separated into two parts: the prior term involving $\boldsymbol{Z}$, $\boldsymbol{\Theta}$ and $\boldsymbol{\alpha}$ and the data likelihood term involving $\boldsymbol{W}$, $\boldsymbol{\Phi}$ and $\boldsymbol{\beta}$. We denote the prior term by $p(z_{dn} = k|...)$ and the likelihood term by $p(w_{dn}|z_{dn} = k, ...)$, where ... denotes any other parameters being conditioned upon but not explicitly listed. A derivation for eq. (3.39) can be found in [102], but here we briefly summarise the result.

For the prior term $p(z_{dn} = k|...)$, marginalising over all $\boldsymbol{\theta}_d$ parameters produces:

$$\begin{aligned} P(z_{dn} = k|...) &= \int_{\boldsymbol{\Theta}} p(\boldsymbol{Z}|\boldsymbol{\Theta}) \cdot p(\boldsymbol{\Theta}|\boldsymbol{\alpha}) \ d\boldsymbol{\Theta} \\ &= \prod_{d=1}^{D} \int_{\boldsymbol{\theta}_d} \left[p(\boldsymbol{z}_d|\boldsymbol{\theta}_d) \cdot p(\boldsymbol{\theta}_d|\boldsymbol{\alpha}) \right] d\boldsymbol{\theta}_d \end{aligned} \tag{3.40}$$

After integrating the Dirichlet-Multinomial distribution from $p(\boldsymbol{z}_d|\boldsymbol{\theta}_d) \cdot p(\boldsymbol{\theta}_d|\boldsymbol{\alpha})$ in eq. (3.40) and simplifying the resulting expression that contains gamma functions, it can be shown in [102] that eq. (3.40) reduces to this simple expression:

$$P(z_{dn} = k|...) \propto c_{dk} + \alpha_k \tag{3.41}$$

with c_{dk} the number of words from document n currently assigned to topic k, excluding the current word being sampled. Similarly for the likelihood term of $P(w_{dn}|z_{dn} = k, ...)$, marginalising over all $\boldsymbol{\phi}_k$ parameters produces:

$$\begin{aligned} P(w_{dn}|z_{dn} = k, ...) &= \int_{\boldsymbol{\Phi}} p(\boldsymbol{W}|\boldsymbol{\Phi}, \boldsymbol{Z}) \cdot p(\boldsymbol{\Phi}|\boldsymbol{\beta}) \ d\boldsymbol{\Phi} \\ &= \prod_{k=1}^{K} \int_{\boldsymbol{\phi}_k} \left[p(\boldsymbol{\phi}_k|\boldsymbol{\beta}) \cdot \prod_{d=1}^{D} \prod_{n=1}^{N} p(w_{dn}|\boldsymbol{\phi}_{z_{dn}}) \right] \ d\boldsymbol{\phi}_k \end{aligned} \tag{3.42}$$

and it can be shown that eq. (3.42) reduces to:

$$P(w_{dn}|z_{dn} = k, ...) \propto \frac{c_{kn} + \beta_n}{\sum_n c_{kn} + \beta_n} \tag{3.43}$$

where c_{kn} is the total number of the n-th word currently assigned to topic k, excluding the current word being sampled.

Putting the prior and likelihood terms together, the following conditional distribution is obtained for the assignment of word n in document d to topic k:

$$P(z_{dn} = k|\boldsymbol{Z}^{-}, \boldsymbol{W}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \propto (c_{dk} + \alpha_k) \cdot \frac{c_{kn} + \beta_n}{\sum_n c_{kn} + \beta_n} \tag{3.44}$$

For each sample, we can also update the multinomial parameter sets $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_d\}$ for all documents and $\boldsymbol{\Phi}_k = \{\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, ..., \boldsymbol{\phi}_k\}$ for all topics. Consider one $\boldsymbol{\theta}_d$, the multinomial parameter of the d-th document-to-topic distribution. This multinomial distribution has a prior Dirichlet distribution parameterised by α and the observed counts c_{dk} of words from document d to topic k. Applying Bayes rule, we obtain the updated posterior which takes the form of a Dirichlet-multinomial distribution parameterised by $Dir(c_{d1}+\alpha_1, c_{d2}+\alpha_2, ..., c_{dk}+\alpha_k)$. The same applies to $\boldsymbol{\phi}_k$, the multinomial parameter of the k-th topic-to-word distribution.

Using the expectation of the Dirichlet distribution, we obtain the updated value for θ_{dk} (the k-th entry in $\boldsymbol{\theta}_d$) and also ϕ_{kn} (the n-th entry in $\boldsymbol{\phi}_k$) as:

$$\theta_{dk} = \frac{c_{dk} + \alpha_k}{\sum_k c_{dk} + \alpha_k}, \quad \phi_{kn} = \frac{c_{kn} + \beta_n}{\sum_n c_{kn} + \beta_n} \tag{3.45}$$

where c_{dk} is the count of words from document d assigned to topic k and c_{kn} is the count of the n-th words assigned to topic k.

The collapsed Gibbs sampling for LDA then proceeds as follows: given K topics, we initialise the sampler by randomly assigning words to topics and setting the count variables for c_{dk} used in eq. (3.41) and for c_{kn} used in eq. (3.43). We then iterate over the words in all documents, removing any information about it from the counts and computing the conditional probability of $P(z_{dn} = k|w_{dn}, ...)$ using eq. (3.39). The word is assigned to the k-th topic, so we update the indicator variable z_{dn} and other relevant count variables to reflect this assignment. We also obtain the updated multinomial parameters for $\boldsymbol{\theta}_d$ and $\boldsymbol{\phi}_k$ using eq. (3.45).

3.6 Conclusion

In this chapter, we have described the principle of mixture model clustering, with a particular example of its application to the generative modelling peak data by their retention time values. The mixture model starts by being a finite model, where the number of cluster is specified. With a Dirichlet Process prior, the mixture model is extended to an infinite model where the number of clusters grow with the data. Next, a hierarchical prior in form of the

hierarchical Dirichlet Process is also introduced to let us deal with shared clustering across multiple input files. Finally, Latent Dirichlet Allocation is introduced to let us model admixture data, where related items in a group is explained by a distribution of mixture. These models serve as the building blocks in the remaining parts of this thesis. In the coming Chapter 4, we explore the idea of using the information from peak grouping to improve the direct-matching alignment step that follows. In Chapter 5, a mixture model is proposed to group ionisation product peaks by the set of user-defined chemical transformations. This grouping is again used to improve the alignment step. In Chapter 6, a model based on the hierarchical Dirichle Process mixture is proposed to perform the grouping of ionisation product peaks across multiple runs, constructing alignment as a natural output and allowing for matching uncertainties of aligned peaksets to be returned to the user. Finally, in Chapter 7, an application of topic modelling is proposed to decompose fragmentation spectra into a set of co-occuring fragment peaks, allowing for better hypothesis generations in the identification of metabolites present in the sample. Common to all the chapters are the idea that many peaks that exist in LC-MS data are not independent, but instead share chemically meaningful relationship and can be grouped. These groupings can be explained by some underying latent variables that potentially correspond to peptides or metabolites, and this information can be used to improve other steps in the LC-MS data pre-processing pipeline.

Chapter 4

Incorporating Clustering Information into Peak Alignment

4.1 Introduction

In liquid chromatography measurements, peaks can experience non-linear shift in retention time (RT) values across runs [103]. RT variation could be due to instrument-specific factors (the condition of the chromatographic column itself. including flow rate variations, gradient slope and temperature [24]) or experiment-specific factors (e.g. instrument malfunctions or columns that need be replaced mid-experiment). In direct matching, several potential matches may be present for a peak from one run to another, but because the elution order of correspondent peaks may swap across runs [42], the candidate peaks nearest in distance are not necessarily the correct match.

As described in Section 2.3.4, ionization product (IP) peaks are the set of chemically-related peaks produced from the mass spectrometry measurement of a single compound, such as a peptide fragment (in the case of proteomics) or a metabolite (in metabolomics). Examples of IP peaks are isotope peaks, multiple adduct and deduct peaks, and fragment peaks. IP peaks of the same compound have similar chromatographic peak shapes as they co-elute from the column. Such information could potentially be used to improve matching since a group of IP peaks in one run should generally be aligned to another group of IP peaks in the other run. In the direct-matching approach (discussed in Section 2.3.2), correspondent peaks from one run to another are directly matched to each other without first correcting for RT drift (instead the assumption on RT noise is built into the distance/similarity function used for matching). A direct matching method can take this structural information of IP peaks into account in order to improve alignment, however none of the direct matching methods discussed in Section 2.3.2 exploit this information.

In this chapter, we propose clustering IP peaks that share similar RT values together. This clustering information is used to modify the similarity score matrix used for matching to bring groups of IP peaks that should be matched closer, with the key assumption that groups of co-eluting peaks corresponding to the same metabolite are generally preserved across runs. This idea is illustrated in Figure 4.1 and further introduced in Sections 4.3 and 4.3.2.

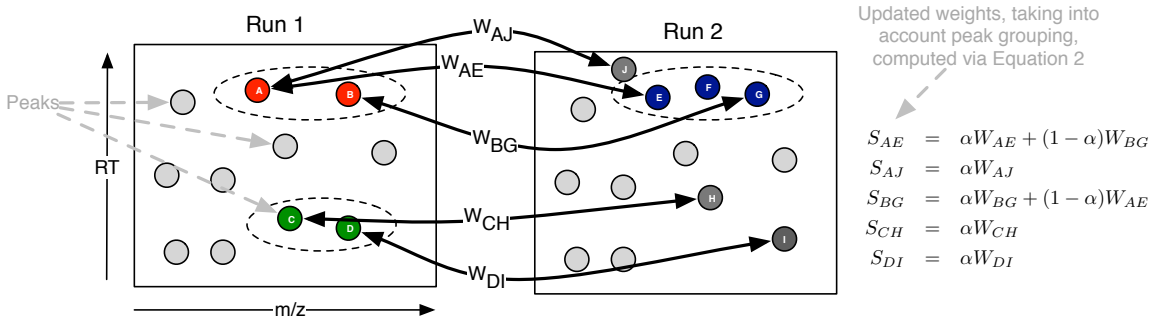


Figure 4.1: Illustrative example of the incorporation of grouping information into the similarity score. Each node in the figure is a peak feature, and dotted ovals represent groups of IP peaks, e.g. isotopes, fragments, etc. Initially weights (e.g. W_{AE}) are computed for pairs of peaks (one from each run) with m/z and RT within pre-defined thresholds. These weights are converted into an overall score by incorporating grouping information. For example, peak pairs (A, E) and (B, G) are both within the threshold. Because A and B are in the same group, and E and G are in the same group, the weights between pairs (A, E) and (B, G) are upweighted. Peak J is not related to any peaks that could be matched with A's IP peaks and the similarity between A and J is therefore downweighted (because $\alpha \leq 1$). The same applies to similarities between pairs (C, H) and (D, I).

As shown in Figure 4.1, initial weights are computed between pairs of peaks in the two runs that are within m/z and RT tolerances (e.g. W_{AE} and W_{AJ}). When related peak information is added, the similarity between peaks A and E is increased due to peak A being related to another peak (B) that is similar to a peak (G) related to E. On the other hand, the similarity between A and J is not increased as J does not have any IP peaks that could potentially be matched to peaks related to A. In other words, we are proposing using the structural dependencies present between peaks in each run to modify the similarity scores and improve alignment performance: the more peaks related to A that could be matched to peaks related to E, the more likely it becomes that A should be matched to E.

4.2 Related Work

Direct matching is introduced in Section 2.3.2, while the grouping of IP peaks is introduced in Section 2.3.4.

Statement of Original Work

The work from this chapter has been published in *Bioinformatics* [104]. The author proposed and implemented the idea of incorporating clustering information into a direct-matching alignment method. The author also performed the evaluation of performance of the proposed approach against the baseline methods.

4.3 A Direct Matching Method That Incorporates Clustering Information

Our proposed alignment method combines a novel similarity score with maximum weighted bipartite matching. This results in pairwise alignments which can be, if desired, extended to multiple alignments with hierarchical merging strategy. In such merging strategies, having an accurate initial pairwise alignments is important because of its influence on the final multiple alignment results. Here, we describe a direct matching approach to performing alignment of peaks across two LC-MS runs.

A peak feature refers to a tuple of $(m/z, RT)$ produced as output after the initial peak detection stage of LC-MS data. Here, m/z is the mass-to-charge value and RT the retention time value of a peak feature. Suppose we wish to align run A containing N_A peaks with run B containing N_B peaks. Alignment between two runs can be represented as a matching problem on a bipartite graph G, where nodes in the graph are the features, edges are the potential correspondence between features and the weights on the edges are the similarity scores (entries in S) between features. In SIMA [36], the Gale-Shapley algorithm [105] is used to find a stable matching in G. A matching is stable if there are no two features in different runs that would prefer to be matched to each other than to their currently matched partners. Since the stable matching is computed based on ranked preference, valuable information could be discarded as distances between features are converted to ranks. As such, we prefer to use a method that maximises the total sum of similarity scores of matched features (maximum weighted matching).

The benefit of maximum weighted bipartite matching in solving the peak correspondence problem has been studied in [38] in their LWBMatch tool. LWBMatch shows that such matching method, coupled to a local regression method, is able to align runs having large and systematic drifts in RT values. The well-known Hungarian algorithm [106] attributed to Kuhn and Munkres is used in LWBMatch to solve this problem. The time complexity of the Hungarian algorithm is $O(n^3)$, where n is the number of peaks in the larger set. While the Hungarian algorithm's implementation can be improved to $O(n^2 log\, n)$ by using Fibonacci heaps for the shortest path computation, the polynomial time complexity required in this

scheme is often too slow to be practical for alignments of the large number of runs produced in large-scale untargeted LC-MS studies. Consequently, we compute an approximation of the maximum weighted matching using a simple greedy algorithm that runs in $O(m \log n)$ time, where n and m denote the number of vertices and edges in the bipartite graph G to be solved. The greedy algorithm is straightforward to describe: pick the heaviest edge e in G, where e represents a potential match between nodes (features). Add e to the matching solution M and remove all other edges adjacent to e from G. Repeat until all edges in G have been exhausted. This simple greedy algorithm is known to provide a lower bound of at least 1/2 of the maximum weight in the matching [107]. From the direct matching of peaks across two runs, we obtain a list of aligned peaksets, defined as the sets of correspondent peaks that have been matched across runs. In the case of two runs, an aligned peakset has a size of at most 2.

4.3.1 Feature Similarity

To define a similarity measure between peaks, we follow SIMA [36] in using the Mahalanobis distance between two peaks $\boldsymbol{p}_i \in A$, $\boldsymbol{p}_j \in B$ where each peak is a vector of its m/z and RT values $\boldsymbol{p}_i = [m_i, t_i]^\mathsf{T}$ and $\boldsymbol{p}_j = [m_j, t_j]^\mathsf{T}$. The distance is given as:

$$D(\boldsymbol{p}_i, \boldsymbol{p}_j) = \sqrt{(\boldsymbol{p}_i - \boldsymbol{p}_j)^\mathsf{T} \boldsymbol{\Sigma}^{-1} (\boldsymbol{p}_i - \boldsymbol{p}_j)},$$

where the covariance matrix $\boldsymbol{\Sigma}$ is a diagonal matrix of mass-to-charge tolerance σ_m^2 and retention time tolerance σ_t^2. The diagonal covariance matrix $\boldsymbol{\Sigma}$ assumes independence between the σ_m^2 and σ_t^2 components since measurement error in m/z is independent of RT. To reduce the computational burden, entries in $\boldsymbol{D}$ are only computed when the peaks' m/z and RT values are within σ_m and σ_t. We now define the similarity score between two peaks as one minus their normalised distance:

$$W(\boldsymbol{p}_i, \boldsymbol{p}_j) = 1 - \frac{D(\boldsymbol{p}_i, \boldsymbol{p}_j)}{D_{max}}, \tag{4.1}$$

where D_{max} is the maximum computed distance between peaks in the two runs being aligned. Collectively, we call the $N_A \times N_B$ matrix of similarity scores between all peaks in run A and B to be $\boldsymbol{W}$.

4.3.2 Combining Related Peak Information

The similarity score matrix $\mathbf{W}$ can now be combined with related peak information to obtain a final score, $\mathbf{S}$:

$$\boldsymbol{S} = \alpha \boldsymbol{W} + (1 - \alpha) \boldsymbol{L} \tag{4.2}$$

where $\boldsymbol{L}$ is the cluster similarity score between the two peaks in a single run (described below), and α ($0 \leq \alpha \leq 1$) is a parameter controlling the relative influence of the two components. To compute $\boldsymbol{L}$, we require related peak groupings from the two runs being aligned. This takes the form of an $N_A \times N_A$ matrix $\mathbf{C}^A$ for run A and an $N_B \times N_B$ matrix $\mathbf{C}^B$ for run B. Entries in $\boldsymbol{C}^A$ and $\boldsymbol{C}^B$ can be either binary (0, 1) or probability values, depending on the peak grouping algorithm used. For example, if a greedy clustering approach is applied to the features in run A, the ij-th element of $\mathbf{C}^A$ will be either 1 or 0, depending on whether the i-th and j-th features (peaks) in A are clustered together (1) or not (0). Note that in the following, we define the diagonal components of both matrices to be zero to avoid double counting. We then compute $\mathbf{L}$ as follows:

$$\boldsymbol{L} = \boldsymbol{C}^A \cdot \boldsymbol{W} \cdot \boldsymbol{C}^B. \tag{4.3}$$

The resulting matrix gives cluster similarity scores such that each element L_{ij} of $\boldsymbol{L}$ is the sum of weight from peaks in the same cluster as i in run A to peaks in the same cluster as j in run B. This allows us to use the matrix $\boldsymbol{L}$ to upweight the similarity scores between peaks in the same cluster in one run that also have more potential matches to peaks in the same cluster in the other run of the matching. Computation of Equation 4.3 is illustrated in Figure 4.1. The ratio parameter α controls how much clustering information we bring into the overall similarity score matrix $\boldsymbol{S}$, with its value bounded in $0 \leq \alpha \leq 1$. Setting $\alpha = 1$ results in a matching that uses only information from $\boldsymbol{W}$, the similarity score matrix. Setting $\alpha = 0$ means that the matching is performed based only on the cluster similarity score $\boldsymbol{L}$. From our experience, a reasonable range of values for α lies between 0.2 to 0.4.

Our proposed approach is independent of the method used to group IP peaks in each run. For comparison, we call our method that does not use the cluster similarity score ($\alpha = 1$) to be Maximum-Weighted (MW). We then demonstrate the performance improvement from incorporating IP peaks information using two different clustering algorithms: a greedy RT clustering approach (described in Section 4.3.3) and a statistical mixture model (Section 4.3.4). The combination of matching with the greedy clustering is called MWG, while the alternative approach that uses the probabilities coming from the mixture model is called MWM.

4.3.3 Greedy Clustering of IP peaks

In the greedy clustering method, the most intense peak in the dataset is selected and clustered with other candidate peaks inside a retention time window g_{tol}. The next most intense peak that has not already been clustered is processed, and the grouping process is repeated until all peaks are exhausted. If chromatographic peak shapes information is available (such as for the Metabolomic dataset used in section 4.5.2), the Pearson correlation coefficient between

the chromatographic peak signals of the most intense peak and the candidate peaks are computed. Only candidate peaks with Pearson correlation values greater than some threshold c are accepted into the newly-formed cluster. This greedy clustering process results in binary grouping matrices $\mathbf{C}^A$ and $\mathbf{C}^B$ that can be used in eq. 4.3.

4.3.4 Mixture Model Clustering of IP peaks

We can also group IP peaks together by their RT values using a mixture model. Our observation consists of a vector of N observed peak's RT values $\mathbf{y} = (y_1, y_2, ..., y_n)$, and our aim is to partition each set of peaks into K groups of IP peaks (clusters) by their RT values. We used a Gaussian mixture model with Dirichlet Process prior (described further in Section 3.3) to model the data. A peak is indexed by the variable $n = 1, ..., N$ and a cluster indexed by the variable $k = 1, ..., K$. Each Gaussian mixture component has some mean μ_k are assumed to have a fixed precision (inverse variance) δ, corresponding to the fixed retention time tolerance for each group of IP peaks. Let the indicator $\boldsymbol{z}_{nk} = 1$ denotes the assignment of peak n to RT cluster k. Then:

$$\begin{aligned} \boldsymbol{\pi}|\alpha &\sim GEM(\gamma) && (4.4)\\ \boldsymbol{z}_{nk} = 1|\boldsymbol{\pi}_k &\sim \boldsymbol{\pi}_k && (4.5)\\ \mu_k|\mu_0, \tau_0 &\sim \mathcal{N}(\mu_k|\mu_0, \tau_0^{-1}) && (4.6)\\ y_n|\boldsymbol{z}_{nk} = 1, \mu_k &\sim \mathcal{N}(\mu_k, \delta^{-1}) && (4.7) \end{aligned}$$

where $\boldsymbol{\pi}$ is the mixing proportions, distributed according to the GEM (Griffiths, Engen and McCloskey) distribution. The GEM distribution over $\boldsymbol{\pi}$ is parameterised by the concentration parameter γ and is described through the stick-breaking construction:

$$\begin{aligned} \beta_k &\sim Beta(1, \gamma) && (4.8)\\ \boldsymbol{\pi}_k &= \beta_k \prod_{l=1}^{k-1}(1 - \beta_l) && (4.9) \end{aligned}$$

The mixture component mean μ_k is drawn from a base Gaussian distribution with mean μ_0 and precision τ_0. We set μ_0 to the mean of the observed data, while τ_0 is set to a broad value of 5E-3. Analytical inference is not tractable here, so we use the Gibbs sampling scheme for inference. To do this, we need the conditional probability of $p(\boldsymbol{z}_{nk} = 1, ...)$ of peak n to be in an existing cluster k (or k^* if a new cluster is to be created), given any other parameters in

the model. This conditional probability is given by:

$$P(\boldsymbol{z}_{nk} = 1|\mathbf{y}_n, \ldots) \propto \begin{cases} c_k \cdot p(\mathbf{y}_n|\boldsymbol{z}_{nk} = 1, ...) \\ \gamma \cdot p(\mathbf{y}_n|\boldsymbol{z}_{nk^*} = 1, ...) \end{cases} \tag{4.10}$$

where c_k is the current number of members (peaks) in an existing cluster k. $p(\mathbf{y}_n|\boldsymbol{z}_{nk} = 1, ...)$ is the likelihood of peak $\mathbf{y}_n$ in an existing cluster k. We can marginalise over all mixture components and get:

$$p(\mathbf{y}_n|\boldsymbol{z}_{nk} = 1...) \quad = \quad \mathcal{N}(\mathbf{y}_n|\mu_k, \lambda_k^{-1}) \tag{4.11}$$

where $\lambda_k = ((\tau_0 + \sigma c_k)^{-1} + \delta^{-1})^{-1}$ and $\mu_k = \frac{1}{\lambda_k}\left[(\mu_0\tau_0) + (\delta \sum_n \mathbf{y}_{n\in k})\right]$. Here, $\mathbf{y}_{n\in k}$ denotes the RT values of any peak n currently assigned to cluster k, and c_k the count of such peaks. The conditional probability of peak n to be in a new cluster k^* is:

$$p(\mathbf{y}_n|\boldsymbol{z}_{nk^*} = 1...) \quad = \quad \mathcal{N}(\mathbf{y}_n|\mu_0, \lambda_{k^*}^{-1}) \tag{4.12}$$

where $\lambda_{k^*} = (\tau_0^{-1} + \sigma^{-1})^{-1}$. In a step of the Gibbs sampling procedure, we perform the assignment of peak n to cluster k, creating new cluster k^* if necessary. Using the posterior summaries across all samples drawn $S^* = \frac{1}{R}\sum_{r=1}^{R} s_r$, where s_r is the r-th posterior sample collected after a suitable burn-in period and R is the total number of samples taken (excluding burn-in samples), we can obtain the marginal posterior of the probability of two features (peaks) to be in the same cluster k averaged across all samples. These probabilities comprise the elements of $\mathbf{C}^A$ and $\mathbf{C}^B$ (i.e. the ij-th element of $\mathbf{C}^A$ is the proportion of samples from run A in which peaks i and j were in the same cluster), which can be used in eq. 4.3.

4.4 Evaluation Study

In this chapter, the performance of the proposed methods and other benchmark methods is evaluated using precision and recall on LC-MS datasets from proteomic, metabolomic and glycomic experiments. The proteomic datasets are obtained from [33] while the glycomic dataset comes from [1]. These datasets provide the ground truth for alignment and have used to benchmark alignment performance in other evaluation studies [33, 19, 35, 36, 1]. Additionally, we also introduce a metabolomic dataset generated from the standard runs used for the calibration of chromatographic columns [54]. The runs were produced from different LC-MS analyses separated by weeks, representing a challenging alignment scenario.

Many direct matching methods work in a pairwise fashion and produce an overall results via some merging strategies of intermediate results. Pairwise performance therefore limits

overall performance, and as such, in this chapter, we focus on evaluation using only pairs of runs. Some (P2, metabolomic and glycomic) of the datasets selected for evaluation in our experiments have more than 2 runs, so we select only 2 runs each to form a training and testing set. The procedure for doing so is described in the respective following sections for each dataset.

4.4.1 Proteomic Datasets

[33] introduces two benchmark LC-MS proteomic sets (P1, P2) constructed to evaluate the ability of alignment tools in dealing with large retention time drifts. Both the P1 and P2 datasets were analysed using an automated LC-LC/MS-MS platform. Each dataset comes in multiple chromatography salt-step fractions, obtained by bumping the salt level at every 10 minutes interval during chromatographic separation. P1 was produced from *E. coli* samples digested by trypsin, and comes in 2 runs for each fraction. P2 was obtained from *M. smegatis* protein extracts, similarly digested by trypsin, and contains 3 runs for each fraction. P2 was constructed to be a greater challenge to align with runs separated by weeks. Alignment ground truth is established in [33] by means of peptides that can be reliably identified during the identification stage. Only identification annotations with SEQUEST Xcorr score >1.2 is included. Annotations are then filtered by their retention times and matched across runs.

For the proteomic datasets, each fraction in P1 has two runs used for alignment, while each fraction in P2 has three runs (we use only the first two to establish pairwise alignments). Tables 4.1 shows the number of features for each run of the P1 and P2 datasets used for evaluations. Both P1 and P2 represent challenging alignment cases, with large deviations in RT values across runs. This is especially true for P2 with LC-MS runs separated by weeks and large differences in the number of features per run. Further details on the nature of the datasets can be found in [33].

4.4.2 Metabolomic Datasets

We use a metabolomic dataset generated from a mixture of 104 standard metabolites used for the calibration of chromatographic columns (details in [54]). These runs were produced by ZIC-HILIC chromatography (Merck Sequant, Darmstadt, DE) on an UltiMate 3000 RSLC system (Thermo, Hemel Hempstead, UK), coupled to an Orbitrap Exactive mass spectrometer (Thermo, Hemel Hempstead, UK) in positive mode. The metabolomic dataset is available in different 11 runs, produced from different LC-MS analyses separated by weeks. While these runs are not true technical replicates, they are similar enough to be treated as replicates for the purpose of performance evaluation, and they represent a realistic and fairly challenging alignment scenario. The output from each of these runs is available in PeakML format,

Fraction	# runs	# features per run (P1)	# features per run (P2)
000	2	5824	5054
		4782	5100
020	2	1114	3271
		1021	529
040	2	1230	1483
		958	678
060	2	1902	-
		1440	-
080	2	1183	474
		903	438
100	2	745	401
		581	429

Table 4.1: No. of features in the proteomic (P1 and P2) datasets. Note that fraction 060 is not present in P2.

which were then converted into a suitable format using the mzMatch suite [48]. Both the original PeakML files and the converted text files can be found in our site. To generate the actual training and testing sets, 30 randomly pairs of runs were extracted as training sets, and another 30 pairs of runs extracted for testing sets. Table 4.2 shows the number of features in each run of the metabolomic dataset.

Metabolomic Run	**# features**	**Metabolomic Run**	**# features**
1	4999	7	6319
2	4986	8	4101
3	6836	9	5485
4	9752	10	5034
5	7076	11	5317
6	4146		

Table 4.2: No. of features in the full metabolomic dataset

Alignment ground truth was constructed from the putative identification of peaks in each of the 11 runs separately at 3 ppm using mzMatch's Identify module, taking as additional input a database of 104 compounds known to be present and a list of common adducts in positive ionisation mode (Table 4.3). This is followed by matching of features that share same annotations across runs to construct the alignment ground truth. Only peaks unambiguously identified with exactly one annotation are used for this purpose, as peaks with more than one annotations per run are discarded from the ground truth construction. The results from this process is an alignment ground truth for a smaller subset of peaks in the runs that can be reliably identified at high mass precision. Note that constructing alignment ground truth in this manner does not introduce bias to the ground truth as the identification information is not used during the alignment stage.

Adduct Types			
M+2H	M+H	M+ACN+H	M+H+NH4
M+NH4	M+ACN+Na	2M+ACN+H	M+ACN+2H
M+Na	M+2ACN+H	M+2ACN+2H	M+CH3OH+H
2M+H			

Table 4.3: List of common adduct types in positive ionisation mode for ESI.

4.4.3 Glycomic Dataset

[1] provides a glycomic dataset containing 23 runs, produced from untargeted LC-MS study for identifying N-glycan disease biomarkers in glyomics studies. LC-MS data were acquired from a Dionex 3000 Ultimate nano-LC system, coupled to an LTQ-Orbitrap Velos mass spectrometer on positive mode. Alignment ground truth is established in [1] based on a manual comparison of measured mass values with theoretical values (taking into account hydrogen adducts) and visual inspection of potentially incorrect assignments. We randomly extracted 30 pairs of runs for training and another 30 pairs of runs for testing performance evaluation from the full glyocomic dataset provided by [1], which comes in 23 runs in total. The following tables show the number of features in each run and the indices of the pairs of files randomly selected as training and testing sets in our Glycomic experiment.

Glycomic Run	**# features**	**Glycomic Run**	**# features**
1	856	13	911
2	1088	14	1144
3	922	15	932
4	808	16	1541
5	886	17	1022
6	850	18	1051
7	979	19	1119
8	1008	20	1047
9	904	21	1017
10	1043	22	990
11	1041	23	977
12	885		

Table 4.4: No. of features in the full glycomic dataset from [1]

4.4.4 Experimental setup

The alignment tools evaluated have in common user-defined mass-to-charge ratio (m/z) and retention time (RT) window parameters. These parameters act as hard thresholds that determine the solution space to be explored in the m/z and RT dimensions when matching features. Performance of all alignment procedures is highly dependent on the assumptions

and choice of parameter values that underpin them [42]. For example, warping methods must make assumptions regarding the mathematical form of the warping function and are dependent on a good choice of reference run. Direct matching approaches typically need to decide on the form of peak similarity function, and define some m/z and RT windows, outside of which, peaks cannot be matched. Whilst the m/z window and parameters can often be determined based on the mass accuracy of the measurement equipment, there is no obvious way to determine the RT window and associated parameters. The optimal choice of such parameters could have a significant influence on the final results [42], and there is no reason to believe that these parameters should remain constant across different experiments.

Previous studies that use the proteomic dataset presented here [33, 35, 36] varied the window parameters and reported the best performance achieved. Whilst informative, this procedure is unrealistic due to the role of the ground truth in choosing the optimal parameter values. To provide a more realistic estimate of performance, we also present the performance on a separate testing set. In other words, we optimise the window parameters on one alignment task and report the performance when using these optimised parameters on a second task (distinct from the first task). This reflects the scenario where the parameters are set based on performance on a previous dataset or due to information supplied from the instrument manufacturer and tells us how critical setting these parameters is for each method.

In this chapter, *training set* refers to the data on which alignment parameters are optimised and *testing set* refers to the independent set on which alignment performance is evaluated. We believe that this represents a more realistic measure of alignment performance and provides us with some information as to how the different algorithms generalise to new datasets. We addressed the lack of comparative evaluation of alignment tools as discussed in [42] by independently reproducing key results from [33] and [36] for the Join and SIMA alignment methods. Our evaluation studies were performed on proteomic, metabolomic and glycomic datasets introduced before to validate the hypothesis that using related-peak information can improve alignment performance. Since most direct matching algorithms work in a pairwise fashion (pairs of runs are matched and the results combined), pairwise performance therefore limits overall performance, justifying the choice for our experiments. For the proteomic datasets, each fraction in P1 has two runs used for alignment, while each fraction in P2 has three runs (we use only the first two to establish pairwise alignments). Similarly for the metabolomic and glycomic datasets, we randomly extracted 30 pairs of runs for training and another 30 pairs of runs for testing performance evaluation.

Performance is evaluated in terms of precision, recall and F_1-score. Looking at pairwise matching, we can define the following positive and negative instances with respect to some pairwise alignment ground truth:

- True Positive ($\boldsymbol{TP}$): pairs of peaks that should be aligned and are aligned.

- False Positive ($\boldsymbol{FP}$): pairs of peaks that should not be aligned but are aligned.
- True Negative ($\boldsymbol{TN}$): pairs of peaks that should not be aligned and are not aligned.
- False Negative ($\boldsymbol{FN}$): pairs of peaks that should be aligned but are not aligned.

In the context of alignment performance, precision ($\frac{\boldsymbol{TP}}{\boldsymbol{TP+FP}}$) is the fraction of aligned pairs in the output that are correct with respect to the ground truth, while recall ($\frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}}$) is the fraction of aligned pairs in the ground truth that are aligned in the output. A perfect alignment would have both precision and recall to be 1. In addition, we also computed the F_1 score (the harmonic mean of precision and recall) such that $F_1 = 2(precision \cdot recall)/(precision + recall)$. Only feature pairs present in the ground truth are considered for evaluation. The idea of using pairwise matching to define alignment performance evaluation is not new, and has also been done in [38]. Collectively for the purpose of performance evaluation, the set of Precision, Recall and F_1 values is referred to as a 'measurement'.

4.4.5 Other Alignment Tools For Comparison

Our proposed approach was benchmarked against MZmine2's Join Aligner [19] and SIMA [36]. Our own matching method (MW) also serves as a useful baseline to demonstrate any difference in performance with or without using clustering information. The two benchmark tools employ different approaches towards alignment. Join Aligner is a greedy direct-matching method, while SIMA is a combinatorial direct-matching method, with an optional warping step to correct RT shifts after an initial matching has been established. Users of the MZmine2's toolkit may have good reasons to prefer Join Aligner to the more recent RANSAC Aligner due to its simplicity and speed. Join Aligner produces a deterministic alignment output (so running it each time on the same input and parameters gives the same result), in contrast to the RANSAC aligner, which is non-deterministic. Join Aligner has relatively few parameters to configure, the most important ones being the *m/z tolerance* and *retention time tolerance* parameters. These parameters are used for thresholding and score calculations, and they were varied within reasonable ranges during our experiments. Similarly, the two most important parameters used in SIMA for thresholding and computing feature similarities are the $T_{(m/z)}$ and T_{rt} parameters (equivalent to our σ_m and σ_t). We let these two parameters vary in our experiments. SIMA also offers an optional step to correct for retention time distortion by constructing a smooth and monotonic warping function for the maximum likelihood alignment path after the initial matching has been done. The utility of this optional step is not obvious to end-users, since it requires additional parameters to configure and relies on having an initial correspondence established. Therefore, we chose to test only the core matching functionality in SIMA.

4.4.6 Parameter Optimisation

For every evaluated method in our experiments, we performed grid-search on the m/z and RT windows parameters using the training set. We then used those optimal parameters to perform alignment on the testing set, giving us the respective performance measures (Precision, Recall, F_1) on the testing set. For testing set consisting of multiple fractions, we report the average performance measures on the testing fractions.

For training using the P1 and P2 datasets in the proteomic experiments, the m/z and RT tolerances were varied within: $\{1.0, 1.2, ..., 2.0\}$ for the m/z tolerance, and $\{5, 10, .., 300\}$ seconds for the RT tolerance. The parameter ranges were chosen based on reasonable estimates of the instrument's precision and prior RT tolerance values as reported by [33]. We kept all the default values for the remaining parameters in each evaluated tool, if any. For MWG, we also varied the ratio parameter α from $\{0.1, 0.2, ..., 1.0\}$ and the grouping parameter g_{tol} from $\{1, 2, ..., 10\}$ seconds and uses the combination that results in the best performance. For MWM, the ratio parameter α was varied from $\{0.1, 0.2, ..., 1.0\}$ but mixture model parameters were kept the same for clustering of all fractions in P1 and P2. When clustering all fractions in a dataset, a broad Gaussian prior was set for the component mean μ_j of each cluster j. The component precision s_j was set to 5 seconds, while the DP concentration parameter γ is set to 1. We drew 2000 posterior samples (with 1000 initial burn-in samples) for each run during the Gibbs sampling steps to construct the probability matrix of peak-vs-peak to be in the same cluster.

For the Metabolomic and Glygomic experiments, 30 pairs of run were randomly extracted from the M1 metabolomic dataset in [33] and from the glycomic dataset in [1]. These were assigned to be the training sets. Another 30 pairs of runs were extracted from each dataset to be the testing sets. Each pair of runs in the training set is assigned a partner pair of runs in the testing set. Parameters were optimised on pairwise runs in the training set and performance evaluated on the assigned partner runs in the testing set. For both datasets, the m/z tolerances used were $\{0.05, 0.1, 0.25\}$ and RT $\{5, 10, 15, ...100\}$ seconds. These ranges of parameters were selected in view of instrument accuracy and RT noise level of the LC-MS instruments that generate our metabolomic data and in [1]. The ratio parameter α was from $\{0.1, 0.2, ..., 1.0\}$ and the grouping parameter g_{tol} from $\{2, 4, ..., 10\}$ seconds for both datasets, and for the metabolomic dataset where chromatographic peak shapes information is available and used for greedy clustering in MWG, the threshold for the Pearson correlation coefficient between peak shape signals was varied from $c = \{0.70, 0.75, 0.80, 0.85, 0.90, 0.95\}$.

4.5 Results and Discussions

4.5.1 Proteomics Experiments

Single-fraction Experiment

Both P1 and P2 data consist of multiple fractions. In the first experiment, we investigate the best possible performance by using the same fraction as training and testing sets. As described in Section 4.4.6, on each training set (a fraction), we optimised the m/z and RT window parameters for alignments. The m/z parameters are in parts per million, normally notated 'ppm' and the range of m/z parameters used were $\{1.0, 1.1, ..., 2.0\}$ and RT $\{5, 10, ..., 300\}$ seconds. Parameters that control the grouping and influence of the cluster similarity score for our MWG and MWM methods were also optimised. The ratio parameter α was set to $\{0.1, 0.2, ..., 1\}$ for both MWG and MWM. The grouping tolerance g_{tol} was set to $\{1, 2, ..., 10\}$ seconds for greedy clustering, while the same hyperparameters were used for clustering of all fractions in case of mixture-model clustering (further details on parameter range selections are in Section 4.4.6).

The results are shown in Tables 4.5 and 4.6. We see that approximate maximum weighted matching (MW) alone performs competitively to other tools. On the P1 data (Table 4.5), incorporating grouping information (MWG, MWM) improves F_1 score performance over MW. MWG outperforms MWM, which may be due to the fact that the greedy approach is easier to optimise. For the P2 data (Table 4.6), which contains features with significantly higher RT drift across runs, again we find that MW is competitive and clustering information (MWG) improves performance for all fractions. The results here show the potential of our proposed approach: any peak grouping results expressed in a suitable matrix format can be incorporated into our method, and used as additional information during the matching stage. Figures 4.2 and 4.3 show how the benefit of incorporating clustering information is realised during matching: it allows the matching methods to explore regimes in the solution space having higher precision and recall values. On some training fractions, both methods that incorporate clustering information show significant increases in the best possible F_1 score. For dataset P1 fraction 000, this is an 11%-improvement for MWG and a 7.5%-improvement for MWM. For dataset P2 fraction 100, this is a 51%-improvement for MWG and 25%-improvement for MWM. Smaller improvements can be observed from other fractions in the Proteomic datasets too.

Multiple-fractions Experiment

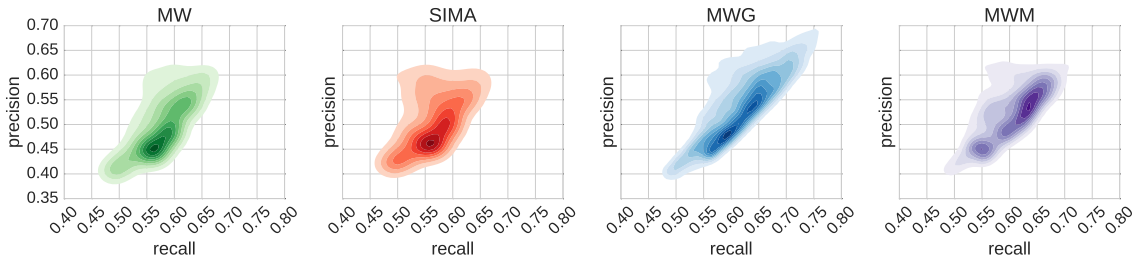
The single-fraction experiment does not represent a very realistic scenario as the optimal parameters were determined with respect to an alignment ground truth; practitioners might

Fraction	Join	SIMA	MW	MWG	MWM
000	0.63	0.64	0.64	**0.77**	0.71
020	0.88	0.88	0.88	**0.95**	0.90
040	0.82	0.83	0.85	**0.87**	0.86
060	0.76	0.78	0.78	**0.88**	0.83
080	0.90	0.89	0.88	**0.92**	0.90
100	0.89	0.89	0.89	**0.91**	0.91
Mean	0.81	0.82	0.82	**0.88**	0.85

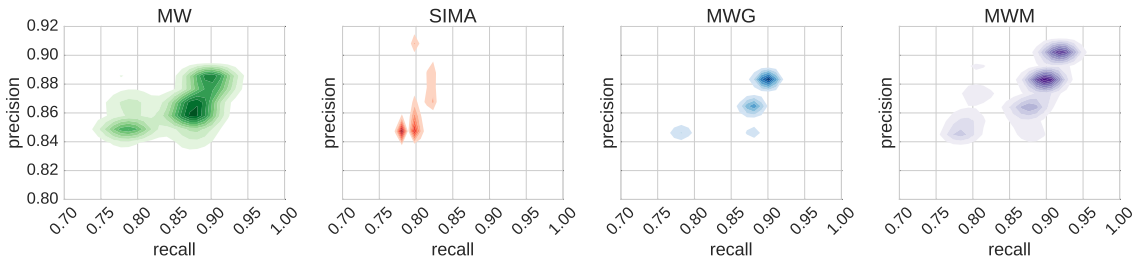
Table 4.5: F_1 scores for the single-fraction experiment results on the P1 dataset. The tool with the highest F_1 score for each fraction is highlighted in bold. The results for 'All' show the average F_1 scores of individual fractions.

Fraction	Join	SIMA	MW	MWG	MWM
000	0.45	0.45	0.45	**0.49**	0.45
020	0.77	0.78	0.79	**0.80**	0.79
040	0.77	0.78	0.77	**0.80**	0.77
080	0.66	0.68	0.67	0.67	**0.72**
100	0.55	0.58	0.56	**0.85**	0.70
Mean	0.64	0.65	0.65	**0.72**	0.69

Table 4.6: F_1 scores for the single-fraction experiment results on the P2 dataset. The tool with the highest F_1 score for each fraction is highlighted in bold. The results for 'All' show the average F_1 scores of individual fractions.

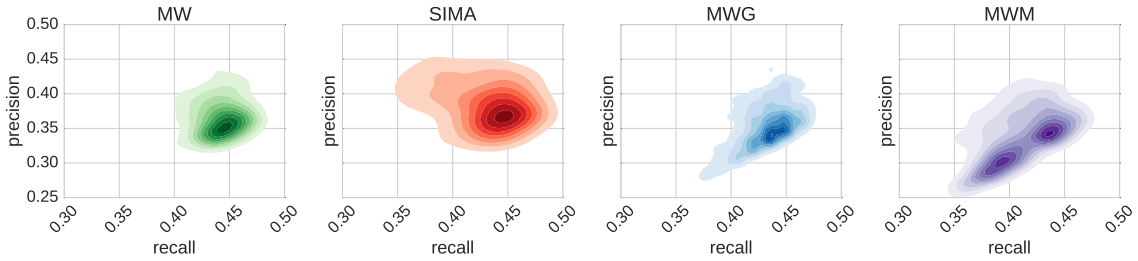


(a) P1 Fraction 000

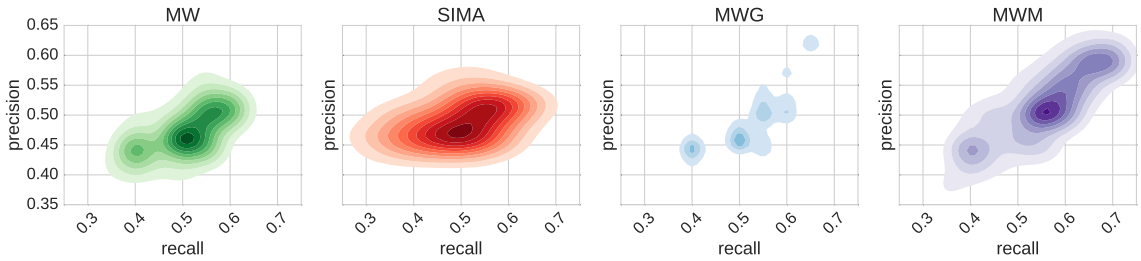


(b) P1 Fraction 100

Figure 4.2: Precision and recall training performance for all parameters (m/z, RT tolerance, α and g_{tol}) varied in the experiment for the fractions containing the most (a) and the least (b) number of features in the P1 dataset.



(a) P2 Fraction 000



(b) P2 Fraction 100

Figure 4.3: Precision and recall training performance for all parameters (m/z, RT tolerance, α and g_{tol}) varied in the experiment for the fractions containing the most (a) and the least (b) number of features in the P2 dataset.

not possess that information in real analytical situations. In this experiment, we improved upon the single-fraction experiments by using each fraction in each dataset as the training set and the remaining fractions as the testing set. Parameters were optimised on the training set and performance evaluations were performed on the testing set. This training-testing procedure produces 6 measurements for P1 and 5 measurements for P2, corresponding to the number of training fractions in each dataset. The overall F_1 score reported for each measurement is the average F_1 scores from individual testing fractions. The aim of this experiment is to investigate how well the different methods generalise to data that may have slightly different characteristics from that used to optimise the parameters – i.e. how critical the particular parameter values are.

Tables 4.7 and 4.8 show the F_1 score across measurements. On P1, the best overall performance is achieved by our methods that incorporate clustering information into alignment (MWG, MWM). On P2, the results are less homogeneous, with no method consistently performing best on all the different testing fractions. In the case of the noisiest data (dataset P2 fraction 000), our proposed approach incorporating greedy clustering (MWG) shows a decrease in overall testing performance instead. This is because the greedy clustering method used is sensitive to the choice of parameters and do not generalise well across the different fractions of P2. For instance, the best MWG's grouping tolerance parameter for fraction 000 is 5 seconds, while it is 1 second for fraction 080. The results suggest the dependence of our methods on the quality of groupings of IP peaks in order to generalise well on different runs.

The heterogeneous testing performance in the multiple-fractions experiment of P2 shows that no method performs best and the choice of optimal parameters that work for certain runs do not generalise well to others on datasets with very high RT variability.

Training Frac.	Testing Performance				
	Join	**SIMA**	**MW**	**MWG**	**MWM**
000	0.82	0.85	0.82	**0.86**	**0.86**
020	0.78	0.76	0.78	**0.79**	0.75
040	0.78	0.76	0.77	0.79	**0.81**
060	0.78	0.78	0.77	**0.84**	0.83
080	0.71	0.73	0.72	0.77	**0.78**
100	0.75	0.77	0.74	0.76	**0.78**

Table 4.7: Multiple-fractions experiment results for the P1 dataset. For each training fraction, the reported testing performance is the average of individual F_1 scores from the testing fractions. The top-performing method (highest F_1 score) is highlighted in bold.

Training Frac.	Testing Performance				
	Join	**SIMA**	**MW**	**MWG**	**MWM**
000	0.62	**0.64**	0.61	0.48	0.61
020	**0.58**	0.56	0.55	0.43	0.55
040	0.52	**0.56**	**0.56**	0.41	**0.56**
080	0.56	0.50	0.50	0.50	**0.57**
100	**0.63**	0.57	0.56	0.44	0.57

Table 4.8: Multiple-fractions experiment results for the P2 dataset. For each training fraction, the reported testing performance is the average of individual F_1 scores from the testing fractions. The top-performing method (highest F_1 score) is highlighted in bold.

4.5.2 Metabolomic and Glycomic Datasets

We further explore the performance of our proposed methods on the metabolomic and glycomic datasets. From the full dataset, we randomly extracted 30 pairs of runs as the training sets and another 30 pairs of runs as the testing sets. Each training set is paired to a testing set. Parameters were optimised on the training set and the best attainable performance reported as the training performance. Generalisation performance is evaluated on testing sets using the optimal parameters from the training stage.

Figures 4.4 and 4.5 summarise the results from the experiments. We see that all methods perform better on the glycomic set than on the metabolomic set. This is explained by the fact that the metabolomic runs represent a generally more challenging alignment scenario with significantly more features to align. MW performs identically to SIMA on both datasets due to the similar form of Mahalanobis distance function used. This is despite the differences

in the actual matching method that establishes feature correspondences in SIMA and MW, emphasising the fact that the actual choice of matching function might be less important than other factors, such as the determination of similarity scores between peaks. On the glycomic dataset, adding clustering information improves the training performance, with an increase in the mean of the F_1 scores across 30 measurements from 0.89 (MW) to 0.93 (MWG) and 0.92 (MWM). This also translates into statistically significant improvements on the testing sets for both MWG (p=0.01, paired t-test) and MWM (p=0.002, paired t-test) over MW.

On the metabolomic dataset, where it is potentially harder to produce good clustering results due to the larger number of peaks and the more complex elution profile, we observe improvements in the mean of the F_1 scores from 0.83 (MW) to 0.90 (MWG) and 0.85 (MWM) on the training sets. These are also statistically significant improvements for both MWG ($p<0.001$, paired t-test) and MWM ($p<0.001$, paired t-test) over MW. The training results confirm our hypothesis that indeed incorporating clustering information (by modifying the similarity matrix used for matching in the proposed manner) can be used to help improve matching results over the case when such information is not used. However, this does not translate into any statistically significant improvements on the testing sets, suggesting that for the metabolomic dataset evaluated here, our proposed methods are also sensitive to parameter choices, and the choices of particular parameters (especially for the clustering step) that work on some runs may not generalise well to others. The results shown for running MWG on the metabolomic data in Figures 4.4 and 4.5 takes into account the Pearson correlations of the chromatographic shapes between peaks during the clustering process, since that information is available and straightforward to incorporate into the greedy clustering process. Results for MWG that consider only the RT values are presented and discussed in the following paragraph.

We also compared the results for MWG on both the training and testing sets on the standard metabolomic dataset when the greedy grouping is performed using only RT information (MWG (RT)) and when chromatographic peak shape correlations are also considered (MWG(RT+PS)) during the grouping process. Statistically significant differences can be observed on the training performance of Figure 4.6, with the mean of F_1 scores for MW 0.83, MWG(RT) 0.88 and MWG(RT+PS) 0.90. However, this does not translate to any improvements on the testing sets, with the mean of F_1 scores for MW 0.86, MWG(RT) 0.83 and MWG(RT+PS) 0.85. Introducing clustering information when only RT information is used during the clustering process (MWG(RT)) reduces testing performance. The training results suggest that where additional information such as chromatographic peak shapes are available, they should be used for the clustering step in the proposed methods. However, the lack of any statistically significant testing improvements between MW and MWG (RT+PS), suggest that the optimal parameters from training runs do not generalise well to different testing runs for the greedy clustering approach in general, especially for complex metabolomic runs,

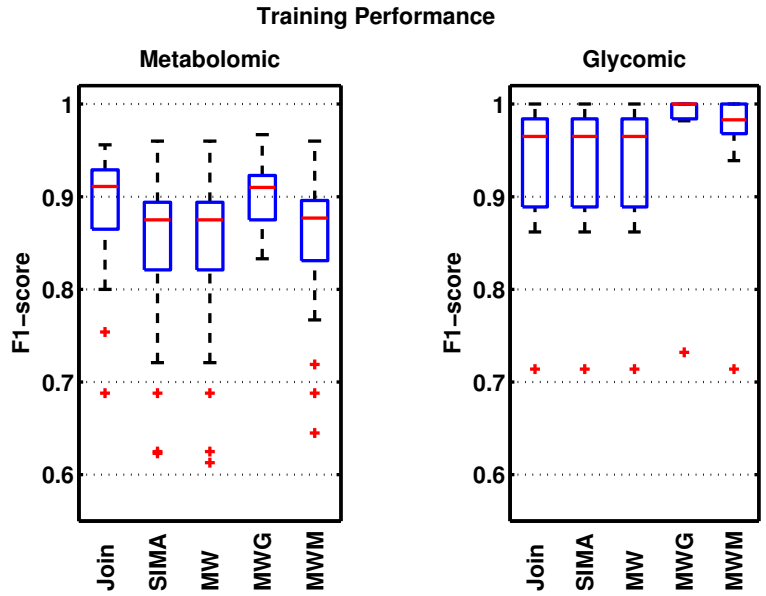


Figure 4.4: Training performance shows the best F_1 scores obtained by each method on 30 pairs of randomly-selected metabolomic and glycomic training sets.

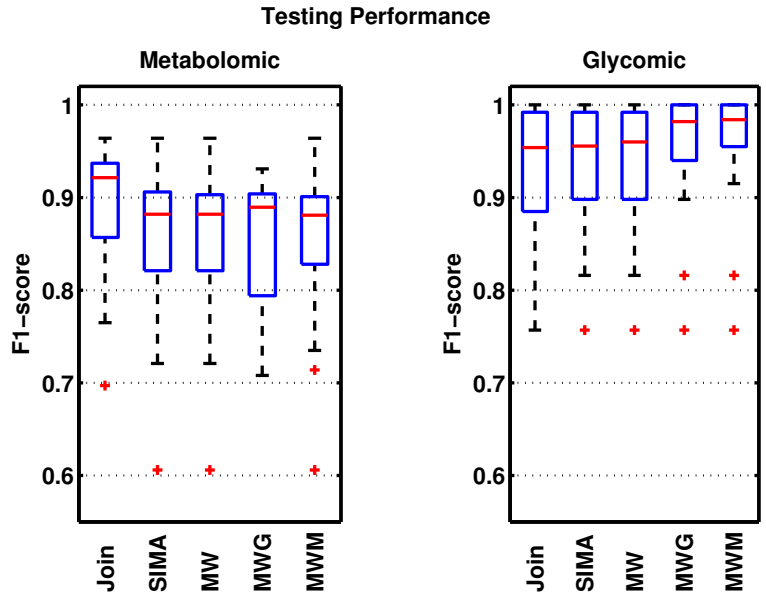


Figure 4.5: Testing performance shows how well each method generalise on the 30 different testing sets, each evaluated using the optimal training parameters from its corresponding training set.

with large number of features that tend to closely co-elute with each other.

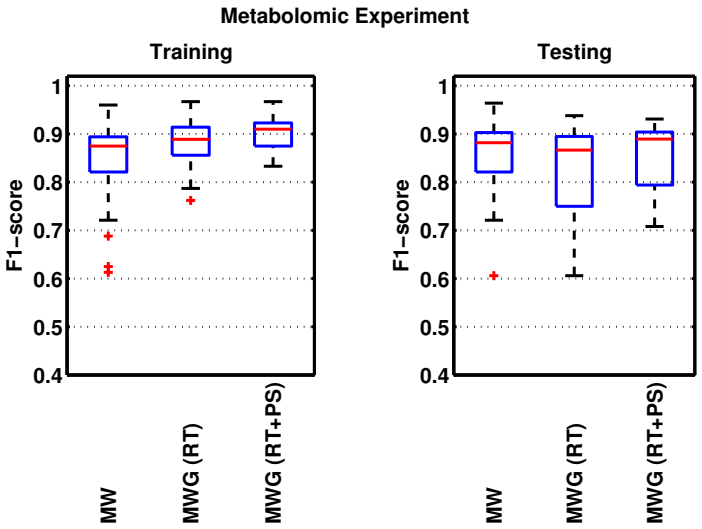


Figure 4.6: Comparisons in matching performance when greedy clustering with retention time (MWG(RT)) and peak shape correlations (MWG(RT+PS)) are used.

4.5.3 Running Time

Computational times of the proposed methods are primarily affected by the number of features in the runs being aligned and to some extent, the thresholding parameters used during similarity score computation and feature matching. Table 4.9 reports the measured running time for each proposed method using the parameters that give the best training performance). For each fraction being aligned, the running times were measured three times on a standard laptop with Intel Core i5 CPU running at 2.5 GHz, and the average value reported for matching only (MW), matching incorporating greedy clustering (MWG) and matching incorporating mixture model clustering (MWM). The time complexity of the mixture-model clustering step in MWM is $O(N)$ where N is the number of features in the run being clustered. We took 2000 posterior samples, discarding the first 1000 samples during the burn-in period. The number of samples were chosen to ensure convergence to the stationary distribution during inference.

4.6 Conclusion

In this chapter, we have proposed a novel peak matching method that incorporates related peak information to improve alignment performance. The method takes related peak information in the form of peak-by-peak binary or real-valued similarity matrices and as such is independent of the particular method used to compute these. The method fits into the

Fraction	Total Features	MW	MWG	MWM
000	10606	9	12	2700
020	2135	1	2	524
040	2188	2	2	540
060	3342	2	3	825
080	2086	2	2	505
100	1326	1	2	321

Table 4.9: Example measured execution time in seconds on fractions of the P1 dataset

category of direct matching approaches — those alignment methods that do not perform an explicit time-warping phase. Our experimental results demonstrate the potential of this approach. From the training results, we see evidence of performance improvement across all evaluated datasets by incorporating grouping information into the matching process in the proposed manner. With the exception of the metabolomic dataset, both the greedy and model-based clustering approaches evaluated in our experiments rely only on the RT information for grouping IP peaks. By looking at the testing performance, our results also explore the ability of the evaluated methods to generalise on different runs using less than optimal parameters. This is important because in the actual analytical situation of LC-MS data, neither the optimal parameters nor the alignment ground truth is known.

Note that our method relies on grouping of IP peaks, and this introduces additional user-defined parameters. However, as our experiments have shown, in some settings, it may be much easier to produce good groupings of IP peaks than accurately determine RT window parameters (the same grouping parameters were used for all evaluation datasets in the case of mixture-model clustering). Depending on the nature of the data, parameters relating to within-run characteristics (e.g. RT window for grouping IP peaks) may be more likely to generalise across runs and experiments than parameters relating to between-run characteristics (particularly RT). For example, changes in the liquid chromatography (LC) column would likely result in related-peaks still co-eluting but could significantly change the absolute RT.

It would be interesting to investigate in greater detail any performance improvements that can be obtained from using other peak grouping methods, such as [49] that uses a mixture model of peak shape correlations or [57] that considers the dependencies between adduct and isotopic peaks when clustering. Exploring alternative approximate matching algorithms (such as the scaling algorithm in [107], which provides a $(1-\epsilon)$ approximation of the maximum weighted matching in optimal linear time for any ϵ) and evaluating the benefits of incorporating different clustering approaches into our proposed alignment method are avenues for future work. Finally, the different alignment methods evaluated in this chapter also suffer from variable behaviours depending on the order of the runs being aligned [14]. This is particularly true in the case of alignment of multiple runs (typical in large-scale LC-MS

studies), where the final alignment results are often constructed through merging of intermediate alignments of pairwise runs. Different alignment methods may employ a different merging approach, for example, Join merges the intermediate results towards a reference run while SIMA allows the possibility of using a greedy hierarchical merging scheme. Systematic evaluation on how the chosen merging scheme may influence alignment performance is beyond the scope of this chapter and is an item for future work.

A limitation of the proposed approaches lies in the fact that the clustering of IP peaks are performed based on RT only. The valuable information present in the m/z domain is not used for clustering. The grouping of IP peaks based on their m/z information is less straightforward as peaks that are related (sharing close RT values) do not necessarily have m/z values that are close to each other. In the evaluation on the complex metabolomic dataset, we observe that the proposed approach using RT clustering manages to improve training performance (due to overfitting) but fails to produce any statistically significant improvement in the testing performance due to its limited generalisation ability. In the next chapter, we address this issue by focusing specifically on metabolomics and proposing a clustering model that explicitly takes into account the chemical relationship between IP peaks in LC-MS-based metabolomics.

Chapter 5

Precursor Clustering of Ionisation Product Peaks

5.1 Introduction

Chapter 4 explores the idea of using the clustering of ionization product (IP) peaks to modify the similarity scores used for matching with the aim of improving alignment results. However, the MWG and MWM methods in Chapter 4 lies on clustering based on the retention time alone. Valuable information present in the mass-to-charge (m/z) domain and also in the chemical relationships of IP peaks is not used for clustering. In this work, we extend upon the methods in the previous chapter and propose a novel Bayesian mixture model (PrecursorCluster) to cluster IP peaks based on m/z and RT values. The key difference from the mixture model RT clustering introduced in the previous chapter lies in how PrecursorCluster uses a set of user-defined transformation rules to relate peaks to a common precursor mass, allowing IP clusters to be formed through the grouping of peaks that share chemically meaningful relationships.

Building upon the clustering results returned by PrecursorCluster, two alternative alignment methods (illustrated in Figure 5.1) are introduced for aligning IP clusters across runs: **(i)** Cluster-Match, a fast direct-matching method of IP clusters that uses the posterior precursor mass and RT values of IP clusters to compute the approximate maximum-weighted matching of the IP clusters and **(ii)** Cluster-Cluster, a second-stage clustering model that constructs alignment by means of grouping IP clusters according to their likelihood of being assigned to the same top-level cluster (corresponding to metabolites shared across all runs). In this manner, IP clusters assigned to the same top-level cluster are considered to be matched. The actual alignment between their member peaks are established by grouping member peaks that share the same IP type. The Bayesian approach in Cluster-Cluster also allows us to incorporate additional information for alignment in a principled manner by adding likelihood

terms. As an example, we illustrate this in Cluster-Cluster by including a likelihood term on the different adduct types of IP peaks assigned to an IP cluster. This allows IP clusters to be placed in the same top-level cluster — and correspondingly having their member peaks matched — only if the characteristic adduct 'signatures' of IP clusters are similar.

The aim of this chapter is to evaluate whether through the proposed methods, the matching of IP clusters can improve upon the matching of peaks alone. For the purpose of evaluations, two benchmark datasets of standard and beer mixtures, alongside their associated alignment ground truth and a list of 14 adduct transformations in positive ionization mode, were used. Using precision, recall and F_1-score as evaluation measures, the performance of the proposed method of matching IP clusters (Cluster-Match) were compared against the direct matching of peak features (MW) and its variant (MWG) in the previous Chapter 4 that modifies the similarity matrix used during matching to bring together group of peaks related by RT closer during matching. Additionally, the probabilistic matching results produced by Cluster-Cluster is also described, demonstrating that it is possible to use its output to extract aligned peaksets with varying degrees of confidence. Cluster-Cluster were evaluated with and without the adduct signature term to determine whether through the addition of that likelihood term, we can obtain better alignment results.

5.2 Related Work

It is suggested in [42] that the objective function used for alignment can be improved by operating on groupings of IP peaks rather than using individual peaks. In addition, [40] proposes minimising an objective function that uses groups of isotopic peaks as objects to be matched, but does not provide any implementation or evaluation on the effectiveness of the proposed objective function. In MetAssign [57], a Bayesian mixture model was introduced to perform the identification of a set of observed peaks based on how well they fit the theoretical mass spectrum of a metabolite computed from a given formula. While the groupings of related peaks extracted from PrecursorCluster can potentially be used in a similar manner as MetAssign to perform a more robust annotation of metabolites present the sample, here we investigate its uses in improving the alignment step. Unlike MetAssign, PrecursorCluster does not require a prior library of possible metabolite formulas to be specified to perform ionization product clustering, relying only on prior chemical knowledge of which ionization transformations are expected to be present in the data. CAMERA [47] approaches the problem of ionization product clustering from a graph-theoretic perspective. In CAMERA, peak features are nodes in a graph, and edges are drawn between nodes if their scores are greater than a predefined threshold. The graph is clustered to find highly-connected subgraphs, and edges in the subgraph are annotated by known rules of chemical transformations. Unlike

CAMERA, PrecursorCluster is a fully probabilistic model, relying on Bayesian inference to update the probabilities of which LC-MS peak features can be explained by which transformations into IP clusters. This additional information can be used to provide an estimate to the uncertainty of IP annotations. The Bayesian model proposed in PrecursorCluster can also be easily extended to incorporate additional sources of information (e.g. chromatographic peak shapes) for clustering peaks in a different manner.

Since alignment is such an important part of the data preprocessing steps, it is useful to be able to robustly identify the uncertainty or confidence in the alignment results. In the absence of ground truth information (typically the case in untargeted metabolomics experiment), the user measures alignment quality through manual inspection or by comparing and visualising the summary statistics (e.g. median, standard deviation of retention time) across different replicates. Alignment methods that can produce matching confidence values is a big research gap that, to our knowledge, has not been addressed by any of existing direct-matching tools. Tools such as MAVEN [108] assigns quality scores to individual peaks by training a predictive model on expert-annotated training data of peak quality metrics, but this does not extend to scoring groups of peaks. Other approach like [109] computes the Pearson correlations between intensity profiles of all peaks across replicates. Moving from these approaches towards a robust method that can provide confidence values for groups of aligned peaks across many label-free experiments is challenging research problem.

The subject of identifying and quantifying uncertainty has been extensively investigated in the problem of multiple sequence alignment (MSA) for genomics and transcriptomics. The probability on MSA alignment allows researchers to focus on regions of the genome that are difficult to align, potentially revealing evolutionary insights as such regions have high alignment uncertainty that can be the result of e.g. the lack of conserved sequences. [110] attempt to quantify the alignment uncertainty of the popular MSA tool ClustalW [111], based on evaluations using synthetic data, and concludes that between half to all columns in their benchmark MSA results contain alignment errors. [112] construct a score that reflects the consensus between all possible pairwise alignments in T-COFFEE, while [113] propose GUIDANCE, a confidence measure obtained from pertubations of guide trees. Statistical approaches that provide a measure of confidence in alignment results have also been explored by [114] and [115], where the MSA results and phylogeny are constructed simultaneously, thus eliminating the need for a guide tree.

Despite the clear benefits of alignment uncertainty quantification in the sequence domain, the challenge of quantifying alignment results remains relatively unaddressed for the alignment of multiple runs in LC-MS-based-omics. Uncertainties on aligned peaksets can be used by metabolomic researchers to flag the low-probabilities peaksets from further analysis. The flip side of this is high-probabilities peaksets, that we are more confident of as being aligned correctly, can be selected for subsequent analysis in the pipeline or as the focus of manual

validation in a targeted manner if they are revealed as corresponding to metabolites having an interesting differential change across samples. Several recent feature-based alignment methods incorporate probabilistic modelling as part of their workflow, making it possible to extract some form of scores or probabilities on the alignment results. These methods are often limited to the alignment of two runs, which is not a realistic assumption in actual LC-MS experiments. For example, [116] propose a model for pairwise peak matching. Matching confidence can be obtained from the model in form of posterior probability for any peak pair in two runs, however constructing multiple alignment results in [116] still requires a greedy search to find candidate features within m/z and RT-RT tolerances to a predetermined set of 'landmark' peaks. [117] describe PeakLink, a workflow for alignment that performs an initial warping using a fourth-degree polynomial. PeakLink poses the pairwise matching problem as a binary classification problem, where a Support Vector Machine (SVM) is trained based on an alignment ground truth derived from MS-MS information and used to differentiate matching and non-matching candidate pairs to produce the actual alignment results. While not explicitly included in the output of PeakLink, a matching score can be extracted from the SVM that represents how far each candidate pair is from the decision boundary. Note that these scores are not well-calibrated in the probabilistic sense, thus making comparisons of matching scores less straightforward. PeakLink is also not extended to the problem of aligning multiple runs, although [117] state that it would be possible to do so with the choice of a suitable reference run.

Statement of Original Work

The work from this chapter has been submitted for review to *Bioinformatics*. The author proposed and implemented the idea of clustering IP peaks by their transformations, and also the matching of the resulting IP clusters to construct alignment. The author also performed the evaluation of performance of the proposed approach against the baseline methods.

5.3 Methods

The workflow is illustrated in Figure 5.1. A novel Bayesian model, **PrecursorCluster**, is introduced to group related peaks into IP clusters (Section 5.3.1). Each LC-MS run is processed separately through PrecursorCluster — the model takes as input the list of m/z, RT and intensity values of peak features and the list of user-defined transformations and produces as output the set of IP clusters per run. Alignment of IP clusters across runs are performed through **Cluster-Match** (Section 5.3.2) or **Cluster-Cluster** (Section 5.3.3). From Cluster-Match, a list of aligned peaksets (the set of peak features matched across runs) is

obtained, while from Cluster-Cluster, the resulting aligned peaksets are produced alongside the probabilities of matching confidence.

5.3.1 PrecursorCluster: clustering of ionization product peaks

PrecursorCluster uses a mixture model to group the multiple ionization products that arise from each metabolite. We describe and evaluate the model for the positive ionization mode data, but the method could easily be adapted to negative mode data. In a run, the n-th peak feature is represented as the vector $\mathbf{d}_n = (d_n^m, d_n^t, d_n^p)$ with d_n^m the m/z value, d_n^t the RT value and d_n^p the intensity value of that peak. A list of T transformation functions of commonly-known IP types is also required (for e.g. see Table 5.1). A transformation function t_k takes as input the observed m/z value of a peak and produces as output the precursor mass into an IP cluster k under that transformation. This takes the form of $t_k(d_n^m) = \frac{d_n^m|c|+ce-\sum_i h_i G_i}{n}$, where c is the charge, e is the mass of an electron, n the multiplicity of the original molecule, and h_i and G_i are the count and atomic masses of the ith adduct part. For example, for $[M+H+NH4]^+$, c is 2, n is 1 while $\sum_i h_i G_i$ is the total atomic mass of $H + NH3$.

Although it is not strictly necessary, we found it useful to add some constraints to our mixture model. In particular, we make the assumption that an IP cluster must contain the $[M+H]^+$ ion peak and this must be the most intense peak. Although this will not always be the case, we found good performance under this assumption. These assumptions allow us to define the complete set of clusters — one for each peak, with the precursor mass of that cluster computed via assuming the peak is an $[M+H]^+$ ion. The k-th cluster is represented by the tuple (c_k^m, c_k^t, c_k^p), where the cluster's precursor mass c_k^m is the M+H transformed precursor mass of the respective peak's m/z value, and the cluster's RT (c_k^t) and intensity (c_k^p) values are the peak's RT and intensity values. Having created the set of clusters, an enumeration step is performed to determine which possible clusters each peak can belong to. A peak $\mathbf{d}_n$ can be assigned to a possible cluster k if **(1)** the m/z value of that peak can be transformed (through any of the T transformations) into a precursor mass value that is within γ_m, the tolerance in parts-per-million (ppm), from c_k^m, **(2)** the RT value of that peak is within a certain tolerance (γ_t seconds) from c_k^t and **(3)** the intensity of that peak is less than the cluster's intensity threshold c_k^p. All observed peaks belong to at least one possible cluster (the one for which it is the $[M+H]^+$ peak).

Let z_{nk} denote the assignment of peak feature $\mathbf{d}_n$ to a possible cluster k, i.e. z_{nk} is 1 if peak n is assigned to cluster k and 0 otherwise. A peak can only be assigned to exactly one cluster ($\sum_{k=1}^{K} \boldsymbol{z}_{nk} = 1$). Following the standard mixture model construction, $\boldsymbol{z}_n$ is modelled as a multinomial distribution having the parameter vector $\boldsymbol{\theta}$, itself drawn from a prior Dirichlet distribution having the symmetric parameter α. The likelihood of a peak $\mathbf{d}_n$ being assigned to a cluster k depends on the likelihood of that peak's transformed precursor mass and RT

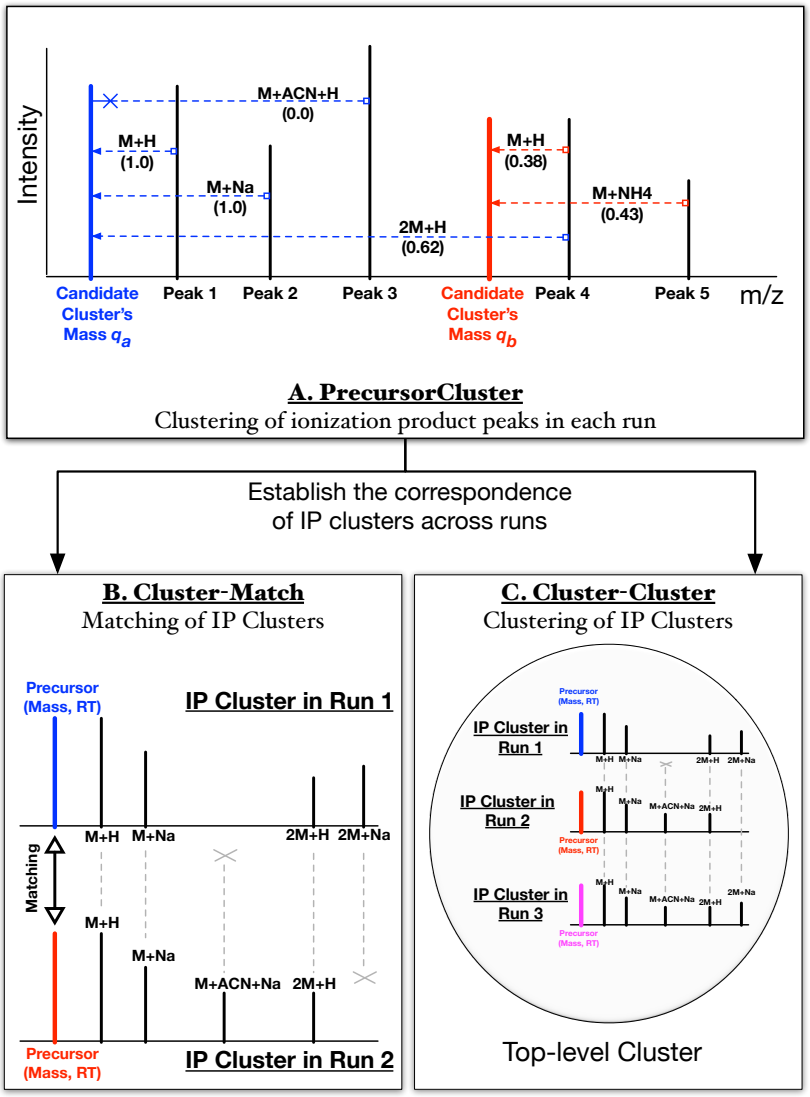


Figure 5.1: The proposed workflow for alignment using ionisation product (IP) clusters.. The input to PrecursorCluster is a list of m/z, RT and intensity values. During the enumeration stage, candidate IP clusters are generated from each peak through the M+H transformation. In this example, Peak 1 and Peak 4 generate candidate IP clusters with precursor masses q_a (blue) and q_b (red). In the inference stage, Peak 1 and Peak 2 are clustered to q_a through transformation M+H and M+Na with probabilities 1.0. Peak 3 has a valid transformation to q_a, but is not allowed to join that cluster since its intensity is $>$ than the intensity of the $[M+H]^+$ peak that generated the cluster (peak 1). Peak 4 can join the q_a cluster through the 2M+H transformation (with probability 0.62) or form its own candidate M+H cluster having the precursor mass q_b (with probability 0.38.) The latter allows for Peak 5 to join that cluster through the M+NH4 transformation (with probability 0.43). The final clustering is established by taking the *maximum a posteriori* assignment for each peak feature. Non-empty IP clusters can be aligned by matching their posterior precursor mass and RT values (Fig. 5.1B) or through a second-stage clustering process (Fig. 5.1C). The correspondence of peak features in matched IP clusters is constructed by grouping peak features having the same transformation types, shown as the gray dotted lines in Figures 5.1B & C.

values under the possible cluster's mass and RT values. Assuming independence between mass and RT terms, this is:

$$p(\boldsymbol{d}_n|\boldsymbol{z}_{nk}=1,...) = p(t_k(d_n^m)|\boldsymbol{z}_{nk}=1,...)\cdot p(d_n^t|\boldsymbol{z}_{nk}=1,...). \tag{5.1}$$

The likelihood of the transformed precursor mass $t_k(d_n^m)$ in the mass term $p(t(d_n^m)|\boldsymbol{z}_{nk}=1,...)$ in eq. (5.1) is a product of two further terms. The first is the indicator function $I(n,t,k)$, set to 1 if no other peaks apart from $\mathbf{d}_n$ are currently assigned to cluster k through transformation t, and 0 otherwise. This allows each transformation type to appear only once in each cluster. We assume that the mass of cluster k, μ_k^m, has a Gaussian prior with mean c_k^m and fixed precision δ. The precision is set to reflect the mass tolerance in parts-per-million used during the enumeration of peaks to possible clusters, such that one standard deviation ($\sqrt{\delta^{-1}}$) is $\frac{\gamma_m * c_k^m/1e6}{3}$. Within a cluster, we assume Gaussian noise in the mass, with the prior mass mean μ_0 set to the value of the cluster's precursor mass c_k^m used during enumeration and precision again equal to δ. The mass component of the likelihood is given by:

$$p(t_k(d_n^m)|\boldsymbol{z}_{nk}=1,...) = I(n,t,k)\cdot\mathcal{N}(t_k(d_n^m)|\mu_k^m,\delta^{-1}) \tag{5.2}$$

$$p(\mu_k^m|\mu_0,\delta) = \mathcal{N}(\mu_k^m|\mu_0,\delta^{-1}) \tag{5.3}$$

Similarly, Gaussian noise is assumed for the RT values. The k-th cluster has mean RT value given by μ_k^t and precision λ set to reflect the RT tolerance used during enumeration of possible assignments, i.e. γ_t is $3\sqrt{\lambda^{-1}}$. Within a cluster, the noise is assumed Gaussian, with the prior RT mean ψ_0 set to the cluster's RT value c_k^t and precision λ:

$$p(d_n^m|\boldsymbol{z}_{nk}=1,\mu_k^t,\lambda) = \mathcal{N}(d_n^m|\mu_k^t,\lambda^{-1}) \tag{5.4}$$

$$p(\mu_k^t|\psi_0,\lambda) = \mathcal{N}(\mu_k^t|\psi_0,\lambda^{-1}) \tag{5.5}$$

A collapsed Gibbs sampling scheme is used to infer $\boldsymbol{z}_{nk}$, the assignments of peak n to cluster k (details in the next section). Averaging over the posterior samples, peaks are assigned to the most likely IP cluster based on their *maximum a-posteriori* (MAP) probabilities. The result from inference is the set of IP clusters, some of which may be empty and can be ignored, while others consist of related ionization products.

PrecursorCluster can be seen as a data-reduction procedure, taking as input the set of observed peak features per run and producing as output their MAP assignments into IP clusters. Non-empty IP clusters can now take the place of individual peak features as objects to be aligned. Each IP cluster in run j can be represented by $\boldsymbol{c}_{jk} = (\bar{q}_{jk}, \bar{r}_{jk}, \bar{\boldsymbol{u}}_{jk})$., with $\bar{q}_{jk}$ the IP cluster's posterior precursor mass value, $\bar{r}_{jk}$ the posterior RT value and $\bar{\boldsymbol{u}}_{jk}$ the adduct 'fingerprint' vector of length T for that IP cluster, created after the MAP assignments of observed peaks into the cluster. This stores the binary flags on which adduct transformations

bring member peaks into that IP cluster (1 if that transformation brings a peak into the cluster and 0 otherwise). These posterior mass, RT and adduct fingerprint values are used during the latter alignment stage.

Gibbs Sampling for PrecursorCluster

For Gibbs sampling, the conditional distribution of a peak $\boldsymbol{d}_n$ currently being sampled to be placed in any of the K IP clusters is given by

$$P(z_{nk} = 1|\boldsymbol{d}_n, \ldots) \propto (\alpha_k + n_k) \cdot p(\mathbf{d}_n|z_{nk} = 1, ...) \tag{5.6}$$

where n_k is the current number of members (peak features) in an IP cluster k, $\alpha_k = \frac{\alpha}{K}$ the symmetric prior on the Dirichlet distribution and $p(\mathbf{d}_n|z_{nk} = 1, ...)$ is the likelihood of peak $\mathbf{d}_n$ in a cluster k. Assuming independence between the mass and RT terms, the likelihood $p(\mathbf{d}_n|z_{nk} = 1, ...)$ can be factorised into its mass and RT terms (see eq. 5.1). However, the probability of a peak n to be placed in cluster k is 0 if the indicator function $I(n, t, k)$ in eq. (5.2) returns 0, i.e. another peak apart from n is already assigned to cluster k through transformation t. Otherwise, marginalising over all mixture components in eq. (5.2), the following posterior predictive distribution is obtained for the mass term:

$$p(t_k(d_n^m)|z_{nk} = 1...) \quad = \quad \mathcal{N}(t_k(d_n^m)|\mu_k, \sigma_k^{-1}) \tag{5.7}$$

where $\sigma_k = (\delta(1 + c_k)^{-1} + \delta^{-1})^{-1}$ and $\mu_k = \frac{1}{\sigma_k}[\delta(\mu_0 + \sum_n t(d_{n\in k}^m))]$. Here, $\sum_n t_k(d_{n\in k}^m)$ denotes the sum of the transformed mass values of all the peaks (excluding the current peak being sampled) that have been assigned to cluster k, and c_k the count of such peaks. Similarly, the RT term in eq. 5.4 can be marginalized into

$$p(d_n^t|z_{nk} = 1...) \quad = \quad \mathcal{N}(d_n^t|\mu_k, \sigma_k^{-1}) \tag{5.8}$$

where $\sigma_k = (\lambda(1 + c_k)^{-1} + \lambda^{-1})^{-1}$ and $\mu_k = \frac{1}{\sigma_k}[\delta(\psi_0 + \sum_n d_{n\in k}^t)]$, with $\sum_n d_{n\in k}^t$ denoting the sum of the RT values of all the peaks (excluding the current one) in cluster k.

5.3.2 Cluster-Match: direct matching of ionization product clusters

The ionization product clustering model described in Section 5.3.1 is essentially a data-reduction procedure, where within a single file j, the model takes as input the set of observed peaks in a single run and produces as output their groupings into IP clusters. Given the set of non-empty IP clusters and the peak features they contain, we can now treat IP clusters

as a reduced set of features within a run and align (match) them across runs. We call this approach Cluster-Match. This contrasts to the conventional approach of matching all peak features directly to produce the alignment of peak features across runs.

As detailed in Chapter 4, in the direct matching alignment of two runs, the problem of establishing the matching between two runs can be viewed as finding the maximum weighted matching in a bipartite graph, where a node in the graph represents a peak feature, an edge represents a potential matching across two sides of the graph and the edge weight is the similarity between two potential matches. The MW method in Chapter 4 is an instance of a greedy algorithm that produces an approximation of at least 1/2 of the maximum weight in the matching of a bipartite graph [107]. Only peaks that are within mass and RT tolerances from each other across runs can possibly be matched (they have an edge linking them in the graph). While simple, the results in Chapter 4 shows that the MW method is generally competitive in performance to more sophisticated direct-matching methods, such as SIMA that relies on constructing stable-matching. We apply this direct matching methods to match IP clusters across runs, with IP clusters taking the place of individual peak features as nodes in the bipartite graph to be matched. The matching is therefore performed based on the precursor mass and RT values of IP clusters, rather than the observed peak's m/z and RT values. Once matching has been constructed, the alignment between the actual peak features in matched IP clusters can be established by grouping peaks that have the same transformation type across matched IP clusters (Figure 5.1B.)

To extend the above procedure to the alignment of multiple runs, two initial runs are first aligned to construct an intermediate merged results. Consensus features are created by taking the average m/z and RT values of matched features, and the next run is then aligned to the merged results. This procedure is repeated until all runs have been exhausted. This match-merge scheme is commonly employed by other direct matching methods [36, 19] and requires selecting a reference run. In practice, the choice of reference run is arbitrary and its effect has not been fully investigated (in our implementation, the first run in alphabetical sorting is used as the reference run and the same ordering of runs is always used for all methods compared.)

5.3.3 Cluster-Cluster: across-run clustering of ionization product clusters

The proposed pairwise matching scheme used by Cluster-Match is frequently extended to the processing of multiple runs in a fairly ad-hoc manner (as seen in the match-merge approach at the end of Section 5.3.2, also commonly used by other direct matching tools). This approach suffers from the limitation of having to set a reference run for the matching and

consequently, the fact that altering the ordering of runs to be processed might change the alignment results [42]. The alternative approach of generalizing from pairwise matching in a bipartite graph into finding the maximum weighted matching in a general graph is typically a computationally expensive operation. Producing a distance measure that works well for measuring similarities of peaks across runs is a non-trivial problem, and such matching procedures, whether through successive pairwise merging or operating on a general graph, generally do not take into account the uncertainties in the matching of peak features across runs.

Here, we propose using another clustering procedure (Cluster-Cluster) to further cluster the IP clusters produced from the first-stage IP clustering in Section 5.3.1. In this manner, IP clusters coming from different runs are further clustered into top-level clusters shared across runs (Figure 5.1C). The actual alignment of peak features can then be established by **(1)** looking at which IP clusters are put together into the same top-level cluster (essentially, their matching) and **(2)** in a top-level cluster, grouping peak features from different runs that have the same transformation type to establish their alignments. In this scheme, there is no need to set a reference run. Crucially, the posterior probabilities of certain IP clusters being assigned into the same top-level cluster provides us with an estimate of matching confidence of peak features.

Only peaks within a certain across-run mass tolerance should be matched, so a partitioning of IP clusters into top-level bins is performed. Across all runs $j = 1, ..., J$, IP clusters are sorted by their posterior mass values $\{\bar{q}_{jk}\}$. The smallest unprocessed mass value $min(\{\bar{q}_{jk}\})$ is used to initialize a top-level bin. Subsequent IP clusters (in ascending mass order) are grouped into the bin until an IP cluster with a posterior mass that differs by γ'_m ppm (a user-defined mass tolerance across runs) from $min(\{\bar{q}_{jk}\})$ is encountered, in which case, a new top-level bin is started using that cluster. The process is repeated until all IP clusters are processed.

If a top-level bin contains only one IP cluster, no possible matching can be constructed, otherwise IP clusters in the same bin can potentially be clustered (into top-level clusters) and therefore matched. To avoid specifying the number of top-level clusters *a priori*, we use an infinite Gaussian mixture model, described in Chapter 3, to model the data. Let $\bar{z}_{jki} = 1$ denote the assignment of IP cluster k coming from file j into top-level cluster i.

Then:

$$\boldsymbol{\pi}|\alpha' \sim GEM(\alpha') \tag{5.9}$$

$$\bar{\boldsymbol{z}}_{jk}|\boldsymbol{\pi} \sim Multinomial(\boldsymbol{\pi}) \tag{5.10}$$

$$\boldsymbol{c}_{jk}|\bar{z}_{jki} = 1, ... \sim p(\boldsymbol{c}_{jk}|\bar{z}_{jki} = 1, ...) \tag{5.11}$$

where $\boldsymbol{\pi}$ are the mixing proportions, distributed according to the GEM (Griffiths, Engen and McCloskey) distribution (details in Chapter 3). The likelihood of $\boldsymbol{c}_{jk}$, the k-th IP cluster from run j, to be placed in a top-level cluster i is assumed to be factorized into independent factors of its mass, RT and adduct signature terms:

$$p(\boldsymbol{c}_{jk}|\bar{z}_{jki}=1,...) = p(\bar{q}_{jk}|\bar{z}_{jki}=1,...)\cdot p(\bar{r}_{jk}|\bar{z}_{jki}=1,...)\cdot p(\bar{\boldsymbol{u}}_{jk}|\bar{z}_{jki}=1,...) \tag{5.12}$$

In eq. (5.12), the mass term $p(\bar{q}_{jk}|\bar{z}_{jki}=1,...)$ is defined analogously to the first-stage clustering step (Section 5.3.1). The indicator function $\bar{I}(k,j,i)$ in eq. (5.13) is set to 1 if there are no other IP clusters from run j, apart from the k-th IP cluster, that are assigned to the i-th top-level cluster, and 0 otherwise. This ensures that there is at most one IP cluster from each run assigned to a top-level cluster. The IP cluster posterior mass $\bar{q}_{jk}$ is distributed according to a Gaussian distribution with mean c_m and precision $\bar{\delta}$, where the across-run mass tolerance γ'_m is set to be equivalent to 3 standard deviations in ppm. The mass of top level cluster c_m is in turn drawn from a base Gaussian distribution having prior mass mean $\bar{\mu}_0$ and precision σ_m (eq. 5.14). The $\bar{\mu}_0$ parameter is set to the mean of the posterior m/z values of the IP clusters in the top-level bin, while σ_m is set to a broad value of 5E-3.

$$p(\bar{q}_{jk}|\bar{z}_{jki}=1,c_m,\bar{\delta},...) = \bar{I}(j,i)\cdot\mathcal{N}(\bar{q}_{jk}|c_m,\bar{\delta}^{-1}) \tag{5.13}$$

$$p(c_m|\bar{\mu}_0,\sigma_m) = \mathcal{N}(c_m|\bar{\mu}_0,\sigma_m^{-1}) \tag{5.14}$$

In the RT term $p(\bar{r}_{jk}|\bar{z}_{jki}=1,...)$, $\bar{r}_{jk}$ is distributed according to a Gaussian distribution with mean c_t and precision $\bar{\lambda}$ (eq. 5.15). Again, the across-run RT tolerance γ'_t is set to be equivalent to 3 standard deviations in seconds. The same uninformative parameter values are set on the prior RT mean parameter $\bar{\psi}_0$ and precision σ_t (eq. 5.16).

$$p(\bar{r}_{jk}|\boldsymbol{z}_{jki}=1,c_t,\bar{\lambda}) = \mathcal{N}(\bar{r}_{jk}|c_t,\bar{\lambda}^{-1}) \tag{5.15}$$

$$p(c_t|\bar{\psi}_0,\sigma_t) = \mathcal{N}(c_t|\bar{\psi}_0,\sigma_t^{-1}) \tag{5.16}$$

Finally, in the adduct fingerprint term $p(\bar{\boldsymbol{u}}_{jk}|\boldsymbol{z}_{jki}=1,...)$, the vector $\bar{\boldsymbol{u}}_{jk}$ is modelled using a multinomial distribution having a Dirichlet prior with symmetric hyper-parameter β. The entire likelihood function of eq. 5.12 ensures that IP clusters from different runs are placed in a single top-level cluster if: **(1)** they are from different runs, **(2)** they share similar posterior precursor mass and RT values, and **(3)** they have similar adduct fingerprint. Inference on model parameters is again performed via Gibbs sampling. Within each posterior sample, peak features in matched IP clusters sharing the same transformation type are grouped (Figure 5.1C), forming aligned peaksets. The occurrences of aligned peaksets are counted and averaged across samples to give matching confidences.

Gibbs Sampling for Cluster-Cluster

Analytical inference is not tractable here, so we use a collapsed Gibbs sampling scheme for inference of Cluster-Cluster. The conditional probability of $P(\bar{z}_{jki} = 1|...)$ of IP cluster k in file j to be placed in an existing top-level cluster i (or i^* if a new top-level cluster is to be created), is given by:

$$P(\bar{z}_{jki} = 1|\boldsymbol{c}_{jk}, \ldots) \propto \begin{cases} n_i \cdot p(\boldsymbol{c}_{jk}|\bar{z}_{jki} = 1, ...) \\ \alpha' \cdot p(\boldsymbol{c}_{jk}|\bar{\boldsymbol{z}}_{jki^*} = 1, ...) \end{cases} \tag{5.17}$$

where n_i is the current number of members (IP clusters) in an existing top-level cluster i. $p(\mathbf{y}_n|z_{nk} = 1, ...)$ is the likelihood of peak $\mathbf{y}_n$ in an existing cluster k. The top part of eq. (5.17) is the conditional probability on existing mixture components of the model, and can be factorized into its independent mass, RT and adduct fingerprint terms. The bottom part of eq. (5.17) represent new components that are created as needed.

1. For the mass term $p(\bar{q}_{jk}|\bar{z}_{jki} = 1, ...)$, we obtain the following predictive distribution after marginalizing over all mixture components:

$$p(\bar{q}_{jk}|\bar{z}_{jki} = 1, ...) \quad = \quad \mathcal{N}(\bar{q}_{jk}|\mu_k, \gamma_k^{-1}) \tag{5.18}$$

   where $\gamma_k = ((\sigma_m + \bar{\delta} n_i)^{-1} + \bar{\delta}^{-1})^{-1}$ and $\mu_k = \frac{1}{\gamma_k}\left[(\sigma_m \bar{\mu}_0) + (\bar{\delta} \sum_j \sum_k \bar{q}_{jk\in i})\right]$. Note that in the summation terms of μ_k, $\sum_j \sum_k \bar{q}_{jk\in i}$ denotes the sum of posterior mass values of IP clusters currently assigned to top-level cluster i (excluding the current IP cluster being sampled), and n_i the count of such IP clusters.

2. Similarly, the RT term $p(\bar{r}_{jk}|\bar{z}_{jki} = 1, ...)$ can be marginalized into a Gaussian with precision $\gamma_k = ((\sigma_t + \bar{\lambda} n_i)^{-1} + \bar{\lambda}^{-1})^{-1}$ and mean $\mu_k = \frac{1}{\gamma_k}\left[(\sigma_t \bar{\psi}_0) + (\bar{\lambda} \sum_j \sum_k \bar{r}_{jk\in i})\right]$, with $\sum_j \sum_k \bar{r}_{jk\in i}$ the sum of posterior RT values of member IP clusters in top-level cluster i, excluding the current IP cluster being sampled.

3. Lastly for the adduct fingerprint term $p(\bar{\boldsymbol{u}}_{jk}|\bar{z}_{jki} = 1, ...)$, we marginalize over the mixture components and obtain $\frac{C\left(\bar{\boldsymbol{u}}_{jk} + \sum_j \sum_k \bar{\boldsymbol{u}}_{jk\in i} + \boldsymbol{\beta}\right)}{C\left(\sum_j \sum_k \bar{\boldsymbol{u}}_{jk\in i} + \boldsymbol{\beta}\right)}$ with $\sum_j \sum_k \bar{\boldsymbol{u}}_{jk\in i}$ the sum of all adduct fingerprint vectors currently assigned to top-level cluster i (excluding the current IP cluster being sampled), $C(\boldsymbol{X}) = \frac{\prod_{j=1}^{m} \Gamma(\boldsymbol{X}_j)}{\Gamma(\sum_{j=1}^{m} \boldsymbol{X}_j)}$ and Γ the gamma function.

For new components, marginalising over the base distributions for the mass term results in a Gaussian with mean $\bar{\mu_0}$ and precision $\bar{\delta}^{-1} + \sigma_m^{-1}$. Similarly, for the RT term, this results in a Gaussian with mean $\bar{\psi}_0$ and precision $\bar{\lambda}^{-1} + \sigma_t^{-1}$. For the adduct term, this results in $\frac{C\left(\bar{\boldsymbol{u}}_{jk} + \boldsymbol{\beta}\right)}{C(\boldsymbol{\beta})}$.

5.4 Evaluation Study

5.4.1 Evaluation Datasets

Two metabolomics datasets were used for performance evaluation. The Standard dataset was generated from a mixture of 104 standard metabolites used for chromatographic columns calibration and has been used for performance evaluation in in Chapter 4. This dataset contains eleven runs and represents a challenging alignment scenario with large RT variability (runs were separated by weeks and generated from different instruments). A Beer dataset of three runs from one batch that is representative of the typical biochemical diversity in a complex metabolomics study is introduced. All runs were processed through PrecursorCluster using the same parameters and the list of transformations in Table 5.1. This list includes the common adduct transformations in positive ionisation mode, but optionally isotopes can also be included.

Alignment ground truth for both datasets was constructed from the putative identification of each run at 3 ppm using the Identify module from mzMatch [48], taking as input a database of the 104 standard compounds known to be present and the transformations in Table 5.1. Peak features with the same unique identifications are matched across runs, resulting in an alignment ground truth for a subset of all peaks. Only peaks present in the ground truth are considered for evaluation. The Standard ground truth accounts for 304 aligned peaksets (the set of peak features matched across runs) spanning 1936 peak features across all Standard runs, while the Beer ground truth consists of 108 aligned peaksets of 300 peak features across all Beer runs.

Table 5.1: List of common adduct transformations in positive mode used for the precursor clustering of the Standard and Beer runs.

M+2H	M+H	M+ACN+H	2M+Na	M+H+NH4
M+NH4	M+ACN+Na	2M+ACN+H	M+ACN+2H	M+Na
M+2ACN+H	M+2ACN+2H	M+CH3OH+H	2M+H	

5.4.2 Performance Measures

Precision and recall are widely used to evaluate alignment performance [33, 19, 35, 36, 118], and also in Chapter 4. To evaluate alignment performance on multiple runs, we propose a generalized definition of precision and recall that extends from the pairwise definition in Chapter 4. From an alignment method or the ground truth, a list of aligned peaksets is obtained. For example, from the alignment of 4 runs, an alignment method returns a list of two aligned peaksets $\{a, b, c, d, \}, \{e, f, g\}$ as output. Here, $\{a, b, c, d, \}$ is an aligned

peakset spanning 4 runs, while $\{e, f, g\}$ contain peaks from 3 runs. From each aligned peakset, we can extract a list of l-size combinations of peaks, each comprising an 'alignment item'. For instance, when $l = 2$, the example output can be enumerated into a list of 9 'alignment items' of all the pairwise combinations of features: $\{a, b\}, \{a, c\}, \{a, d\}, \{b, c\}, \{b, d\}, \{c, d\}, \{e, f\}, \{e, g\}, \{f, g\}$. Let M and G be the results from such enumeration from a method's output and the ground truth respectively. Each distinct combination of features in M and G can be considered as an item during performance evaluation. Intuitively, the choice of l reflects the strictness of what is considered to be a true positive item, with larger values of l demanding an alignment method that produces results spanning more runs correctly. In this manner, l goes from 2 to as many runs being aligned.

For a given l, the following positive and negative instances of alignment item can now be defined for the purpose of performance evaluation:

- True Positive ($\boldsymbol{TP}$): items that should be aligned (present in G) and are aligned (present in M).
- False Positive ($\boldsymbol{FP}$): items that should not be aligned (absent from G) but are aligned (present in M).
- True Negative ($\boldsymbol{TN}$): items that should not be aligned (absent from G) and are not aligned (absent from M).
- False Negative ($\boldsymbol{FN}$): items that should be aligned (present in G) but are not aligned (absent from M).

In a similar manner to the pairwise definition in Chapter 4, precision ($\frac{\boldsymbol{TP}}{\boldsymbol{TP+FP}}$) is the fraction of alignment items in M that are correct with respect to some alignment ground truth G, while recall ($\frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}}$) is the fraction of alignment items specified in G that are actually aligned in the alignment results M. By definition, a perfect alignment method would have precision and recall scores of 1. In practice, there is a trade-off between precision and recall, where increasing recall often results in lower precision and vice versa. To summarize these two numbers, we also report the F_1 score, which is the harmonic mean of precision and recall, defined as $F_1 = 2(precision \cdot recall)/(precision + recall)$. Since our alignment ground truth is usually smaller than the set of all pairs of peaks returned by a method, only those peaks present in the ground truth are considered for evaluation.

5.4.3 Evaluation Procedure

As the baselines for evaluation, we compare the performance of our proposed methods against the method of direct matching of peak features (MW) and its variant (MWG) that

modifies the similarity matrix used during matching to bring together group of peaks related by RT closer during matching – described in Chapter 4.

To evaluate Cluster-Match, the procedure in Chapter 4 is followed. 30 random pairs of Standard runs were selected as the training set, and another 30 as the testing set. Matching tolerance parameters were varied within reasonable ranges (details in Section 5.4.4) on one pair in the training set, and parameters resulting in the best training performance (highest F_1-score) of one pair were used to align the associated pair in the testing set. The three Beer runs are too few to allow separation into training and testing sets, so each method is trained and evaluated on all three Beer runs. The direct matching of peak features (MW) and its variant (MWG) from in Chapter 4 that incorporates grouping information (based on RT and not mass) into the similarity matrix used for matching are used as a baseline.

To evaluate Cluster-Cluster, five sets of 2, 3, and 4 Standard runs were selected randomly as well as all 3 Beer runs. For each data set, parameters for Cluster-Match were varied to obtain the best attainable alignment performance. These are plotted alongside the results from Cluster-Cluster on the same data using a fixed (and potentially non-optimal) set of parameters. Cluster-Cluster was also run with and without the adduct fingerprint term to evaluate its importance. More details on parameter optimization can be found in Section 5.4.4.

5.4.4 Parameter Optimization

Following the parameter optimization procedure in Section 4.4.6, the same grid search on the m/z and Rt window tolerance parameters is used. The m/z and RT window tolerance parameters define the maximum deviation acceptable for a candidate matching is allowed in the bipartite graph. The choice of m/z parameter is often determined by the accuracy of the mass spectrometry instrument and can be reasonably determined in advance. Due to RT drift, selecting the RT window is less straightforward.

For the evaluation of feature matching (MW, MWG) vs. cluster matching (Cluster-Match) on the Standard dataset, we performed grid-search on the m/z and RT windows parameters using the training set. The optimal training parameters are used to perform alignment on the testing set, giving the respective performance measures (testing Precision, Recall, F_1). On the Standard datasets, we varied the mass tolerance window of the methods tested within the range $\{2, 4, 6, 8, 10\}$ m/z and the RT tolerance window within $\{5, 10, 15, ..., 100\}$ seconds during the training stage. Parameter combinations that result in the best F1-score were then used for performance evaluation in the testing stage. For MWG, additional parameters are also required for the threshold t_g on greedy clustering of related peaks and α_g, the contribution on the different parts to the similarity score. We let t_g vary within $\{2, 4, 6, 8, 10\}$ seconds and α_g within $\{0, 0.2, 0.4, 0.6, 1.0\}$ in the training stage and use the best combinations of parameter values for the testing stage. The three Beer runs are too few to allow separation into

training and testing sets, so each method is trained and evaluated on all three Beer runs using the previously-described parameters same as the Standards.

The following parameters were used for the first-stage clustering of the PrecursorCluster model for all the Standard runs being processed: within-run mass tolerance $\gamma_m = 5$ ppm, within-run RT tolerance $\gamma_t = 30$ seconds. For the Beer runs, we used the within-run mass tolerance $\gamma_m = 3$ ppm and the within-run RT tolerance $\gamma_t = 10$ seconds. The prior on the Dirichlet distribution α is set to 1.0 and Table 5.1 shows the list of common adduct transformations in positive ionization mode used for precursor clustering. 5000 posterior samples were obtained from Gibbs sampling.

For the second-stage clustering in Cluster-Cluster, the following parameters were used for all input Standard and Beer runs: across-run mass tolerance $\gamma'_m = 10$ ppm, across-run RT tolerance $\gamma'_t = 60$ seconds, α' the Dirichlet Process concentration parameter is set to 1000.0. As relatively few number of runs are being aligned in our experiments, the large value of $alpha'$ encourages more top-level clusters, each having fewer member IP clusters inside. β, the symmetric prior on the Dirichlet prior distribution for adduct signature vector is set to 0.1. Inference is performed on each top-level bin that has more than 1 IP clusters inside, with 500 posterior samples drawn for each top-level bin.

5.5 Results and Discussions

With PrecursorCluster, the large number of peaks present within a single LC-MS run can now be reduced to a smaller number of IP clusters, making alignment easier as fewer objects have to be matched across runs. Section 5.5.1 presents the results of running the ionization product clustering on the Standard and Beer datasets.

While the resulting IP clusters potentially have many uses (e.g. to the problem of annotation of related peaks and the identification of metabolites), peaks assigned to any IP cluster have now been annotated with the transformation type that brings them into the clusters. IP clusters can therefore be aligned across runs (through direct-matching or a second-stage clustering process) and their member peak features (sharing the same transformation type) matched to produce alignment. Section 5.5.2 demonstrated from our experiments how the proposed approach of direct-matching IP clusters can improve upon the matching of LC-MS peak features alone, while Section 5.5.3 describes how the resulting probabilities from Cluster-Cluster can be used to robustly quantify the matching uncertainties.

Being a direct matching method, Cluster-Match performs nearly as fast as alignment by matching of peak features alone while offering better performance. As Cluster-Cluster performs Bayesian inference on which IP clusters should be put together into the same top-level

clusters, the alignment of LC-MS features can now be established without the need for a reference run. While this requires more computational time than the direct-matching alternative (Section **??**), Cluster-Cluster is able to produce confidence scores on the matching quality of aligned peaksets from the posterior summaries computed during inference. This has a potential use in assisting the selection of high-confident aligned peaksets for subsequent analysis in the latter stage of the LC-MS pipeline

5.5.1 Ionization Product Clustering from PrecursorCluster

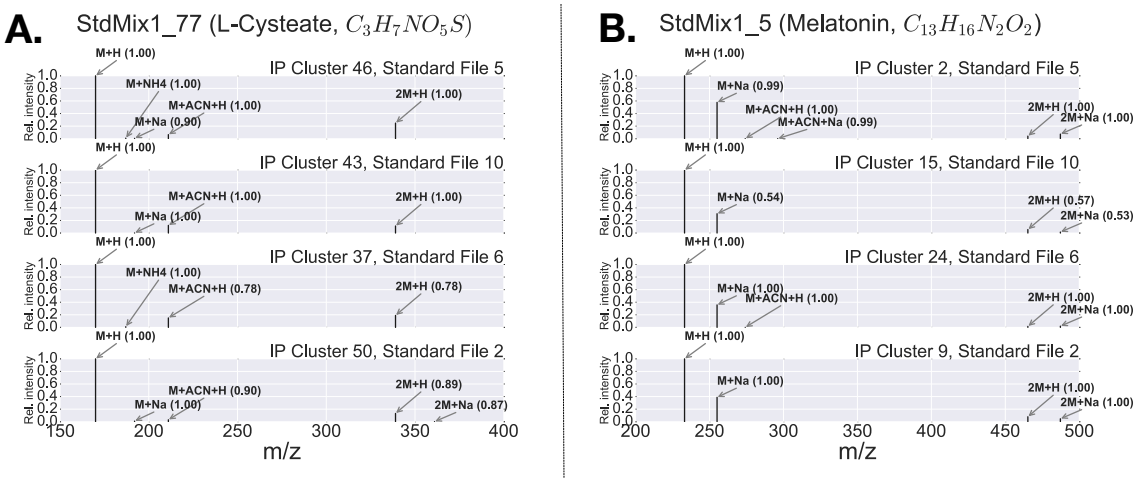


Figure 5.2: Different IP clusters in four different Standard runs, identified as Cysteic acid (Figure 5.2A) and melatonin (Figure 5.2B). The MAP transformation type of a peak and its probability are annotated as a labelled arrow and the bracketed number beside. According to the ground truth, all member peaks with the same transformation type should be aligned.

Within each run, PrecursorCluster produces the *maximum a posteriori* (MAP) assignments of peaks to IP clusters. An example of four IP clusters found in the Standard runs, identified as Cysteic acid, is shown in Figure 5.2A. According to the ground truth, all member peaks across these four clusters should be aligned. Similarly, we show in Figure 5.2B four IP clusters corresponding to the compound melatonin, whose member peaks should all be aligned according to the ground truth. The existence of these correspondent clusters provide an initial indication that it makes sense of match the IP clusters. However, the complexity of the data means that, as Table 5.2 shows, a large number of peaks cannot be clustered to other peaks in the same run and can therefore only form an IP cluster with itself as the only member through the M+H transformation (we call these clusters of only one member peak the singleton IP clusters). In both the Standard and Beer runs, non-singleton IP clusters (containing more than one member peaks) comprise approximately 6% to 10% of the total IP clusters of that run. The distributions of the cluster sizes of these non-singleton clusters when only adduct transformations are used are given in Figure 5.3 for the Standard and Beer

runs. We also note that for any given cluster size, the counts of IP clusters of that size tend to differ significantly across the Standard runs, due to the varying number of LC-MS peak features present in each Standard run. This is the consequence of the Standard runs being produced in several batches separated over a period of time. The distributions of cluster sizes in Figure 5.3 across the three Beer runs are more consistently reproduced, reflecting the fact that the runs were generated within the same batch. As shown in Figure 5.3, the largest IP clusters of the Beer and Standard runs have 6 and 7 member peaks respectively.

Table 5.2: The number of peak features and the counts of singleton and non-singleton IP clusters in each run of the Standard and Beer datasets. A singleton cluster is defined to be an IP cluster having only one member peak after MAP assignments, while a non-singleton IP cluster has more than one member peaks. The last column in the Table shows the counts of non-singleton IP clusters and also the percentage of non-singleton IP clusters from the total IP clusters in that run.

Data	# Peak Features	# Singleton IP Cluster	# Non- singleton IP Cluster
Std 01	4999	4327	301 (6.5%)
Std 02	4986	4341	288 (6.2%)
Std 03	6836	5755	481 (7.7%)
Std 04	9752	8011	775 (8.8%)
Std 05	7076	5801	551 (8.7%)
Std 06	4146	3655	216 (5.6%)
Std 07	6319	5272	469 (8.2%)
Std 08	4101	3579	232 (6.1%)
Std 09	5485	4789	312 (6.1%)
Std 10	5034	4304	310 (6.7%)
Std 11	5317	4574	337 (6.8%)
Beer 01	7553	6179	633 (9.3%)
Beer 02	7579	6203	631 (9.2%)
Beer 03	7240	5983	574 (8.6%)

Consistent with the number of singleton clusters, the M+H transformation dominate in the data. Non M+H transformations comprise 8% of the total MAP transformations for the Standard dataset and 10% for the Beer dataset. In both datasets, the M+ACN+H and M+Na transformations are highly prevalent (Figure 5.4). The presence of the M+ACN+H and M+NH4 transformations in the Beer dataset is expected, given the use of acetonitrile and ammonium carbonate buffers during chromatography. Similarly, the M+CH3OH+H adducts in the Beer data can also be explained by the use of methanol during the sample preparation process. The consistency of the example clusters in Figure 5.2 and the explainable transformations in Figure 5.4 suggest a valid result from PrecursorCluster, providing confidence that it can be used for alignment.

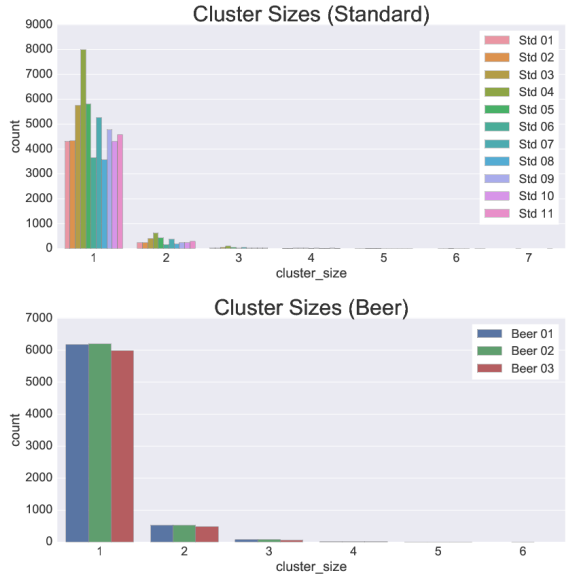


Figure 5.3: Ionization product cluster sizes for all runs in the Standard and Beer datasets. For any given size, the number of clusters are generally more consistent in the Beer runs compared to the Standard runs, which shows greater variability due to the differences in the number of peak features per run.

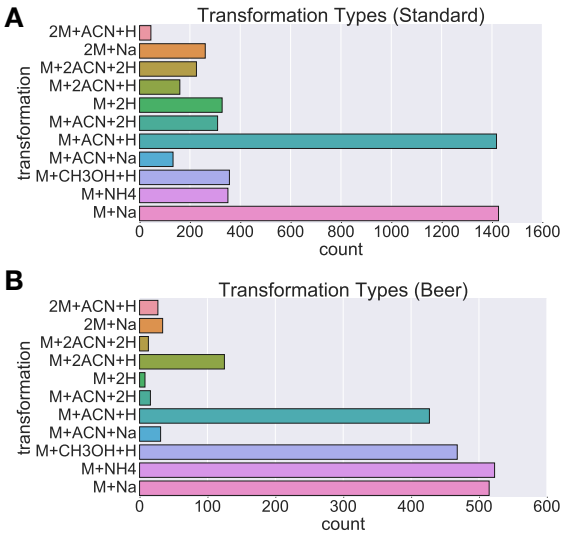


Figure 5.4: Barcharts showing the counts of transformation types in all Standard and Beer runs, excluding the M+H transformation.

5.5.2 Improved Alignment Performance from Cluster-Match

Precision and recall values produced by the different methods (across all parameter ranges) for the entire 30 Standard training sets and the 3 Beer runs can be found in Figure 5.5. Here, l (the size of peakset combinations to be considered during performance evaluation) to 2 to consider only pairwise features for performance evaluation as pairwise performance limits overall performance in direct matching methods that employ the merge-match scheme to construct an overall result. The results in Figure 5.5 (top row) shows that across all the m/z and RT window tolerances varied, Cluster-Match can produce higher precision while retaining similar recall values to feature matching (MW) or modified feature matching (MWG). This increase in precision comes from the increase of true positives and the decrease in false positives by taking into account the ionization product relationships between peak features when constructing the matching. The results here suggest that, regardless of the parameters selected for the m/z and RT tolerance windows, the proposed methods of matching by IP clusters can return a better alignment result compared to matching by peak features only.

Similar results can also be observed for the Beer dataset (Figure 5.5, bottom row). The complex Beer runs being aligned have minimal RT deviations when compared to the Standard runs, so all evaluated methods perform well, demonstrating smaller deviations in performance values despite varying the tolerances parameters. Again a general increase in precision of the results from Cluster-Match is observed over the other two baseline methods. MWG, which relies on the grouping of related peaks using their retention time values only, does not appear to produce any improvements over MW. The results here suggest that on complex LC-MS data such as the Beer data, the richer information present in the m/z and RT values of related peak features, alongside their possible IP transformations and relationships to the precursor peak, is essential and has to be taken into account.

Optimizing parameters on the training set and evaluating performance on the testing set measures how well a method generalizes to new and unseen data. The best Standard training and testing F_1-scores from each method are reported in Figure 5.6. Using a one-sided paired t-test, Cluster-Match is found to be statistically greater than that of MW in both the training (p-value=0.002) and the testing cases (p-value=0.026), suggesting that Cluster-Match generalizes better to new and unseen datasets. MWG produces even higher training F_1-scores compared to the other two methods. This difference is found to be statistically significant using a one-sided paired t-test (p-value=0.01). The higher training performance of MWG can be explained by the fact that the RT grouping tolerance parameter and matching ratio for MWG were optimized during the training phase, while the same (potentially non-optimal) clustering parameters were used for the ionization product clustering of all Standard runs. On the testing results, no statistically significant differences were found on the testing F_1-scores of MWG and Cluster-Match, suggesting that both methods generalize well.

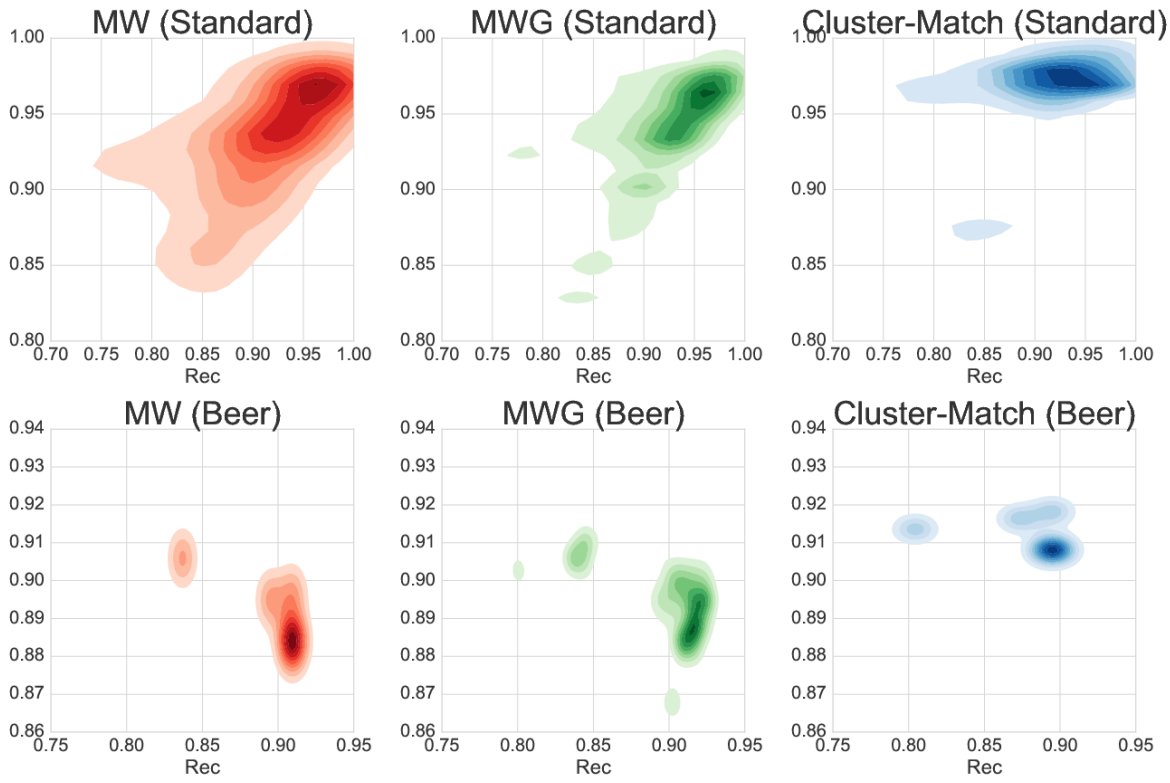


Figure 5.5: All the training results obtained by varying the m/z and RT window parameters from the alignment of the entire 30 sets of pairwise Standard runs (top row) and the 3 Beer runs (bottom row). For MWG, the grouping parameter t and score contribution α were also varied, while for Cluster-Match, the same set parameters of first-stage clustering was used for all input files.

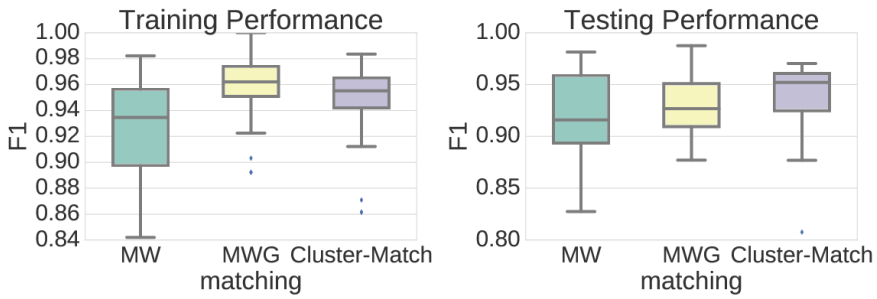


Figure 5.6: The best training and testing F_1-scores obtained from the alignment of 30 sets of pairwise Standard runs.

5.5.3 Probabilistic Matching Results from Cluster-Cluster

Direct-matching methods such as MW and Cluster-Match can only return a definite matching solution to the alignment problem (a peak from one run is either aligned to a peak in the other run, or not). In contrast, the second-stage clustering process of the IP clusters employed in the Cluster-Cluster method allows us to produce an estimate in the uncertainties of matching of peak features, producing as the alignment result a list of aligned peaksets that have been matched at varying levels of confidence. Figure 5.7 shows how a Precision-Recall (PR) curve, which shows how precision and recall change together, can be computed from the output of Cluster-Cluster on one of the sets of 4 randomly selected Standard runs and the set of 3 Beer runs. In Figure 5.7, the PR curves are plotted alongside the results from Cluster-Match at varying m/z and RT tolerance parameters (note that for Cluster-Cluster, we used only one set of potentially sub-optimal parameters for the second-stage clustering). Along both the PR curves on Figure 5.7, we see that generally, a decrease in the recall values is accompanied by an increase in the precision values. This applies to both the Standard and the Beer datasets, suggesting that by setting an appropriate threshold on the probabilities of aligned peaksets returned by Cluster-Cluster, we can obtain fewer aligned peaksets (lower recall) but at a higher confidence level of being correctly aligned (higher precision). In the face of further uncertainties with regard of user-defined parameters from the previous parts of the pipeline, the probabilistic alignment results returned by Cluster-Cluster allows the user to focus on peaksets of high matching confidence for subsequent analysis. This introduces the possibility of returning a smaller subset from the overall aligned peaksets that have a higher confidence score of being correctly aligned — an ability that few other matching methods can provide.

The results in Table 5.3 from running Cluster-Cluster, averaging over the sets of 2, 3, and 4 Standard runs, and on the entire 3 Beer runs, demonstrate that by setting some threshold values $\{0.30, 0.60, 0.90\}$ on the aligned peakset probabilities, various precision and recall values can be extracted. Upon aligning the sets of 4 Standard runs at threshold=0.30, Cluster-Cluster has a lower average precision of 0.81 than the best average performance from Cluster-Match at precision=0.87. Raising the threshold to 0.90 (consequently, decreasing recall as fewer aligned peaksets are returned) produces an average precision=0.90 for Cluster-Cluster, higher than 0.87 for Cluster-Match. On the complex Beer data, Cluster-Cluster produces precision=0.76 at threshold 0.30. Increasing the threshold to 0.90 produces a precision=0.94, which is again higher than the best attainable precision=0.91 from Cluster-Match. This demonstrates how recall can be traded for precision in Cluster-Cluster; a potentially useful ability in untargeted metabolomics experiments when the alignment ground truth is not available. In this situation, analysis effort can be focused on aligned peaksets with high confidence.

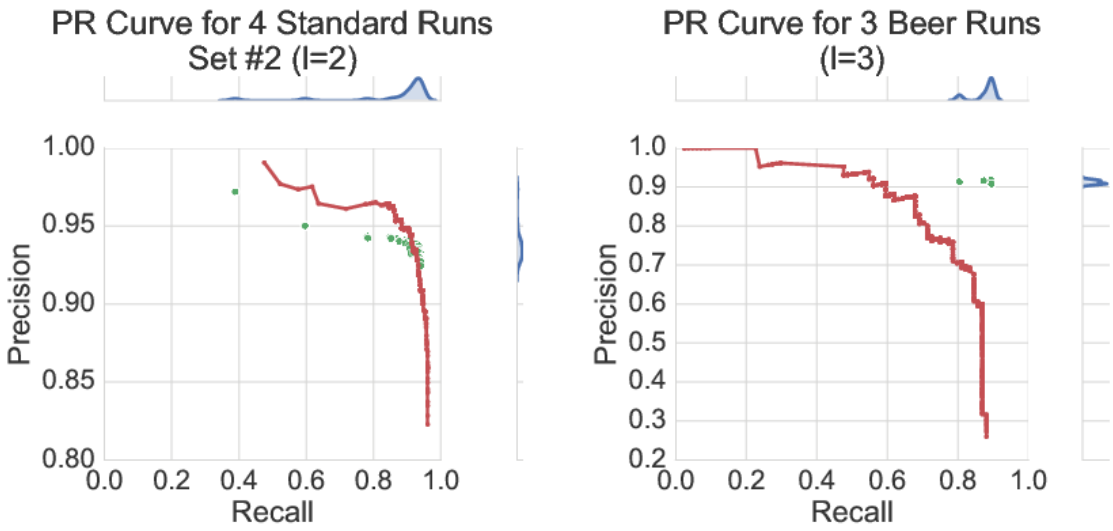


Figure 5.7: PR curves obtained from running Cluster-Cluster on one of the sets of 4 randomly selected Standard runs (left) and the 3 Beer runs (right). Green dots are performance points obtained from running Cluster-Match at varying m/z and RT tolerance parameters on the same datasets, with their distributions of the points plotted along the marginals. The same first-stage clustering results were used as input to both Cluster-Match and Cluster-Cluster.

The importance of the adduct fingerprint term is shown in Table 5.3. Without the adduct fingerprint, a lower average F_1-score is produced by Cluster-Cluster at the probability threshold 0.90. This can be explained by the fact that excluding the adduct fingerprint term allows IP clusters having highly similar precursor mass and RT values to be placed in the same top-level cluster, despite each having an entirely different set of member adduct ions (and potentially corresponding to different metabolites). More false positive alignment items are produced, resulting in lower recall and F_1 scores. Using Figure 5.2 as an example, the aligned peakset consisting of the four $[M+H]^+$ peaks in Figure 5.2 has a high matching probability (0.97) when the adduct fingerprint term is used and almost never be placed together (near 0 probability) without. Similar observations can be concluded for the ions for other transformation types (e.g. the $[M+Na]^+$, $[M+ACN+H]^+$ adduct ions, etc.) shared by the clusters in Figure 5.2. The inclusion of the adduct fingerprint term in Cluster-Cluster is necessary to ensure that well-calibrated probabilities on the alignment results are obtained, especially on aligned peaksets with higher matching confidence.

5.5.4 Running time

Efficient inference is possible in PrecursorCluster as many peaks can only be placed in one cluster and need not be reassigned during Gibbs sampling. For the Standard runs with up to 5000 peak features, less than a quarter of peak features have to be reassigned. Taking 10000

Table 5.3: Precision, recall and F_1 values from Cluster-Cluster for randomly selected sets of 2, 3 and 4 Standard runs (averaged) and the Beer runs for various l and thresholding levels $th = \{0.30, 0.60, 0.90\}$. Best results from Cluster-Match and the result of running Cluster-Cluster without the adduct fingerprint term are shown for comparison. Note that for Cluster-Cluster, the results come from using one set of potentially sub-optimal parameters for the second-stage clustering.

Dataset	l	Best Cluster-Match			Cluster-Cluster (CC)				CC (without adduct term)
		Avg. Prec.	Avg. Rec.	Avg. F_1	Threshold	Avg. Prec.	Avg. Rec	Avg. F_1	Avg. F_1
Standard	2	0.93	0.92	**0.93**	0.30	0.96	0.95	0.95	**0.95**
					0.60	0.98	0.93	**0.96**	0.93
					0.90	1.00	0.90	0.94	0.80
Standard	3	0.89	0.90	**0.89**	0.30	0.82	0.91	0.84	**0.86**
					0.60	0.86	0.88	**0.86**	0.85
					0.90	0.89	0.81	0.84	0.62
Standard	4	0.87	0.92	**0.89**	0.30	0.81	0.92	0.85	**0.89**
					0.60	0.84	0.89	0.85	0.86
					0.90	0.90	0.83	**0.86**	0.65
Beer 3 runs	3	0.92	0.89	**0.91**	0.30	0.76	0.77	**0.77**	**0.79**
					0.60	0.88	0.67	0.76	0.68
					0.90	0.94	0.54	0.68	0.63

posterior samples, Gibbs sampling for PrecursorCluster requires 20 minutes to process one Standard run on an Intel Core i5, 3.3GHz PC. Runs are processed independently and can be parallelized. In Cluster-Match, the matching of IP clusters via MW has a time complexity of $O(m \log n)$ time, where n and m are the number of vertices and edges in the bipartite graph to be solved, translating to a wall clock of less than a minute for each run. Cluster-Cluster requires longer computational time. With 1000 posterior samples per top-level bin, the processing of 2 Standard runs requires approximately half an hour. Each top-level bin can also be processed in parallel.

5.6 Conclusions

We have proposed an integrative workflow that performs the precursor clustering of ionization product peaks and uses that to improve alignment. The PrecursorCluster model introduced is a data reduction process that can reduce the number of peaks to IP clusters based on a list of possible ionization transformation types. The clustering information extracted from PrecursorCluster can be used to improve other steps in the pipeline too. For instance, metabolite identification, currently the main bottlenecks in high-throughput metabolomics, might be improved through analyzing IP clusters as the objects of interest rather than individual peak features. In this chapter, the PrecursorCluster model is optimised on metabolomics data by focusing on the set of adduct transformations as ionization product transformations. However this does not preclude the model from being applied to other MS-based omics as

well. For instance, adduct peaks are less of a problem in proteomics data, and since peptide fragments being fragmented are larger than metabolites, the resulting proteomic spectra often contains more isotopic peaks. This rule on isotopic transformations can also be incorporated as part of PrecursorCluster.

One of the key assumption made in PrecursorCluster is that the peak with M+H transformation must be the peak having the largest intensity in the IP cluster (other peaks are not allowed to join the IP cluster if their intensity values are smaller than the M+H peak). While this is a reasonable assumption to make, some IP peaks do not obey this modelling assumption. An alternative clustering methods that are more flexible and does not have the intensity constraint can be considered, e.g. by allowing peaks to form IP clusters if they can be transformed (within tolerance) to any potential precursor mass that other peaks also jointly 'vote' for. The weight of a potential precursor mass can then be updated based on the likelihood of the set of peaks having valid transformation paths to that precursor mass.

Taking the MAP results from PrecursorCluster, we have also demonstrated how IP clustering can be used to improve alignment. Our results show that in comparison to the conventional direct-matching of peak features, the proposed approach Cluster-Match, which performs the matching of IP clusters and subsequently groups member peaks having the same IP types, allows us produce a better (more precise) alignment result. It is also noteworthy that while Cluster-Match still makes the assumption that across runs, correspondent IP clusters always exist, this assumption is more relaxed when it comes to the construction of the actual alignment of peak features. Since member peaks in matched IP clusters are grouped according to their IP types, peaks that do not have correspondent type across runs will never be matched together — even if they are close in distance and would otherwise has been matched in the conventional direct-matching scheme. In this manner, Cluster-Match can potentially produce fewer false positives in matching compared to the direct matching of peak features alone.

The proposed pairwise matching scheme used by Cluster-Match is frequently extended to the processing of multiple runs in a fairly ad-hoc manner and suffers from having to set a reference run (which can be considered another parameter to set). Producing a distance measure that works well for measuring similarities of peaks across multiple runs is non-trivial, in particular in the merged-match scheme from the sequential processing of pairs of runs. We propose the Cluster-Cluster method that addresses this issue by not requiring a reference run when constructing alignment through a second-stage clustering of IP clusters. To our knowledge, no literature has systematically evaluated the effect of choosing a different reference run for alignment or how changing the order of runs being processed might affect the alignment result, but we hypothesise that methods like Cluster-Cluster that does not require a reference run will have an advantage when aligning a large dataset — typical in modern large-scale metabolomics experiments having hundreds of runs to process.

In addition, most methods also do not take into account the uncertainties inherent in the matching of peaks across runs. Cluster-Cluster is able to return aligned peaksets at varying probabilities. Our experiments show that by setting a suitable threshold, we can extract from Cluster-Cluster results alignment results having a higher precision than what can be obtained from other methods. As future work, an interactive visualisation module can be developed to let user visualize ionization product clustering and aligned peaksets (with their probabilities) from a single graphical interface. Such module can be incorporated as part of a larger metabolomics pipeline.

A weakness of the alignment methods described in this chapter is the fact that as a second-stage clustering step, both Cluster-Match and Cluster-Cluster requires the MAP assignment of peaks into their IP clusters from PrecursorCluster. The complete uncertainties from PrecursorCluster are not propagated to the matching stage. The next chapter addresses this problem by introducing a fully-hierarchical model that performs the clustering of peaks within run and across runs at once.

Chapter 6

Hierarchical Clustering of LC-MS Peaks

6.1 Introduction

The Cluster-Cluster method introduced in Chapter 5 performs the direct-matching of peaks in ionisation product (IP) clusters that themselves have been clustered together. However, related peaks are assigned into IP clusters based on their *maximum a-posteriori* probabilities. In this chapter, we expand upon the idea of alignment as a hierarchical clustering problem by proposing **HDP-Align**, a Bayesian non-parametric model that groups related peaks within runs by their retention time (RT) and assigns them to global clusters shared across runs. Within each global cluster, peaks are further grouped by their m/z values into mass clusters, representing the various ionisation products derived from the global compound. In this manner, the local clusters in HDP-Align correspond to the within-file IP clusters from running PrecursorCluster on each run, while the global clusters in HDP-Align correspond to the top-level clusters produced from Cluster-Cluster (described in Chapter 5).

The proposed HDP-Align model introduced in this chapter allows us to infer the matching of peaks across all runs at once without the need for any intermediate merging of pairwise runs. Similar to the Cluster-Cluster model introduced in Section 5.3.2, the proposed model of HDP-Align also introduces the possibility of allowing the user to trade recall for precision from the alignment results by returning a smaller subset of the results having a higher confidence score of being correctly aligned. Figure 6.1 shows an illustration of the clustering process in HDP-Align. Additionally, the latent variables inferred in the model may correspond to chemically meaningful compounds and can be used for further analysis. Using a metabolomic dataset, we demonstrate the usefulness of such latent objects by using the mass clusters derived from the model and a set of defined ionisation product transformations to perform the putative annotations of peaks based on their potential adduct types and

metabolite identities.

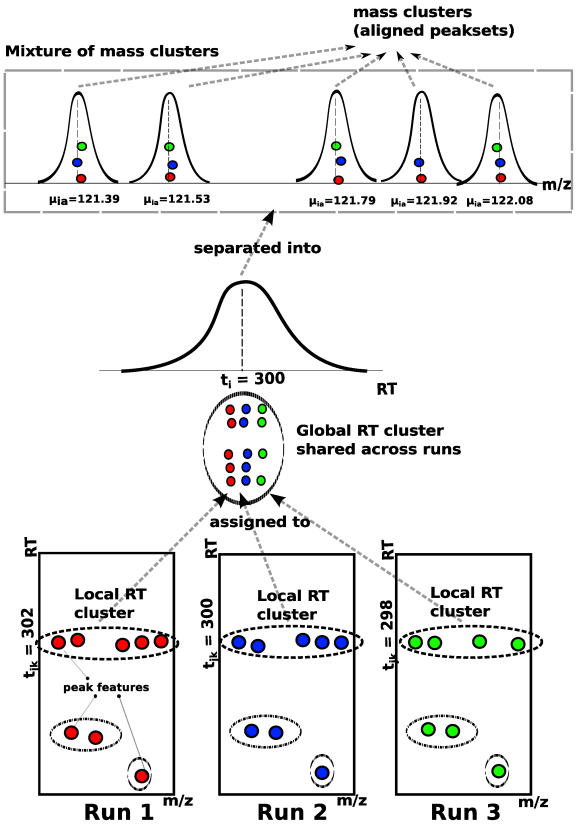


Figure 6.1: An illustrative example of how the proposed model in HDP-Align simultaneously **(1)** performs the clustering of related peaks into within-run local clusters by their RT values, **(2)** assigns the peaks to global RT clusters shared across runs, and **(3)** separates peaks into mass clusters, which correspond to aligned peaksets.

6.2 Related Work

The goal of establishing the matching of peaks across multiple runs at once can be viewed as a clustering problem, where a set of peaks can be grouped (by their m/z, RT and other suitable features) into local clusters within each run (representing all of the peaks from an individual compound), which are further grouped into global clusters shared across runs. Hierarchical clustering has been used for the matching of peaks across runs [119, 81]. In [119], peaks are hierarchically clustered based on their m/z values to construct matching across runs, while in [81], peaks from the entire dataset are pooled and a hierarchical clustering scheme based on RT only is used to group peaks into within-run local clusters, which are further grouped into across-run super clusters. Both approaches require choosing various user-defined parameters, such as determining a suitable cut-off for the dendogram produced, deciding on a suitable linkage method and defining an appropriate distance measure between groups of

peaks. In [119], no chromatographic separation is performed, so only the m/z values of peaks are used. The nature of the gas chromatography data used in [81], where retention time across runs is more reproducible, means that even without using the m/z information, good alignment performance can still be obtained. This will not be the case of LC-MS data, where retention time drift is common and the highly accurate m/z information is crucial for alignment. The proposed HDP-Align model fills this gap where both m/z and RT values, important for LC-MS peak alignment, are used for the hierarchical clustering process. The probabilistic approach employed by HDP-Align also allows us to extract confidence values from aligned peaksets.

6.3 Hierarchical Dirichlet Process Mixture Model for Alignment

The proposed model for HDP-Align is framed as a Hierarchical Dirichlet Process (HDP) mixture model [96]. Essential modifications to the basic HDP model, described in Section 3.4, were performed to suit the nature of the multiple peak alignment problem. Figure 6.2 shows the conditional dependencies between random variables in the HDP-Align model.

Our input consists of J input files, indexed by $j = 1, ..., J$, corresponding to the J LC-MS runs to be aligned. Each j-th input file contains N_j peaks in total, which can be separated into K_j local clusters of related-peaks. In a j-th file, peaks are indexed by $n = 1, ..., N_j$ and local clusters are indexed by $k = 1, ..., K_j$. Across all files, we assign each local cluster k in file j to a global cluster $i = 1, ..., I$, where I is the total number of global clusters, using the indicator variable v, as described in the following paragraph. A global cluster corresponds to the compound of interest during LC-MS analysis, e.g. metabolite or peptide fragment, that is present across runs, while local clusters are realisations of the global clusters in a specific run. Finally, within each global cluster i, we can further group peaks by their m/z values into A mass clusters (indexed by $a = 1, ..., A$). Each mass cluster therefore corresponds to the ionization product peaks coming from the different runs that are produced by a global compound during mass spectrometry.

We use the indicator variable $z_{jnk} = 1$ to denote the assignment of peak n in file j to local cluster k in that file. Similarly, $v_{jni} = 1$ if peak n in file j is assigned to global cluster i, and $v_{jnia} = 1$ if peak n in file j is assigned to mass cluster a linked to metabolite i. Let d_j be the list of observed data of peaks in file j, $d_j = (\mathbf{d}_{j1}, \mathbf{d}_{j2}, ..., \mathbf{d}_{jn})$ where $\mathbf{d}_{jn} = (x_{jn}, y_{jn})$ with x_{jn} the RT value and y_{jn} the log m/z value of the peak feature. The log of m/z value is here used as the m/z error is assumed to increase linearly with the observed m/z value [120]. $\boldsymbol{\theta}$ denotes the global mixing proportions and π_j the local mixing proportions for file j. The global mixing proportions $\boldsymbol{\theta}$ are distributed according to the Griffiths, Engen and McCloskey

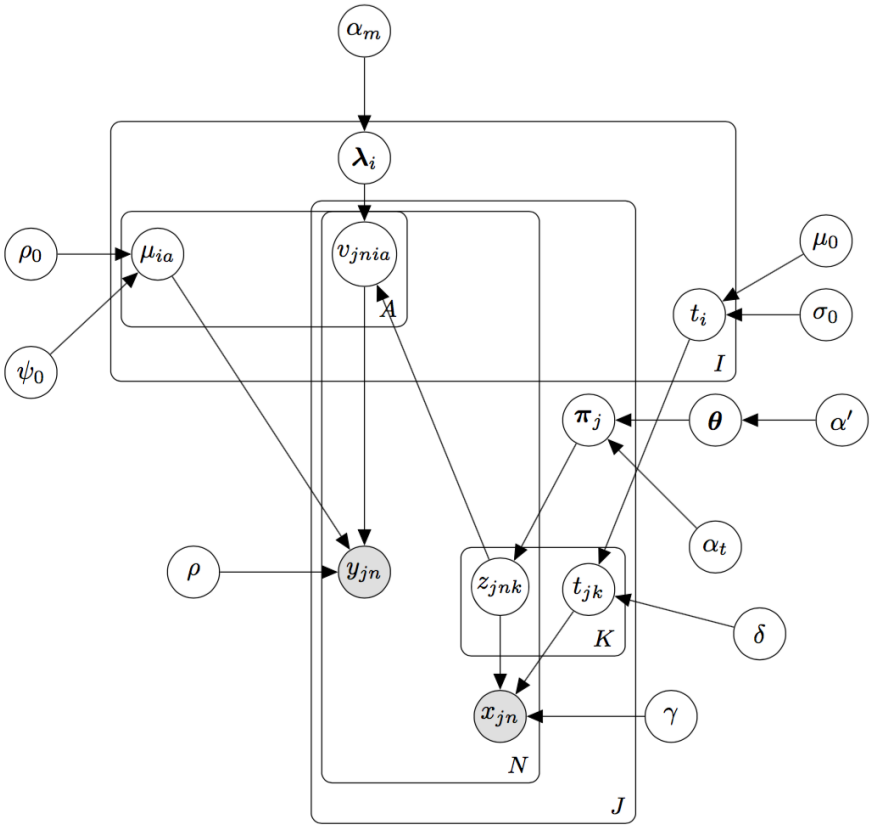


Figure 6.2: Graphical model for HDP-Align. x_{jn} is the observed RT value of peak n in file j, while y_{jn} is the observed m/z value.

(GEM) distribution:

$$\boldsymbol{\theta}|\alpha' \sim GEM(\alpha') \tag{6.1}$$

where the GEM distribution over $\boldsymbol{\theta}$ is described through the stick-breaking construction:

$$\beta_i \sim Beta(1, \alpha') \tag{6.2}$$

$$\theta_i = \beta_i \prod_{l=1}^{i-1}(1 - \beta_l) \tag{6.3}$$

The local mixing proportions π_j are distributed according to a Dirichlet Process (DP) prior with the base measure $\boldsymbol{\theta}$ and concentration parameter α_t.

$$\pi_j|\alpha_t, \boldsymbol{\theta} \sim DP(\alpha_t, \boldsymbol{\theta}) \tag{6.4}$$

Within each file j, the indicator variable $z_{jnk} = 1$ denotes the assignment of peak n in file j to local RT cluster k in that file. This follows the local mixing proportions for that file.

$$z_{jnk} = 1|\pi_j \sim \pi_j \tag{6.5}$$

The RT value t_i of a global mixture component is drawn from a base Gaussian distribution with mean μ_0 and precision (inverse variance) σ_0.

$$t_i|\mu_0, \sigma_0 \sim \mathcal{N}(\mu_0, \sigma_0^{-1}) \tag{6.6}$$

The RT value t_{ij} of a local mixture component in file j is normally distributed with mean t_i and precision δ. The precision controls how much RT values of related-peak groups across runs are allowed to deviate from the parent global compound's RT.

$$t_{jk}|t_i, \delta \sim \mathcal{N}(t_i, \delta^{-1}) \tag{6.7}$$

Finally, the observed peak RT value is normally distributed with mean t_{jk} and precision γ. The precision controls how much RT values of peaks can deviate from their related-peak group.

$$x_{jn}|z_{jnk} = 1, t_{jk}, \gamma \sim \mathcal{N}(t_{jk}, \gamma^{-1}) \tag{6.8}$$

The m/z value produced through high-precision MS instrument is highly accurate, and its correspondence is often preserved across runs. Once peaks have been assigned to their respective global clusters, we need to further separate peaks within each global cluster into mass clusters to obtain the actual alignment. These mass cluster corresponds to ionisation products. We do this by incorporating an internal DP mixture model on the m/z values (y_{jn}) within each global cluster i. Let the indicator $v_{jnia} = 1$ denotes the assignment of peak n in

file j to mass cluster a in the i-th global cluster. Then:

$$\begin{aligned}
\boldsymbol{\lambda}_i|\alpha_m &\sim GEM(\alpha_m) && (6.9)\\
v_{jnia}=1|\boldsymbol{\lambda}_i &\sim \boldsymbol{\lambda}_i && (6.10)\\
\mu_{ia}|\psi_0,\rho_0 &\sim \mathcal{N}(\mu_{ia}|\psi_0,\rho_0^{-1}) && (6.11)\\
y_{jn}|v_{jnia}=1,\mu_{ia} &\sim \mathcal{N}(\mu_{ia},\rho^{-1})\cdot I(\mathbf{d}_{jn}) && (6.12)
\end{aligned}$$

where the index ia refers to the a-th mass cluster of the i-th global cluster. $\boldsymbol{\lambda}_i$ is the mixing proportions of the i-th internal DP mixture for the masses, with α_m the concentration parameter. μ_{ia} is the mass cluster mean, drawn from the Gaussian base distribution with mean ψ_0 and precision ρ_0. The observed mass value is drawn from a Gaussian distribution with the component mean μ_{ia} and precision ρ, for which the value is set based on the MS instrument's resolution. Additionally, we add an additional constraint on the likelihood of y_{jn} using the indicator function $I(\cdot)$ such that $I(\mathbf{d}_{jn})=1$ if there are no other peaks inside the mass cluster that come from the same file as the current $\mathbf{d}_{jn}$ peak, and 0 otherwise. This constraint captures the restriction that a peak feature can only be matched to other peaks from different files, reflecting the assumption that within each LC-MS run, compounds produce ionisation products with distinct mass-to-charge fingerprints that can be used for matching to other runs.

6.4 Inference

Inference within the model is performed via a Gibbs sampling scheme, allowing us to compute the alignment probabilities through the proportion of posterior samples in which any sets of peaks are placed in the same mass component (a) in the same top-level cluster. In this manner, peaks coming from different runs that are in the same mass component are considered to be aligned as they have similar RT and m/z values. In each iteration of the sampling procedure, we instantiate the mixture component parameters for the local RT cluster (t_{jk}) and global RT cluster (t_i) in the mixture model. In the internal DP mixture linked to each global cluster i, we marginalise out the mass cluster parameters (μ_{ia}). The initialisation step of the sampler is performed by assigning all peaks in each run into a single local RT cluster. Across runs, these local clusters are assigned under a global cluster shared across runs. Within a global cluster, peaks coming from different runs are assigned to a single mass cluster. The sampler than iterates through each peak feature, removing it from the model, updating the assignment of peaks to clusters and performing the necessary book-keeping on any instantiated mixture components. Further details on the specific Gibbs update statements can be found in following sections.

6.4.1 Updating peak assignments

We use the following variables to denote the count of items in any clustering object: c_{jk} is the number of peaks in a local cluster k of file j. c_i is the number of local clusters in a global cluster i, and c_{ia} is the number of peaks in a mass cluster a inside a global RT cluster i. To update the assignment of a peak $\mathbf{d}_{jn}$ to local RT cluster k during Gibbs sampling, we need the conditional probability of $P(z_{jnk} = 1)$ given every other parameters, denoted as $P(z_{jnk} = 1|\mathbf{d}_{jn}, \ldots)$.

$$P(z_{jnk} = 1|\mathbf{d}_{jn}, ...) \propto \begin{cases} c_{jk} \cdot p(\mathbf{d}_{jn}|z_{jnk} = 1, ...) \\ \alpha_t \cdot p(\mathbf{d}_{jn}|z_{jnk^*} = 1, ...) \end{cases} \tag{6.13}$$

We consider the top and bottom terms of eq. (6.13) separately in the following.

1. The likelihood of the peak $\mathbf{d}_{jn}$ to be in an existing local RT cluster k, $p(\mathbf{d}_{jn}|z_{jnk} = 1, ...)$ is proportional to c_{jk}. This is assumed to factorise across the RT (x_{jn}) and mass (y_{jn}) terms

$$p(\mathbf{d}_{jn}|z_{jnk} = 1, ...) = p(x_{jn}|z_{jnk} = 1, ...) \cdot p(y_{jn}|z_{jnk} = 1, ...) \tag{6.14}$$

   The RT term $p(x_{jn}|z_{jnk} = 1, ...)$ in eq. (6.14) is normally distributed with mean t_{jk} and precision γ, while the mass term $p(y_{jn}|z_{jnk} = 1, ...)$ is an internal DP mixture of mass components linked to the parent global cluster i of an existing local cluster k. We then marginalise over all mass clusters in i to get $p(y_{jn}|z_{jnk} = 1, v_{jni} = 1...)$

$$\begin{aligned} p(y_{jn}|z_{jnk} = 1, v_{jni} = 1...) &= \sum_a \frac{c_{ia}}{\alpha_m + \sum_a c_{ia}} p(y_{jn}|v_{jnia} = 1, ...) \\ &+ \frac{\alpha_m}{\alpha_m + \sum_a c_{ia}} p(y_{jn}|\boldsymbol{v}_{jnia^*} = 1, ...) \end{aligned} \tag{6.15}$$

   To compute the terms in eq. (6.15), first we consider the case for an existing mass cluster a in the global RT cluster i. Then,

$$p(y_{jn}|v_{jnia} = 1, ...) = \mathcal{N}(\mu_{ia}, \rho^{-1}) \tag{6.16}$$

   For a new mass cluster a^* in the global RT cluster i, we marginalise out μ_{ia} to obtain

$$p(y_{jn}|\boldsymbol{v}_{jnia^*} = 1, ...) = \mathcal{N}(\psi_0, \rho^{-1} + \rho_0^{-1}) \tag{6.17}$$

2. The likelihood of the peak $\mathbf{d}_{jn}$ to be in a new local cluster k^* is proportional to α_t. Marginalising over all global clusters in the top-level DP, we get

$$p(\mathbf{d}_{jn}|z_{jnk^*}=1,...) = \sum_i \left[\frac{c_i}{\alpha' + \sum_i c_i} p(\mathbf{d}_{jn}|v_{jni}=1,...)\right] + \frac{\alpha'}{\alpha' + \sum_i c_i} p(\mathbf{d}_{jn}|\boldsymbol{v}_{jni^*}=1,...) \tag{6.18}$$

There are two terms to compute in eq. (6.18): whether peak $\mathbf{d}_{jn}$ is in an existing global cluster i or a new global cluster i^*. For an existing global RT cluster i in eq. (6.18), $p(\mathbf{d}_{jn}|v_{jni}=1,...)$ is assumed to factorise into its RT and mass terms, so $p(\mathbf{d}_{jn}|v_{jni}=1,...) = p(x_{jn}|v_{jni}=1,...)\cdot p(y_{jn}|v_{jni}=1,...)$. We marginalise over all local RT clusters to obtain

$$p(x_{jn}|v_{jni}=1,...) = \mathcal{N}(x_{jn}|t_i, \gamma^{-1}+\delta^{-1}) \tag{6.19}$$

and marginalise over all possible mass clusters in the internal DP linked to global cluster i to obtain $p(y_{jn}|v_{jni}=1,...)$. This is defined in eq. (6.15). Finally, for a new global RT cluster i^* in eq. (6.18), $p(\mathbf{d}_{jn}|\boldsymbol{v}_{jni^*}=1,...)$ is also assumed to factorise into its RT and mass terms. Then, we marginalise over t_{jk} and t_i to obtain

$$p(x_{jn}|\boldsymbol{v}_{jni^*}=1,...) = \mathcal{N}(x_{jn}|\mu_0, \sigma_0^{-1}+\gamma^{-1}+\delta^{-1}) \tag{6.20}$$

and marginalise over μ_{ia} to get

$$p(y_{jn}|\boldsymbol{v}_{jni^*}=1,...) = \mathcal{N}(y_{jn}|\psi_0, \rho^{-1}+\rho_0^{-1}) \tag{6.21}$$

6.4.2 Updating instantiated variables

The following expressions are used to update the instantiated mixture component parameters in the model during Gibbs sampling.

1. Updating global cluster's RT t_i: here, $t_{jk\in i}$ refers only to local RT clusters currently assigned to the global cluster i, and c_i is the count of such peaks. Then

$$p(t_i|...) \propto p(t_i|\mu_0, \sigma_0^{-1}) \prod_j^J \prod_k^K p(t_{jk\in i}|t_i, \delta) = \mathcal{N}(\mu_i, \gamma_i^{-1}) \tag{6.22}$$

where $\mu_i = \frac{1}{\gamma_i}\left[\mu_0\sigma_0 + \delta\sum_j\sum_k t_{jk\in i}\right]$ and $\gamma_i = \sigma_0 + \delta c_i$.

2. Updating local cluster's RT t_{jk}: here, $x_{jn\in k}$ refers only to peaks currently assigned to the local RT cluster k, and c_{jk} is the count of such peaks.

$$p(t_{jk}|...) \propto p(t_{jk}|t_i, \delta^{-1}) \prod_{j}^{J} \prod_{n}^{N} p(x_{jn\in k}|t_{jk}, \gamma) = \mathcal{N}(\mu_k, \gamma_k^{-1}) \quad (6.23)$$

where $\mu_k = \frac{1}{\gamma_k} \left[t_i\delta + \gamma \sum_j \sum_n x_{jn\in k} \right]$ and $\gamma_k = \delta + \gamma c_{jk}$.

6.4.3 Using the Inference Results

Using the posterior samples from Gibbs sampling, we can compute various posterior summaries and more interestingly, extract the alignment of peaks from the inference results (since features assigned into the same mass cluster within the same global RT cluster are considered to be aligned). For each sample, we record the aligned peaksets of peaks put into the same mass cluster. Averaging over all samples provides a distribution over these aligned peaksets. Note that across all the aligned peaksets from all samples, it is possible for the same peak to be matched to different partners with varying probabilities, depending on how often they co-occur together in the same mass cluster. To allow the possibility of controlling precision and recall from the results, we provide another user-defined threshold t, where aligned peaksets are returned only when they occur with matching probability $>t$. In the same manner as the Cluster-Cluster method in the previous chapter, varying this threshold allows the user to use HDP-Align to trade precision for recall: a low value for t gives a larger set of results that are potentially less precise, while conversely a high t provides a smaller, more precise set of aligned peaksets. This is an output not available from the other baseline alignment methods and can potentially be useful in problem domains where high precision is required from the alignment results.

6.4.4 Isotopic Product and Metabolite Identity Annotations

As described in Section 2.3.4, in metabolomics studies using electrospray ionisation, a single metabolite can generate multiple ionisation products peaks, (such as isotopic variants, adducts, fragment peaks), alongside other peaks resulting from noise and artifacts introduced during mass spectrometry [2]. Determining and annotating these IP peaks are desirable to remove extraneous peaks and reduce the burden of subsequent analysis in the data processing pipeline. Additionally, deducing the precursor mass of the compound that generates the IP peaks is necessary to query compound library databases in order to assign metabolite identities.

The resulting clustering objects inferred from HDP-Align lend themselves to further analysis in a natural fashion, as global RT clusters in HDP-Align may correspond to metabolites, while local RT clusters may correspond to the noisy realisations of these metabolites within each run. Mass clusters in the internal mixture of each global cluster could correspond to the ionisation products of a metabolite. To demonstrate the possibility of obtaining additional information beyond alignment from the output of HDP-Align, we follow the workflow in [2] that performs IPs and metabolite annotations of peaks. This workflow is composed of multiple key steps: peak matching, ionisation product clustering and metabolite mass matching. A key difference of HDP-Align to the workflow in [2] lies in the fact that HDP-Align is able to perform the two separate steps of peak alignment and potential IP clustering simultaneously, as shown in Figure 6.3.

Given the set of potential IP clusters, we can perform IP annotations on the peaks. To do this using the metabolomic dataset, first we take the set of clustering objects produced in a posterior sample. For each mass cluster, we assign its m/z value to be the average m/z values of features assigned to it, denoted by m. The list of common adducts (Table 4.3) in positive ionisation mode is used to compute the inverse transformation for the precursor mass that generates the observed mass. Following [2], any two mass clusters sharing the same precursor mass (within tolerance) provide a vote on the presence of that consensus precursor mass. The mass clusters and peaks inside them can be annotated with the adduct type that produces the transformation type to the shared precursor mass. The set of precursor masses deduced in this manner can also be used to query KEGG (a database of metabolite compounds) in order to assign metabolite identities to the global compound. Note the difference from the PrecursorCluster method in the previous chapter. In PrecursorCluster, the list of possible IP transformation rules are specified in advance as a prior information that guides the clustering process, while in HDP-Align, we take the resulting clustering objects and attempt to match them to the list of transformations.

6.5 Evaluation Study

6.5.1 Evaluation Datasets

The performance of the proposed methods and other benchmark methods is evaluated on the LC-MS datasets for the proteomic, glycomic and metabolomic experiments first introduced in Section 4.4. Unlike the PrecusorCluster model in Chapter 5 that requires a list of transformation rules (generally specific to each -omics) to be specified, the HDP-Align model is more flexible as it does not require such rules to group peaks across runs that share close m/z values together.

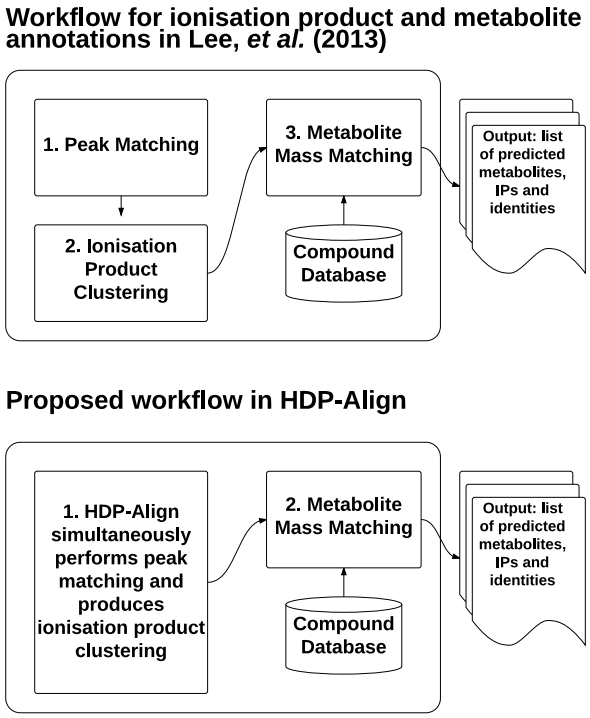


Figure 6.3: Comparisons on the workflow to assign putative annotations on isotopic products and metabolites described in Lee. *et al.* (2013) [2] and in HDP-Align.

As before, all 6 fractions from the P1 Proteomic dataset in [33] are used. Each fraction contains 2 runs of features having high RT variations across runs are used in our experiments. Unlike Section **??** where only pairs of runs used, here we use the first 10 runs of the Glycomic dataset provided by [1] for our multiple-runs experiment. Additionally, the Standard metabolomic dataset, first introduced in Section **??**, is also used. Here, we selected 6 runs for our experiment. Table 6.1 summarises the different evaluation datasets and the number of features each has.

Dataset	No. runs	Total Features
P1 Frac 000	2	10606
P1 Frac 020	2	2135
P1 Frac 040	2	2188
P1 Frac 060	2	3342
P1 Frac 080	2	2086
P1 Frac 100	2	1326
Glycomic	10	9344
Metabolomic	6	7477

Table 6.1: Total number of runs and features of the selected evaluation datasets.

6.5.2 Baseline Methods for Evaluation

Alignment performance is evaluated following the definition of precision and recall in Section 5.4.2. To summarise, every alignment method return a list of aligned peaksets. Each aligned peakset can be broken into all the possible l-size combinations of peaks in the peakset, with each combination constituting an alignment item. From a method we obtain the set of alignment items, M, while from the ground truth, we obtain the set of alignment items G. True Positives ($\boldsymbol{TP}$) are items that should be matched (in G) and are actually matched (in M). False Positives ($\boldsymbol{FP}$) are items that should not be matched (not in G) but are matched (in M), while False Negatives ($\boldsymbol{FN}$) are items that should be matched (in G) but are not matched (in M). A method with a perfect alignment output would have both precision and recall values of 1.0.

Following Chapter 4, we benchmark HDP-Align against two established alignment methods: SIMA [36] and MZmine2's Join Aligner [19]. The selection of SIMA and Join as the baseline methods is motivated by the fact that both methods are direct matching methods (thus easily comparable to HDP-Align) that work on generally any LC-MS-based omics data. They also sufficiently differ in how they establish the final alignment results, in particular when it comes to the alignment of multiple runs. This is primarily due to the differences between both methods in the form of the distance/similarity function between peaks, the actual matching algorithm itself and the merging order of pairwise results to construct the full alignment results. The two most important parameters to configure in the baseline methods are the mass and RT tolerance parameters, used for thresholding and computing feature similarities during matching. We label these common parameters as the $T_{(m/z)}$ and T_{rt} parameters. Note that despite the common label, each method may use the parameter values differently during the alignment process. In our experiments, we let $T_{(m/z)}$ and T_{rt} vary within reasonable ranges (details in Section 6.5.3) and report all performance values generated by each combination of the two parameters.

6.5.3 Parameter Optimisations

Tables 6.2 and 6.3 describe the parameter ranges of each method during performance evaluation. For HDP-Align (Table 6.2), we perform the experiments based on our initial choices on the appropriate parameter values. These are almost certainly less than optimal and can be optimised further. The mass cluster standard deviation $\sqrt{\rho^{-1}}$ for HDP-ALign is set to the equivalent value in parts-per-million (ppm). These are 500 ppm for the Proteomic dataset and 3 ppm for the Glycomic and Metabolomic datasets. The local (within-run) cluster RT standard deviation $\sqrt{\gamma^{-1}}$ is assumed to be fairly constant and set to 2 seconds for all datasets, while the global cluster standard deviation $\sqrt{\delta^{-1}}$ is set in the following dataset-specific

manner: 50 seconds for the Proteomic dataset and 20 seconds for the remaining datasets. The larger standard deviation value is required for the Proteomic dataset to accomodate for greater RT drifts across runs. Other hyperparameters in HDP-Align are fixed to the following values: $\alpha' = 10$, $\alpha_t = 10$, $\alpha_m = 100$. The values of the precision hyperparameters for global cluster RT (σ_0) and mass cluster (ρ_0) are set to a broad value of 1/5E6. No significant changes were found to the results when these hyperparameters for the DP concentrations and cluster precisions were varied. The mean hyperparameters μ_0 and ψ_0 are set to the means of the RT and m/z values of the input data respectively. During inference, 10000 posterior samples were obtained with the first 5000 used as burn-in, and taking every 10-th sample after burn-in for the posterior probabilities of peaks to be matched.

Dataset	HDP
P1 Frac 000	$\sqrt{\rho^{-1}} = 500$ ppm, $\sqrt{\gamma^{-1}} = 2$ s, $\sqrt{\delta^{-1}} = 50$ s
P1 Frac 020	
P1 Frac 040	
P1 Frac 060	
P1 Frac 080	
P1 Frac 100	
Glycomic	$\sqrt{\rho^{-1}} = 3$ ppm, $\sqrt{\gamma^{-1}} = 2$ s, $\sqrt{\delta^{-1}} = 20$ s
Metabolomic	$\sqrt{\rho^{-1}} = 3$ ppm, $\sqrt{\gamma^{-1}} = 2$ s, $\sqrt{\delta^{-1}} = 20$ s

Table 6.2: Parameters used for HDP-Align

For SIMA and Join, we report the results from all combinations of the mass and RT tolerance parameters within reasonable ranges listed in Table 6.3. This follows from the range of parameters selected for evaluation experiments in the previous Chapter 4. The ranges of $T_{(m/z)}$ and T_{rt} parameters used are based values reported on [33] for the Proteomic dataset and [1] for the Glycomic dataset. For the Metabolomic dataset, they were chosen in light of the mass accuracy and RT deviations of the data.

Dataset	Benchmark (SIMA, Join)
P1 Frac 000	$T_{(m/z)} = \{1.0, 1.1, ..., 2.0\}$, $T_{rt} = \{10, 20, ..., 180\}$ s
P1 Frac 020	
P1 Frac 040	
P1 Frac 060	
P1 Frac 080	
P1 Frac 100	
Glycomic	$T_{(m/z)} = \{0.05, 0.1, 0.25\}$,$T_{rt} = \{5, 10, ..., 120\}$ s
Metabolomic	$T_{(m/z)} = \{0.001, 0.01, 0.1\}$, $T_{rt} = \{5, 10, ..., 120\}$ s

Table 6.3: Parameters used for the benchmark methods (SIMA, Join).

6.6 Results and Discussions

The performance of the evaluated methods methods on the different datasets are presented in Sections 6.6.1 and 6.6.2. Additionally, an example of the further annotations for the putative adduct type and metabolite identity that can be produced by HDP-Align is also shown in Section 6.6.2.

6.6.1 Proteomic (P1) Results

Figure 6.4 shows the results from performance evaluation on the Proteomic (P1) dataset. We see that both benchmark methods (SIMA and Join) produce a wide range of performance depending on the parameter values for $(T_{(m/z)}, T_{rt})$ chosen. Sensitivity to parameter values is expected on this dataset due to the low mass accuracy in the MS instrument that produces the data and the high RT drifts present across runs (further details in [33]). HDP-Align performs well on several fractions (particularly fractions 040, 060, 080, 100) with precision-recall performance close to the optimal performance attainable by the benchmark methods. On all fractions, HDP-Align is also able to produce higher-precision results compared to the benchmark methods by reducing recall through setting the appropriate values for the threshold t. The primary benefits of quantifying alignment uncertainties is realised here as the well-calibrated probability scores on the matching confidence of aligned peaks produced by HDP-Align allows the user to choose which point along the PR curve to operate on. It is less obvious how this can be accomplished in the benchmark methods by varying the RT (T_{rt}) and m/z ($T_{m/z}$) thresholding parameters, if at all possible.

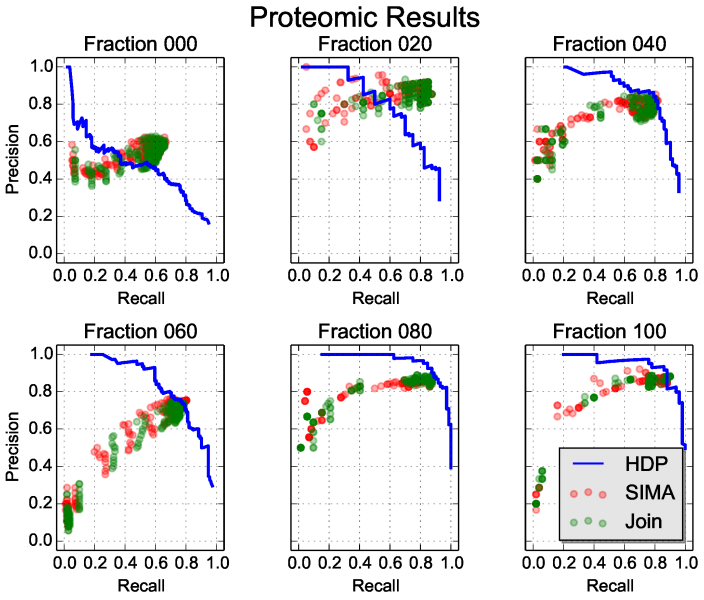


Figure 6.4: Precision-recall values on the different fractions of the Proteomic (P1) dataset.

The P1 datasets also represent the most challenging alignment scenario as they have the largest RT drift and low mass accuracy in comparison the glycomic and metabolomic data. We use the largest (P1 Fraction 000) and the smallest (P1 Fraction 100) from P1 to examine how well our chain converges during Gibbs sampling.

Figure 6.5 (left) shows the traceplots of the number of global clusters from three randomly initialised MCMC chains when running HDP-Align on the largest P1 Fraction 000 dataset containing 10606 features. From the jumps in the number of global clusters in the traceplots, we see some evidence of bad mixing in the chains. This is explained by the fact that in our sampler, we do not allow for the block reassignment of a group of peaks (that are together placed in a local cluster) into a new global cluster. Consequently, a global cluster can only be deleted when all its individual peaks have completely moved elsewhere. This leads to the slow convergence and poor mixing of the model. An inspection of the distributions of global clusters after burn-in from the three chains, shown in Figure 6.5 (right), also suggests that the chains have not fully converged yet.

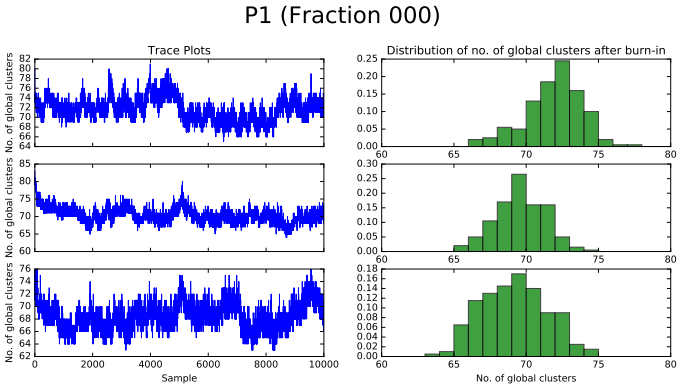


Figure 6.5: Traceplots from three randomly initialised MCMC chains for the number of global clusters across all posterior samples for the largest fraction (000) from the Proteomic (P1) dataset.

Running HDP-Align on the P1 Fraction 000 data and collecting 10000 posterior samples requires several days of walltime. The main factor affecting the running time of HDP-Align is the total number of peaks across all runs to be processed and the number of samples produced during Gibbs sampling. In each iteration of Gibbs sampling, HDP-Align removes a peak from the model, updates parameters of the model conditioned on every other parameters, and reassigns a peak into RT and mass clusters. In practice, additional time will also be spent on various necessary book-keeping operations, such as deleting empty local and global clusters that are no longer required, updating internal data structures, etc. Running a longer chain may not be not entirely practical in actual analytical situation — particularly in comparison to the speed of the baseline methods that completes in minutes. Despite this poor mixing, as the results show in Figure 6.4, we still see some evidence that by reducing recall,

it is still possible to extract from HDP-Align alignment results having a higher precision than what the baseline methods can achieve.

Inspecting the diagnostic plots Figure 6.6 for the smaller P1 Fraction 100 dataset, we observe a better mixing behaviour. The traceplots that are less jumpy and the distributions of the number of global clusters that are more consistent across the three chains. This may be due to the smaller (1326) number of features in this dataset. On this dataset, HDP-Align took two hours to process. This is a reasonable time a user can tolerate for a data analysis pipeline to complete (although still significantly longer than the baseline methods that require seconds to complete). Again from Figure 6.4, the results for this P1 Fraction 100 dataset show that by trading off recall, we can obtain a higher precision than the baseline methods.

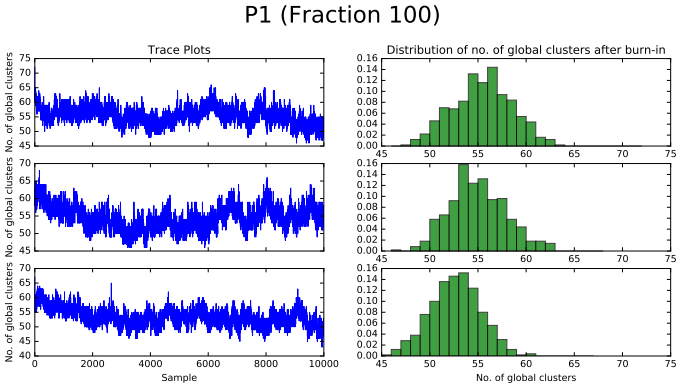


Figure 6.6: Traceplots from three randomly initialised MCMC chains for the number of global clusters across all posterior samples for the smallest fraction (100) from the Proteomic (P1) dataset.

6.6.2 Glycomic and Metabolomic Results

Figures 6.7 and 6.8 show the results from experiments on the Glycomic and Metabolomic datasets. Similar to the Proteomic dataset, a range of precision-recall values can be observed in the results for the benchmark methods on the two datasets. Consistent with our expectation, reducing the tolerance window on the retention time produces a smaller recall value, however this does not necessarily result in a better alignment precision. The performance of HDP-Align, using the same set of parameters on both datasets, come close to the optimal results from the benchmark methods, while still allowing the user to control the desired point along the precision-recall curve to operate on.

The results for the Glycomic dataset (Figure 6.7) also show some additional results on how the measured precision-recall values might change depending on the strictness of what constitutes an alignment item during performance evaluation. This is accomplished by gradually increasing the value for l that determines the size of the feature combinations enumerated

from a method's output. For example, l=2 considers all pairwise combinations of features from the method's output during performance evaluation, while $l = 4$ considers all combinations of size 4, and so on. Figure 6.7 shows that as l is increased, parameter sensitivity seems to become more of an issue for the benchmark methods, with more parameter sets having lower precisions in the results. Across all ls evaluated, parameter pairs that produce the best alignment performance (points with high precision and recall values) are generally small $T_{(m/z)}$ and large T_{rt} values. Examples of parameter pairs that produce the best and worse performance for SIMA are shown in Figure 6.8. The results here appear to suggest the importance of having high mass precision during matching. Importantly, we see from Figure 6.7 that the performance of HDP-Align remains fairly consistent as l is increased.

The Metabolomic dataset also provides us with additional results in form of annotations of putative adduct type and metabolite identities. A thorough evaluation on the quality of such annotations, in comparison to e.g. the workflow proposed in [2], is beyond the scope of this chapter and would likely necessitate using a different and more appropriate evaluation dataset. Instead, we present an example of the further analysis performed by HDP-Align (as proposed in Section 6.4.4) on the resulting clustering objects after inference. Figure 6.9 shows a global RT cluster where peaks across runs have been grouped by their RT and m/z values. Within this global cluster, peaks are further separated into 6 mass clusters – corresponding to ionisation products produced by the global cluster during mass spectometry. In Figure 6.9, mass cluster A and B contain features aligned from several runs but they do not have any other mass cluster sharing a possible precursor mass. Mass cluster C and D share a common precursor mass (292.12696) and can thus be annotated by the adduct type that produce the transformation. Similarly, mass cluster E and F share a common precursor mass at 383.14278. Queries to a local KEGG database are issued based on the precursor mass values, producing several compound identities that can be putatively assigned to the global RT cluster. It is a strength of the HDP-Align approach that this putative identification step appears very naturally from the alignment results.

6.7 Conclusion

In this chapter, we present HDP-Align, a hierarchical non-parametric Bayesian model that simultaneously performs the within-run clustering and across-run clustering of peaks. As a natural consequence of the clustering process, the direct matching of peaks can be extracted from the model. In addition, the clustering objects from the model can also be used for further analysis in the pipeline as they potentially correspond to the actual chemical compounds that generate the generate. Similar to the two-stage clustering methods (Cluster-Cluster) introduced in the previous chapter, the HDP-Align model is able to produce well-calibrated

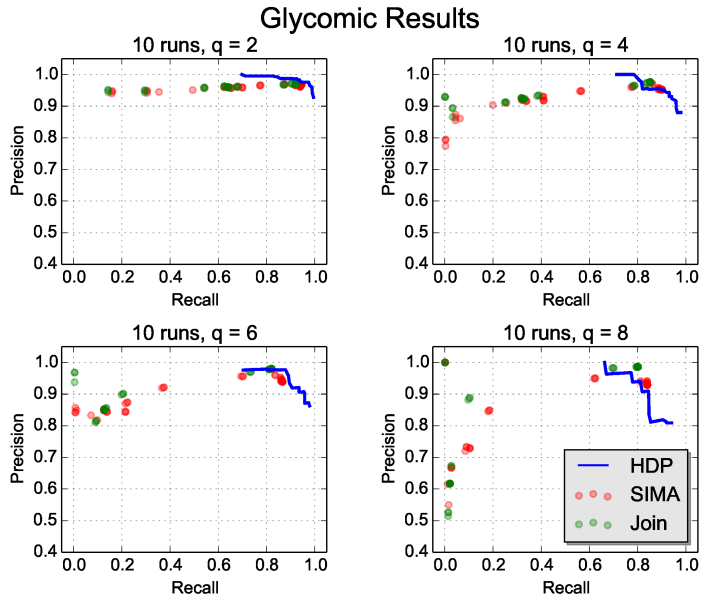


Figure 6.7: Precision-recall values on the alignment of 10 runs from the Glycomic dataset when q (the strictness of performance evaluation as described in Section 5.4.2) is gradually increased.

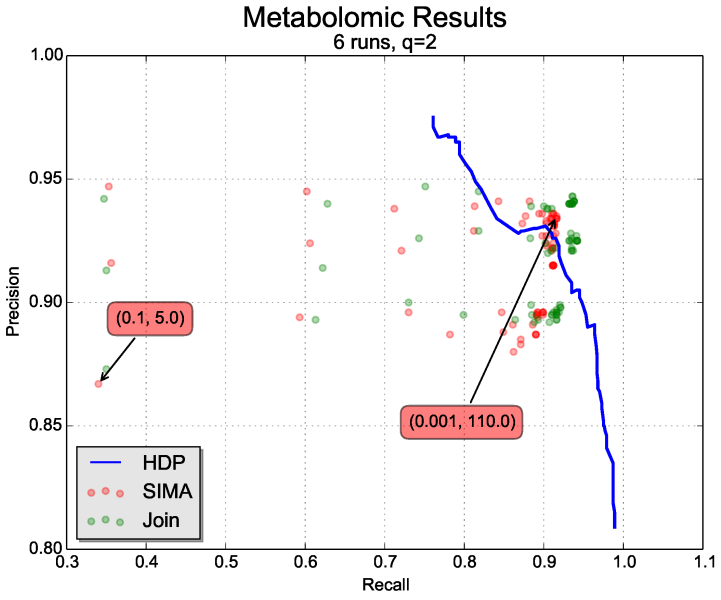


Figure 6.8: Precision-recall values on the alignment of 6 runs from the Metabolomic dataset. The parameter values $(T_{m/z}, T_{rt})$ that produce the best and worst performance in SIMA are also annotated in the Figure (red boxes).

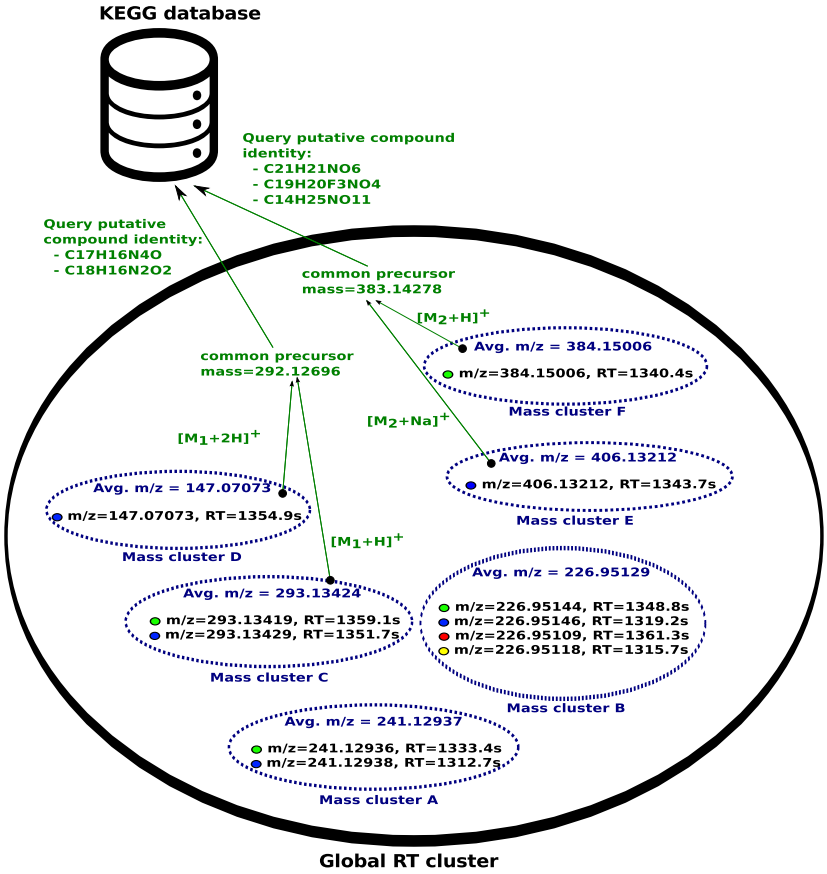


Figure 6.9: Example analysis that can be performed on the clustering objects inferred in a Gibbs sample from HDP-Align. The outer black oval denotes a global RT cluster (generally corresponding to a metabolite compound), while the smaller dotted ovals within denote mass clusters (labelled as mass cluster A, B, C, D, E, F). peaks are denoted by the filled circle, with the fill colour indicating the originating run of a peak. Green colours denote additional analysis steps that can be performed on the mass cluster objects.

probability scores on the matching confidence of aligned peaks (evidenced by the increasing precision and decreasing recall as the threshold t is increased). This is accomplished by casting the multiple alignment problem of LC-MS peaks as a hierarchical clustering problem. Matching confidence can be obtained based on the probabilities of co-eluting peaks to be placed under the same mass component in the same global cluster. Experiments based on datasets from real proteomic, glycomic and metabolomic experiments show that HDP-Align is able to produce alignment results competitive to the benchmark direct-matching alignment methods, with the added benefit of being able to provide a measure of confidence in the alignment quality. This can be useful in real analytical situations, where neither the optimal parameters nor the alignment ground truth is known to the user.

A primary weakness of HDP-Align lies in the long computational time required to produce results. This is due to the slow mixing of the chains, as the consequence of our incremental Gibbs sampling that samples one variable at a time. The split-and-merge MCMC algorithm for the HDP proposed in [121] may help to improve sampling performance and is an avenue for future work. The actual running time for the sampling can also be improved by taking the lessons from the Cluster-Cluster approach introduced in the previous chapter, for instance it may be possible to partition the data into subsets of peaks based on their retention time as only peaks within a certain RT tolerance should ever be clustered and matched to each other. The key insight of the HDP-Align model lies in the way related IP peaks are modelled as within-file clusters in a single run but the model also allows these within-file clusters to be generated by globally-shared clusters spanning multiple runs. The results presented in the current chapter suggest the method shows enough promise to warrant the effort to speed it up.

Additional sources of information present in the LC-MS data, such as chromatographic peak shapes, can also be used to improve alignment performance and subsequent analyses that follow. The mixture of mass components used in HDP-Align with a more appropriate mass model, such as that in MetAssign [57] that specifically takes into account the inter-dependency structure of peaks. Alternatively, it may be possible to build the transformation rules employed by the PrecursorCluster model (from the previous chapter) into HDP-Align. However, such modifications will introduce even more complexity to an already complex model, requiring a more sophisticated inference scheme and perhaps an even longer running time.

Through comparisons against benchmark methods, our studies have also investigated the effect of sub-optimal parameter choices on alignment performance. While beyond the scope of our paper, we agree with [42, 41] that thorough investigations into the influence of numerous configurable parameters (prevalent in nearly all LC-MS data processing pipeline) on the resulting biological conclusions are of utmost importance. This should be followed by the development of methods to minimise or automatically-tune such configurable parameters.

Despite the abundance of new methods proposed for LC-MS data pre-processing, relatively few studies have been done on the subject of quantifying uncertainties and alleviating the burden of parameter optimisations during actual data analysis. One way to minimise the number of parameters is through the integration of multiple steps in the typical LC-MS pipeline into fewer steps. Our proposed model in HDP-Align can potentially be extended in this manner, as evidenced by the metabolomic dataset results where we directly use the clustering objects inferred from the model to perform further analysis on putative adduct and metabolite type annotations. While the proposed annotation approach in Section 6.4.4 is fairly simple, it can be easily extended to more sophisticated annotation strategies, such as in CAMERA [47]. This will be particularly useful when we aim to extend the proposed model in HDP-Align into a single inferential model that encompasses many intermediate steps in a typical LC-MS data processing pipeline.

Our experiments of taking the clustering objects from HDP-Align and using them to assign metabolite identities to the clusters through matching to a compound database also shows the potential of HDP-Align in assisting compound identification by allowing identifying labels to be assigned to a group of matched peaks from several runs at once. However, identification of metabolites, particularly in large-scale untargeted experiments, is challenging. The next chapter explores this in greater details and proposes the use of a different type of structural information, present in mass spectrometry fragmentation data, to improve identification.

Chapter 7

Substructure Discovery in Tandem Mass Spectometry Data

7.1 Introduction

As the results from Chapter 5 shows, the ionization product (IP) types of many observed peaks are often unknown and therefore the molecular mass of metabolites that generate these peaks are also unknown. This makes identification difficult as mass is often a required information when querying metabolite identities against publicly-available databases, such as KEGG [59] and PubChem [122]. In addition, while modern mass spectrometry instruments can be highly accurate up to 3 parts-per-million (ppm), even a mass accuracy of 1 ppm is not sufficient to reliably determine the elemental composition (formula) of a metabolite [123] during database queries. The presence of isomers (metabolites having the same formula and mass but are structurally different from each other) suggests that when relying on mass alone, the same peak might be incorrectly matched to multiple isomeric metabolites. Retention time (RT) might help to distinguish certain isomers that have different elution profiles, but RT drift, a main challenge in alignment, means observed RT values can vary across different chromatographic platforms and cannot be easily used as a characteristic information in public databases during identification. Apart from the small number of metabolites present in a standard solution that can be identified with a high degree of confidence (as they produce measured peaks having reliably known m/z and RT values), information on the mass and RT values alone are not enough to establish the identity of many metabolites in untargeted studies.

Fragmentation spectra are the results of chaining two stages of mass spetrometry steps. In data-dependent acquisition, a precursor or parent (MS1) peak is selected according to a certain criteria, frequently the top-N most intense peaks in a scan, for further fragmentation. This produces for each fragmented parent peak a distinct pattern of fragment (MS2) peaks.

Fragmentation patterns can be used to aid identification through the matching of a query spectrum to a database of reference spectra. In recent years, a growing number of fragmentation spectra databases have been made public, including METLIN [124], ChemSpider [61] and MassBank [60]. However, mass spectral databases are not comprehensive and contain only a small number of known metabolites. The large variance in submitted spectra further limits potential matches as sensible results can only be obtained when matching spectra generated from measurement platforms having similar characteristics (for e.g., produced through the same ionization method under a similar mass accuracy). According to [53], approximately 2% of spectra in an untargeted metabolomics experiment can be matched and subsequently identified – a small number in contrast of the vast collection of metabolites that comprise the metabolic pathways of an organism.

Multiple metabolites can share the same chemical substructure. For example, carboxylic acid (Figure 7.1) is a generic substructure shared by many amino acids and organic acids, such as acetic acid (CH_3COOH) that is commonly found in vinegar or butyric acid ($CH_3(CH_2)_2COOH$) that is present in butter. During fragmentation in positive ionization mode, the neutral carboxyl group (COOH) breaks from the parent ion, forming CO and H2O (the extra hydrogen in H2O comes from the addition of a positively charged proton, H+, during ionization). From this, we can expect to observe a characteristic neutral loss in the spectra of metabolites that share carboxylic acid as a substructure. In a fragmentation spectrum, this will be represented by a fragment peak that is 46 Da smaller than the mass of the parent peak. Thinking generatively, observing a neutral loss of 46 Da therefore provides a hint that a fragmentation spectrum is generated from the measurement of a metabolite that contains carboxylic acid as a substructure.

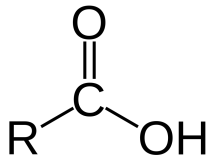


Figure 7.1: The carboxylic acid substructure. In the diagram, R refers to the residue, which is the rest of the metabolite attached to this substructure.

As illustrated by the very simplified example above, the knowledge of the constituent substructures that comprise a metabolite, particularly of the larger and more specific substructures, can be used to provide a hint as to the overall identity of the metabolite. Classification method, such as Support Vector Machine, decision tree and neural networks [125, 126, 86, 87], have been trained to learn spectral features that represent substructures and predict the presence or absence of substructures from fragmentation spectra. Combined with information from the parent peak (such as the m/z, RT values and IP types if available), this provides additional information that can aid in the identification of metabolites that cannot be resolved

through the traditional method of spectral database matching alone.

A common shortcoming of these classification approaches higlighted before is the need of the supervised training of the classifier (classification-based approaches may fail to generalise well to new dataset produced from different analytical platforms). Based on the assumption that fragmentation spectra contain fragment peaks that represent shared substructures of metabolites, we propose a workflow that applies the Latent Dirichlet Allocation (LDA) model to spectral fragmentation data. The proposed workflow produces the decomposition of fragmentation spectra (equivalently a document in standard LDA) into the set of *Mass2Motifs* (equivalently a topic in standard LDA). Here, a Mass2Motif is defined to be the recurring set of fragment peaks and neutral losses that potentially correspond to a biochemically-relevant substructure shared by many metabolites. Unlike the classification-based methods highlighted earlier, the decomposition of fragmentation spectra into Mass2Motifs is achieved in an unsupervised manner. The MS2LDA workflow is introduced in Section 7.4.

7.2 Related Work

Clustering is commonly used for group fragmentation spectra that are similar to each other. Clusters of spectra can be used for identification by forming a consensus spectrum and matching it against spectral databases. Molecular networking clusters MS1 peaks by their MS2 spectral similarity such that one identifiable metabolite in a cluster facilitates structural annotation of its neighbors [127, 128, 129]. However, only MS2 spectra with high overall (e.g. cosine) spectral similarity are grouped in Molecular Networking. Consequently Molecular Networking may fail to group molecules that share small substructures. In particular, spectra may be placed in different clusters if they share a small number of fragment peaks that related to a common substruture, but their overall global similarities are too different. Even for spectra placed into the same cluster, often manual analysis (by eyes) is required to select the characteristic fragment peaks that represent a potential substructure and are shared by members of the clusters. Another package, MS2Analyzer [130] mines MS2 spectra given the prior knowledge on the fragment patterns of interest to be specified in advance. While generic features, such as CO or H2O losses, will be common to many experiments, sample-specific features can be easily overlooked if they have not been specified *a priori*.

The assumption that spectral consist of building blocks that correspond to substructures is alluded in certain works but not directly mined from the data. Prior knowledge on substructures have been used for the annotations of a small number of moelcules in fragmentation data [131] and for metabolite classification in GC-MS [132, 126]. In CSI:FingerID [87], a fragmentation tree is used to predict (using Support Vector Machine) the molecular 'fingerprint', computed through the implicit assumption that fragments share substructures, of an

unknown compound. The resulting fingerprint is used to improve the matching of spectrum of the unknown compound against a vast chemical database (PubChem). Implicit in these methods are the assumption that recurring patterns of fragment peaks and neutral losses values explain the presence of common biological substructures (e.g. a hexose unit, or a CO loss) shared by metabolites.

Latent Dirichlet Allocation has not been applied to metabolomics or mass spectrometry data, but it has been applied to other fields of computational biology in e.g. genomics [133], metagenomics [134], and transcriptomics [99]. In [133], DNA sequence from genomics studies is decomposed into recurring patterns of N-mers nucleotides. A topic in this context corresponds to the set of N-mers (e.g. 'ATGC' as an instance of a 4-mers) that co-occur together across the different genomic sequences of a species, and the objective of the study is characterise the sets of N-mers that corresponds to conserved genes of the species. Similarly in [134], a metagenomic read (essentially a DNA sequence) is decomposed into its topic distribution. The unsupervised decomposition of metagenomic reads into topic distributions is used to improve the binning (clustering) of reads from the same species. In [99], a sample or gene from transcriptomics studies is decomposed into multiple processes in a manner similar to how a document is decomposed into different topics in traditional LDA for text.

7.3 Statement of Original Work

The work discussed in this chapter, of which the author is a joint first author, has been submitted to the *Proceedings of the National Academy of Sciences* and is under review. Justin van der Hooft (JvdH) performed the measurements of the Beer samples through mass spectometry, generating the set of fragmentation data that can be used for topic modelling. The author contributed to the design and development of the MS2LDA workflow. This includes the development and optimisation of the feature extraction process, the implementation and testing of inference via LDA and also model validation against multinomial mixture model.

JvdH then analysed the results from MS2LDA for biochemical significance. Throughout the analysis, the author and JvdH worked closely together. To assist JvdH in his analysis, the author proposed and developed the visualisation module, MS2LDAVis. To improve the visualisation module, the author integrated elemental formula annotation functionalities. This includes writing a wrapper in MS2LDA to call SIRIUS [135], a Java-based elemental formula annotator. Cristina Mihailescu (CM), an Msc student, implemented another Python-based elemental formula annotator, which was also customised and integrated into MS2LDA by the author.

JvdH then performed molecular networking analysis on the same dataset, which was used for comparison to MS2LDA results. The author performed the identification of metabo-

lites through matching to reference standard compounds and also the differential analysis of Mass2Motifs, and with JvdH validated the results.

7.4 A Workflow for Substructure Discoveries and Annotations

Substructure discovery through the MS2LDA workflow consists of two stages: i) the data conversion stage, which prepares the acquired fragmentation data into suitable input format for the workflow, followed by ii) the Mass2Motif discovery stage, which performs topic modelling via LDA to discover mass fragmental patterns, assigns potential candidate elemental formulae to MS1 and MS2 peaks, and visualises the Mass2Motifs in an interactive environment. The key insight of MS2LDA lies in emphasising the parallel between text and mass spectrometry fragmentation data (Figure 7.2A-B). As a text analysis pipeline relying on LDA to decompose documents into topics based on frequently co-occurring words, so MS2LDA decomposes fragmentation spectra into their constituent building blocks of frequently co-occurring fragments and neutral losses (referred to as Mass2Motifs). The complete workflow is illustrated in Figure 7.2C.

Acquired fragmentation data cannot readily be used for the purpose of pattern searching via LDA and has to be converted into a suitable format. As input, the MS2LDA workflow accepts the combination of a single full-scan file for the MS1 peaks and a separate fragmentation file for the MS2 peaks. The data conversion process starts with the detection of MS1 peak in the input .mzXML file obtained from full-scan mode spectra using the CentWave algorithm from the XCMS library [30]. This constitutes information on the parent (MS1) level. Fragmentation data, in the form of .mzML file obtained from tandem MS mode, are processed using an R script based on the RMassBank package [136].

A linking step is required to match the most intense MS2 spectrum in a scan to a parent MS1 peak. Matching is performed via a greedy search within a specified retention time tolerance window, selecting the top few most intense peaks for the matching. This simulates the generative process that produces the spectral data in data-dependant experiments. A filtering step, based on RT and intensity, is applied to remove noisy peaks. Any MS1 peak not having paired MS2 peaks is also discarded for further processing. The aim of the filtering step is to exclude identical fragmentation spectra produced by low-intensity MS1 peaks that were fragmented multiple times, potentially forming spurious and uninformative Mass2Motifs on their own.

Following the bag-of-words assumption, LDA does not consider word orders but instead take into account only the number of times word co-occur in a document. The next step is trans-

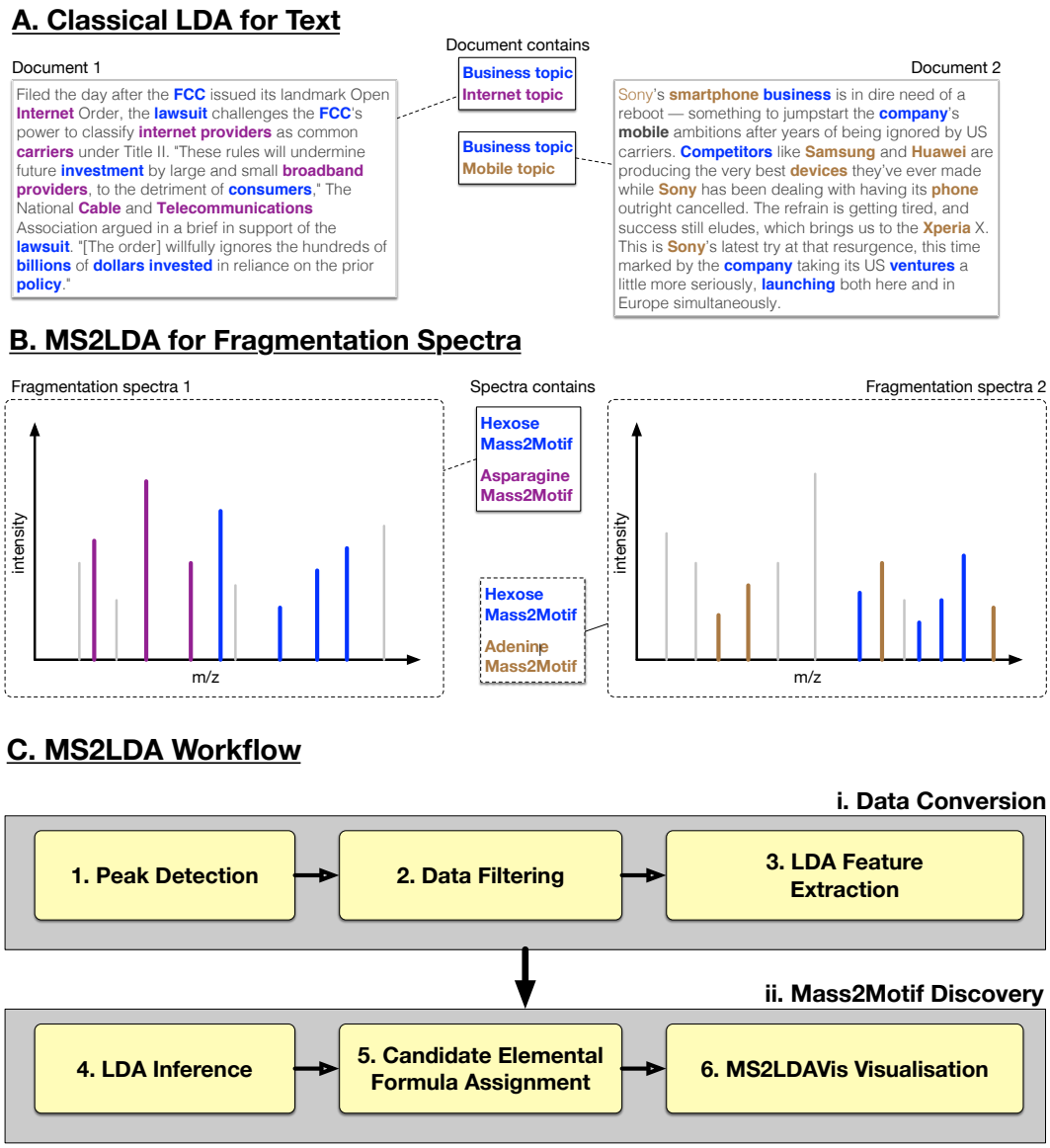


Figure 7.2: **A.** LDA applied to text decomposes a document into its topic distributions (e.g. football, business and enrivonment topics). **B.** Similarly, MS2LDA decomposes a fragmentation spectrum into its topics (Mass2Motifs) that can be characterised as asparagine, hexose and adenine related. Each fragmentation spectra comprise of one or more Mass2Motifs. **C.** Schematic overview of the MS2LDA workflow.

forming spectral data into a bag-of-word count matrix (illustrated in Figure 7.3), with entries in the matrix the co-occurrences of discrete MS2 features ('words') in the fragmentation spectra linked to a parent MS1 peak ('document'). From each fragmentation spectra, two types of features can be extracted: fragment features and loss features. Fragment features are the discretised m/z values of MS2 peaks, while loss features are formed by discretising neutral losses. A neutral loss, defined as the mass differences between a precursor MS1 peak and each of the child MS2 peaks in the spectrum, corresponds to the removal of a specific neutral fragment from the molecular ion.

MS1 feature extracted →

Binned MS2 features ↓

	MS1_a	**MS1_b**	**MS1_c**	**MS1_d**	**MS1_e**	**…**
Fragment_119.0351	0	100	24	37	0	
Fragment_136.0629	87	0	17	18	0	
Fragment_156.0769	55	20	0	10	100	
…						
Loss_18.0080	56	0	0	10	15	
Loss_36.0183	0	0	30	0	0	
Loss_46.0053	40	40	10	87	100	
…						

Figure 7.3: The matrix of co-occurrences of fragment and loss features (rows) in each fragmentation spectrum linked to a parent MS1 peak (columns). Entries of the matrix are the counts of the feature from the normalized (0 100 scale) intensities.

Discretisation is performed via a greedy binning process. To group continuous m/z values and create fragment features, a priority queue is used that efficiently maintains the ordering of m/z values of peaks upon insertion. Successive items are popped from the priority queue in ascending order, forming a group of contiguous features — until the next encountered item has an m/z value larger by a predefined tolerance in parts-per-million from the average values of the group, in which case a new group is created. The average m/z values of a group, rounded to 5 decimal places, becomes the discrete representation of fragment peaks in their originating spectra. The count of a fragment feature in a spectrum is computed by dividing the MS2 peak's intensity value to the largest intensity in the spectrum and multiplying by an scaling factor of 100 (equivalent to the discretisation resolution). In this manner, MS2 peaks with larger intensity values are represented more often in the spectra. Neutral loss features are discretised and computed in a manner similar to fragment features. The resulting matrices for fragment and loss features are concatenated and used as input to LDA.

In the context of fragmentation data, the standard LDA model as applied to substructure discovery is described next. The observation on the n-th fragment or loss feature in the d-

th fragmentation spectra (w_{dn}) is conditioned on the assignment of feature w_{dn} to the k-th Mass2Motif multinomial distribution. This corresponds to the topic distribution over words in the original LDA model. This assignment is denoted by the indicator variable z_{dn}, so $z_{dn} = k$ if feature w_{dn} is assigned to a k-th Mass2Motif. The k-th multinomial distribution that a feature is assigned to is characterised by the parameter vector $\boldsymbol{\phi}_{z_{dn}}$, with $\boldsymbol{\phi}_{z_{dn}}$ drawn from a prior Dirichlet distribution with a symmetric parameter β.

$$w_{dn}|\boldsymbol{\phi}_{z_{dn}} \sim Multinomial(\boldsymbol{\phi}_{z_{dn}}) \tag{7.1}$$

$$\boldsymbol{\phi}_k|\beta \sim Dir(\beta) \tag{7.2}$$

The probability of seeing certain Mass2Motifs for each d-th fragmentation spectra is drawn from a multinomial distribution with a parameter vector $\boldsymbol{\theta}_d$, corresponding to the topic decomposition of a document in the original LDA model. This parameter vector $\boldsymbol{\theta}_d$ is in turn drawn from a prior Dirichlet distribution having a symmetric parameter α.

$$z_{dn}|\boldsymbol{\theta}_d \sim Multinomial(\boldsymbol{\theta}_d) \tag{7.3}$$

$$\boldsymbol{\theta}_d|\alpha \sim Dir(\alpha) \tag{7.4}$$

A collapsed Gibbs sampling scheme is implemented in Python for inference (details in Section 3.5). The output from inference is a set of Mass2Motifs and assignments of Mass2Motifs to each MS1 peak.

MS2LDAVis

Given its hypothesis-generating nature, the analysis of Mass2Motifs to characterise and examine their correspondence to actual biochemical substructures is an iterative and exploratory process. This is made possible through the MS2LDAVis module, an interactive web-based visualisation build upon the combination of the Javascript and the D3 library (http://d3js.org). MS2LDAVis is extended from the Python port of the topic modelling visualisation interface LDAVis [137] used in the text domain, but our adaptation MS2LDAVis introduces fragmentation-specific views.

Similar to the original LDAVis, the left panel of MS2LDAVis module shows a global view of the model, whilst the right panel zooms into a specific Mass2Motif (see Figure 7.4A). However, unlike LDAVis where topics are displayed on the left panel through multidimensional scaling that projects topics to two dimensions, the two axes in MS2LDAVis panel are the log-degree and the h-index of Mass2Motifs. The degree of a Mass2Motif as the number of fragmentation spectra explained by the Mass2Motif at the user-defined threshold t_θ on the fragmentation-spectra-to-Mass2Motif distributions ($\boldsymbol{\theta}$). The h-index of a Mass2Motif is

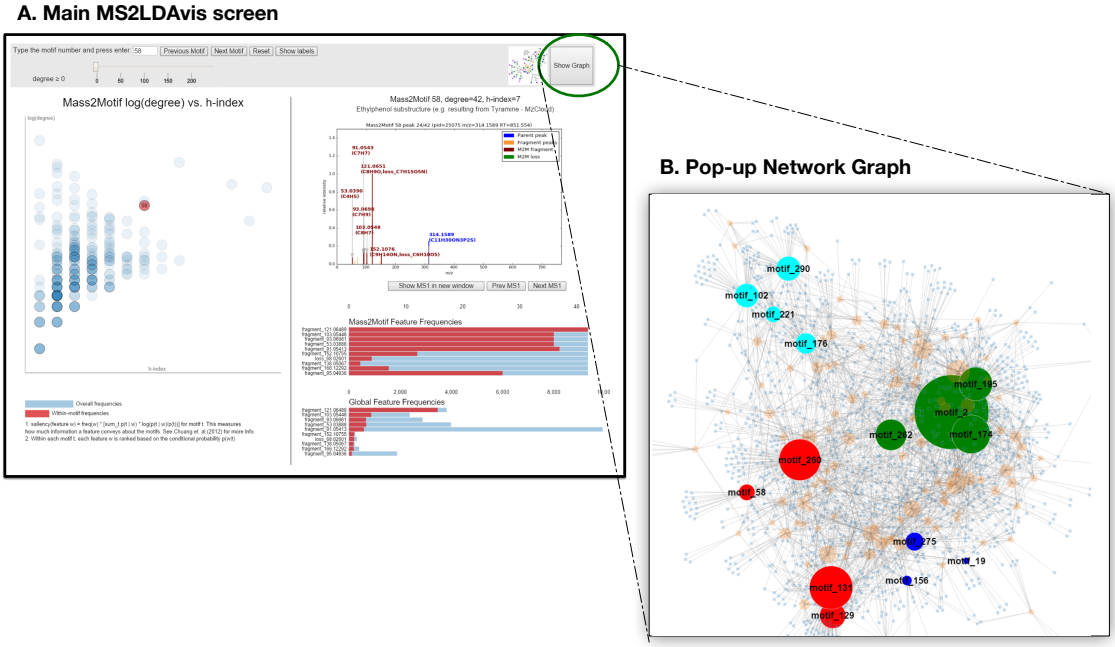


Figure 7.4: Screenshot of MS2LDAVis. See text for explanations of the different panels.

defined in a similar manner to the conventional h-index for scientific publications of a researcher. A Mass2Motif has an index of h if it has h fragment or loss features obtained after setting a user-defined threshold t_ϕ on the Mass2Motif-to-features distributions (ϕ), each of which occur in the set of thresholded spectra at least h times. Intuitively, a Mass2Motif with high degree but low h-index may potentially correspond to simple substructures that occur in many fragmentation spectra, while a Mass2Motif with high h-index but low degree are more unique and complex substructures shared by fewer MS2 spectra.

Selecting a Mass2Motif on the left panel of Figure 7.4A changes the specific information displayed on the right panel. Fragmentation spectra that can be explained by the currently selected Mass2Motif (above the threshold t_θ) are plotted, and clicking the Previous MS1 and Next MS1 buttons allows the flipping through consecutive spectra plots. Fragment and loss features that can be explained by the selected Mass2Motif (above the threshold t_ϕ) that also occur in the plotted spectra are highlighted in bold. Two barplots can be found on the bottom right panel: the Mass2Motif Feature Frequencies displays the counts of each fragment or loss features within the entire fragmentation spectra explained by the currently selected Mass2Motif, while the Global Feature Frequencies displays the counts of the fragments or loss features within the complete data set that can be explained by the currently selected Mass2Motif.

To complement the main visualisation view, inference results can also be visualised in a pop-up network graph (Figure 7.4B) by clicking the Show Graph button. In the network

view, Mass2Motifs and fragmentation spectra, represented by their parent MS1 peaks in the graph, form the nodes in the graph, and edges are drawn between the nodes if a spectra can be explained by a Mass2Motif above the threshold t_θ. To minimise clutter in the graph, a slider is provided to filter nodes based on their degree values. Nodes in the graph can also be annotated and coloured according to user specifications before the visualisation interface is called. The two complementary views are linked such that clicking a Mass2Motif node on the network graph will select the corresponding Mass2Motif on the main view and vice versa. The network graph is particularly useful in exploring the relationships between Mass2Motifs and investigating which spectra can be explained by multiple Mass2Motifs.

To aid data interpretation, putative elemental formulae is displayed on the plots of fragmentation spectra explained by a certain Mass2Motif (top-right panel, Figure 7.4). Two methods are integrated within MS2LDA to assign candidate elemental formulae. SIRIUS [135] employs a dynamic programming approach, termed 'Round Robin' [138], to solve elemental formula assignment as an integer decomposition problem. SIRIUS is freely-available and, as it is written in Java, can in theory be run platform-independently on any Windows, Unix and Mac environment (in practice, library dependencies have to be satisfied before SIRIUS can run). Integration of SIRIUS into the MS2LDA workflow is achieved by wrapping calls to the Java classes of SIRIUS through a separate sub-process, passing it a temporary MGF file that corresponds to a fragmentation tree. SIRIUS assigns elemental formulae to each fragmentation tree independently, which may lead to mass fragments of similar m/z value being assigned an elemental formula in some spectra, but not in all.

As an alternative to elemental formula annotation via SIRIUS, CM developed EF-Assigner, a pure Python implementation of an elemental formula assigner based on the Round Robin algorithm on which SIRIUS is based on. In EF-Assigner, candidate formulae are filtered using an implementation of the 7-golden rules, a set of heuristic rules introduced in [123] to remove chemically-unlikely elemental formula compositions from the candidate list. The advantages of EF-Assigner are its easy integration with the rest of the workflow (it is also written in Python) and it can assign elemental formulae to an entire group of MS2 peaks as represented by their discrete fragment and loss features at once. Unlike SIRIUS that uses the complete information of the precursor ion and fragments peaks in a spectrum for annotation, EF-Assigner assigns the elemental formulae for the MS1 peaks and MS2 fragment and loss features independently. The author included EF-Assigner in the MS2LDA workflow, passing it the necessary MS1 peaks and MS2 fragment and loss features for annotation. EF-Assigner is also modified to limit the maximum atom occurrences of certain elements in a candidate formula. For a greater annotation coverage, a second stage process is implemented. After an initial pass of EF-Assigner using a list of common chemical elements of CHNOPS, unannotated MS1 peaks and MS2 features are then re-annotated using an expanded list of possible elements that includes less common elements, such as the C-13 isotope of Carbon, Fluorine

and Chlorine.

7.5 Evaluation Study

7.5.1 Evaluation Dataset

To evaluate MS2LDA, four beer samples representative of complex mixtures of diverse biochemically relevant compound classes (such as amino acids, nucleotides, and sugars) typical in metabolomics studies are used. The beer extracts, acquired from one home-brewed beer and three different commercially available beers, are shown in Table 7.1. One of the beer samples (Beer3) is also used for the evaluation of the alignment methods in Chapter 4. Approximately 10 ml of beer was sampled from each bottle directly after opening. As well as the four individual extracts, a pooled aliquot of the four beer extracts was prepared. A Thermo Scientific Ultimate 3000 RSLCnano liquid chromatography system, coupled to a Thermo Scientific Q-Exactive Orbitrap mass spectrometer comprise the overall LC-MS setup.

Following mass spectrometry, blank runs, quality control samples, and 3 standard mixes containing 150 reference compounds were run to assess the quality of the mass spectrometer and aid in metabolite annotation and identification [54]. The pooled sample was run prior to and across the batch to monitor the stability and quality of the LC-MS runs. Beer samples were run in a randomized order. Immediately after acquisition, all RAW files containing information stored in a proprietary vendor-dependant format were converted into the open mzXML format. Mass spectra are centroided and separated into positive and negative ionization modes using the command line version of MSconvert (ProteoWizard). Fragmentation files were also converted into .mzML formats using the GUI version of MSconvert. Accurate masses of standards were obtained well within 3 ppm accuracy and intensities of the quality control samples (a beer extract and a serum extract) were as expected.

Label	Source
Beer1	A home-brewed bottle of German Wheat Beer.
Beer2	A bottle of ‘Jaw Glyde Ale brewed by JAW Brew.
Beer3	A bottle of ‘Seven Giraffes Extraordinary Ale brewed by William Bros. Brewery Company.
Beer4	A bottle of ‘Black Sheep Ale brewed by Black Sheep Brewery.

Table 7.1: A list of the Beer samples used for evaluation.

7.5.2 Model Comparison

We performed model selection via a 4-folds cross validation approach on one of the data file (Beer3 positive ionization mode). For each test fold being held out in the Beer3 data file, an estimate of the model evidence is computed after training the model on the remaining training folds in the file. The number of Mass2Motifs was also selected in this manner from cross-validation.

A crucial difference between LDA and the multinomial mixture-model (clustering) lies in the modelling assumption that a document is a mixture of one or more topics (LDA) as opposed to each document having exactly one topic (clustering). To validate one of our key assumptions of Mass2Motifs represent biological building blocks (i.e. fragmentation spectrum contains more than one Mass2Motifs), we compared the LDA model to a multinomial mixture model that can also be used for the clustering of fragmentation spectra. A comparison of LDA to a multinomial mixture model was performed by the author to assess and validate model fit, evaluated based on perplexity on the held-out data. Perplexity measures how well a probability distribution or probability model predicts a sample. First we define some notations. Let $\boldsymbol{w}_d$ denotes the d-th spectra, comprised of the entire set of fragment and loss features in that spectra. During cross-validation, spectra are divided into training and testing folds. We denote the set of spectra in the testing fold as $\boldsymbol{W}_t = \{\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_d\}$. The perplexity of $\boldsymbol{W}_t$ is given by:

$$perplexity(\boldsymbol{W}_t) = exp\left(\frac{\sum_d log(P(\boldsymbol{w}_d)}{\sum_d N_d}\right)$$

Here, $P(\boldsymbol{w}_d)$ is the marginal probability of a testing spectra $\boldsymbol{w}_d$ after integrating over all the parameters of the model. This is approximated via an importance sampling method in [139]. N_d is the number of features in each testing spectra $\boldsymbol{w}_d$. Following [140], the hyper-parameters were set to $\alpha = K/50$ and $\beta = 0.1$ during cross-validation. For mixture model clustering, a non-informative Dirichlet prior ($\alpha = K/50$, where K is now the number of clusters) is set on the proportions of the mixture components and another Dirichlet prior ($\beta = 0.1$) is set on cluster-specific word distributions. The Gibbs sampler for LDA and multinomial mixture model is run for 1000 samples, discarding the first 500 for burn-in. The last sample is used computing the posterior estimates. Minimal differences were found when the inferred model parameters were averaged over samples in comparison to simply using the last sample.

7.5.3 Biochemical Analysis

JvdH performed analyses on each of the beer samples described in Section 7.5.1. Each beer sample was processed independently of the others through MS2LDA. The aim of the analysis was to structurally characterized and annotate any chemically-relevant Mass2Motifs that potentially correspond to actual substructures shared by metabolites.

Validation to Reference Standard Molecules

Mixtures of known standard molecules were run along the beer extracts. On the beer data, the resulting accurate of these standards molecules were within 3 ppm accuracy, making their identification possible. As the identity of these molecules is known, we can use them to validate our structurally annotated Mass2Motifs. Given the database of exact mass and RT values of the standard molecules, a simple greedy matching scheme is used to establish the identity of MS1 parent peaks in MS2LDA. For each database entry of a standard molecule, we loop over all MS1 peaks in MS2LDA finding peaks that match the accurate mass of the standard molecule within the mass tolerance of 3 ppm and RT tolerance of 5 seconds. If there are multiple candidate MS1 peaks, the peak nearest in mass to the database accurate mass is selected. As these identified MS1 peak have linked spectra that are explained by characterised Mass2Motifs, this allows JvdH to validate the consistency of characterised Mass2Motifs against the identification information of reference standard molecules.

Comparison to Spectral Clustering

Molecular Networking [127, 128, 129] analysis can be used to compare inferred Mass2Motifs from MS2LDA against the clusters produced through the cosine clustering of fragmentation spectra. Spectral clustering (molecular networking analysis) of the four Beer samples was performed by JvdH using the **G**lobal **N**atural **P**roducts **S**ocial (GNPS) environment. The resulting fragmentation spectra for each Beer's .mzXML file was clustered using the MS-Cluster module with a precursor mass tolerance of 0.25 Da and a MS/MS fragment ion tolerance of 0.005 Da. Clustered fragmentation spectra originating from different files are merged to create the consensus spectra (consensus spectra containing less than 2 spectra were discarded). A graph network is created where nodes are consensus spectra and edges are drawn if the cosine similarities between nodes are above 0.55. For identification, spectra in the graph were searched against GNPS' spectral libraries, with a cosine threshold of 0.6 and having at least 4 matched fragment peaks. The resulting graph was exported into Cytoscape and visualised using the FM3 graph layout. Comparison against MS2LDA results were performed manually by JvdH. We also examined the data to find exemplar spectra that can be used to highlight the differences between MS2LDA results and spectral clustering.

Differential Analysis of Mass2Motifs

By linking MS2LDA analysis with the fold changes of MS1 peaks, the differential expression of Mass2Motifs can be assessed. This allows for the comparison of biochemical changes across groups of samples based on which metabolites can be explained by a Mass2Motif. As we hypothesise that more fragmentation spectra can be explainable by MassMotifs — in comparison to the number of spectra that can be annotated or identified through conventional matching to spectral library — the presence of shared substructures can reveal a shared pattern of differential expression among the set of metabolites explained by a Mass2Motif. This is possible even if these metabolites do not share a large degree of overall spectral similarity, which is often a necessary prerequisite in the identification of groups metabolites that share the same substructure.

The full-scan (MS1) LC-MS run for each Beer extract was processed using an in-house metabolomics pipeline based on XCMS [30] and MzMatch [48]. A peak table, containing information on the MS1 peak intensities, was exported to .csv files and linked to the parent MS1 peaks in MS2LDA through a greedy matching scheme that establishes the correspondence of parent peaks in MS2LDA to the MS1 peaks in the exported peak table within a specified m/z and RT tolerance values (3 ppm, 30 seconds). If there are multiple possible matches, the one with the nearest m/z difference is selected. Following this, for each Mass2Motif, a matrix is constructed where each row is a linked MS1 peak that can be explained by that Mass2Motif and the columns are intensity values from the different case and control groups. This matrix is used as input to our implementation of PLAGE [141]. PLAGE is selected as it is evaluated to be the best method in [142], however this does not preclude using any other methods surveyed in e.g. [142] from being applied to the differential analysis of Mass2Motifs.

7.6 Results & Discussions

7.6.1 Model Comparison

Figure 7.5 shows the perplexity for the two models on one of the Beer extracts (Beer3) as a function of K, the number of Mass2Motifs (for LDA) or clusters (for the mixture model). The mixture model is essentially equivalent to LDA with each spectrum being forced to consist of only one Mass2Motif. As such, if LDA is indeed finding structural features as conserved patterns of fragments and losses, it should explain the data with fewer Mass2Motifs than the mixture model. This is because the mixture model has to create separate Mass2Motifs for all observed combinations of structural features. The lower perplexity in Figure 7.5

demonstrates that LDA provides a better model fit on the held-out data compared to multinomial mixture model due to its lower perplexity. This validates our assumption that allowing multiple conserved blocks to be present in small molecule fragmentation data is a better representation of the biochemical properties of the fragmented molecules. The perplexity result on the held-out data in Figure 7.5 suggests a reasonable value for K to be in the range of 200 to 400, at the elbow of the curve where increasing the number of topics does not result in further decrease of perplexity.

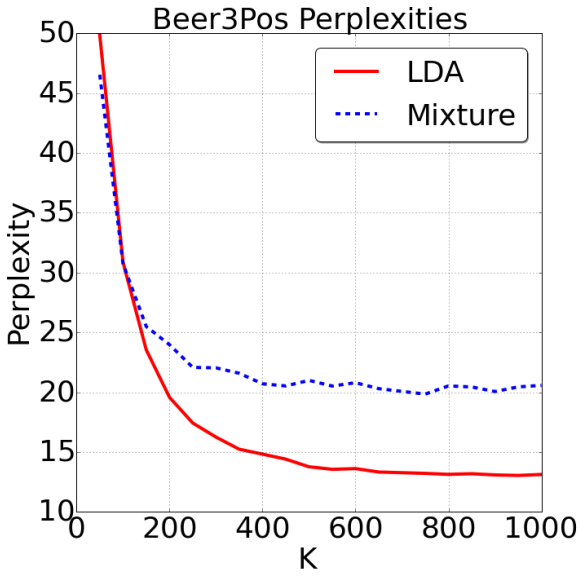


Figure 7.5: Results of model comparisons of LDA and multinomial mixture model on the Beer3 data. The lower perplexity values for $K > 100$ demonstrates that LDA provides a better model fit on the held-out data when compared to the mixture model.

7.6.2 Biochemical Analysis

With K the number of Mass2Motifs set to 300 and other hyperparameters set to be the same as in cross-validation, Mass2Motifs were extracted for each Beer data and characterised by JvdH for biochemical relevance. As discussed in Section 7.4, the distributions over the features that make up the Mass2motifs and the distributions over Mass2motifs for each fragmentation spectrum can be thresholded in MS2LDAVis for results interpretation. For analysis, the threshold values of 0.05 and 0.01 for t_θ and t_ϕ were set, but they can easily be varied. The selection of these threshold values was based on JvdH's expert knowledge to allow for the extraction of a chemically-plausible set of features that comprise a Mass2Motif.

In the subsequent analysis that follows, Mass2Motifs with degrees 10 (i.e. that were present in ten or more spectra after thresholding) were manually inspected and annotated at different levels of confidence through integrating multiple supporting evidence such as the matching to

File	Total MS1 peaks	Linked to at least one structurally annotated M2M	%
Beer1Pos	1282	951	74
Beer2Pos	1567	1160	74
Beer3Pos	1422	1055	74
Beer4Pos	1363	930	68

Table 7.2: Mass2Motif coverage of MS1 peaks by percentage of MS1 peaks that can be explained by at least one structurally annotated Mass2Motif for the files acquired in positive ionization mode.

a database of known reference standard compounds and spectral matching of the MS2 spectra containing the associated fragments and/or neutral losses to the reference spectra in MzCloud (www.mzcloud.org). Key fragment or loss features from the annotated Mass2Motifs in one sample were then searched against the list of Mass2Motifs in other samples and their correspondences established if those key fragment or loss features were present in both.

Across the four Beer data, an average of 70% of spectra (Table 7.2) include at least one annotated Mass2Motif, with Mass2Motifs related to the same substructure consistently found across multiple beers (e.g. hexose-related Mass2Motifs were present in all positive ionization mode files with degrees from 58 to more than 100), despite the fact that each sample was processed through the workflow independently. Between 30 to 40 Mass2Motifs in each of the Beer sample could be structurally annotated as corresponding to a diverse set of biochemical substructures, including amino acid related (i.e. histidine, leucine, tryptophan, and tyrosine), nucleotide related (i.e. adenine, cytosine, and xanthine), and other molecules such as cinnamic acid, ferulic acid, ribose and N-acetylputrescine. In general, the more Mass2Motifs present in a particular spectrum, the more specific our annotations can potentially become. An exhaustive identification effort to characterise all spectra (metabolites) present in the data was not attempted by JvdH, as it would be a major undertaking on its own, however it is noted that annotating just 30 to 40 of the discovered Mass2Motifs provide some structural biochemical insights into 70% of the spectra. This suggests that a large percentage of metabolites can be automatically classified according to function (based on presence of functional groups or as a part of biological pathways).

As an example of the biochemical insights that can be obtained by an expert from MS2LDAVis, Figure 7.6 shows three of the eleven spectra that include Mass2Motif 19, characterised as corresponding to ferulic acid substructure. Ferulic acid is a compound found in the hard outer layer of grain (the bran) of cereals (an ingredient of beer) and is expected to be shared by the metabolites in beer as a substructure. Across the three spectra, we see conserved fragment and loss features shared by the spectra explained by Mass2Motif 19, with the most conserved features highlighted in Figure 7.6D. Unlike MS2Analyzer [130] where the prior information on the fragment features of interest has to be specified in advance, the discovery of conserved features in MS2LDA is performed in an unsupervised manner. In addition,

JvdH verified that the *loss˙176.1086* feature in Figure 7.6B is an informative feature related to the complete ferulic acid substructure. While this is easily observed from the visualisation, information on conserved patterns of neutral loss will be difficult to extract from any other tools apart from MS2LDA. The entire results in Figure 7.6 shows that through MS2LDA, we can extract a biochemically relevant pattern present in just eleven of the entire set (>1000) of spectra, although the individual spectra can be quite different.

In a comparison to metabolite identification via spectral library matching using the NIST MS/MS database for small molecules (http://chemdata.nist.gov/mass-spc/msms-search/) and MassBank [60], only one from the eleven spectra explained by Mass2Motif 19 returns a ferulic acid related hit. This is despite the clear presence of fragment and loss features corresponding to ferulic acid substructure across the eleven spectra. Similarly, the beer metabolites explained by Mass2Motifs related to histidine, tyrosine, and tryptophan were subjected to spectral matching. In the verification by JvdH, matches to reference spectra were found for 33 spectra with 15 matches consistent with their characterised Mass2Motifs. These results demonstrate how MS2LDA effectively recognizes core substructures in mass spectral data and can serve as an aid to the classification and annotation of metabolites. Critically, it does this by matching only small portions of the spectra (substructures) rather than relying on complete spectral matches. In summary, for this subset of four Mass2Motifs, spectral matching allows classification of 45% of the associated metabolites whereas MS2LDA is able to functionally annotate all of them. In addition, MS2LDA can annotate and group spectra based on neutral losses (e.g. the loss of a free carboxylic acid group) which is not possible via spectral matching.

Validation to Reference Standard Molecules

Of the 45 molecules we were able to identify as standard molecules in one or more of the beer extracts, 38 can be explained by one or more annotated Mass2Motif, and 32 of the annotated Mass2Motifs correspond to known biochemical features that are consistent with the standard molecules. This demonstrates that characterised Mass2Motifs represent conserved patterns of metabolites' fragmentation spectra in authentic standard mixtures. Figure 7.7 shows some examples for these fragmentation spectra coloured by characterised Mass2Motifs. The spectra for phenylalanine (Figure 7.7A) and histidine (Figure 7.7B) share Mass2Motif 262, and indeed feature *loss_46.0054*, which has been verified by JvdH as informative that a carboxylic acid group (CHOOH) is lost from the molecular ion during fragmentation, is a common characteristic of phenylalanine and histidine. Similarly, other Mass2Motifs (115, 241) in Figures 7.7A and 7.7B are related to phenylalanine and histidine compounds. Finally, Figure 7.7D is the MS2 spectrum of adenosine, which consists of an adenine molecule conjugated to a ribose sugar molecule. The two associated Mass2Motifs 156 and 220 correctly

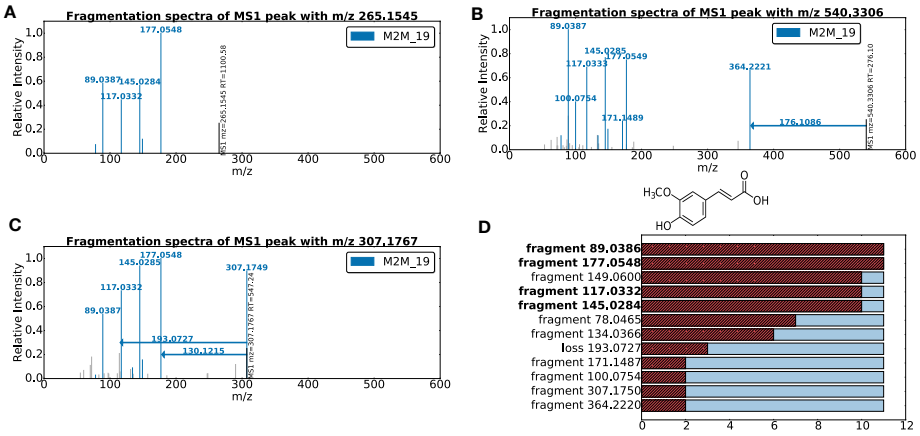


Figure 7.6: Three spectra, from the beer3 positive ionization mode file, each of which includes Mass2Motif 19, annotated as the plant derived ferulic acid substructure. A-C highlight mass fragments and neutral losses (arrows originating at the precursor ions) included in Mass2Motif 19 (fragments not explained by Mass2Motif 19 are light grey). Ferulic acid substructure is illustrated at the top of D, while the boxplot in D shows how common each fragment or loss features (representative of the substructure) are found in the 11 spectra explained by Mass2Motif 19 found in the dataset. Features highlighted in bold are consistently present in Mass2Motifs inferred across the four beer samples.

represent the two biochemically relevant substructures (i.e., adenine substructure and a loss corresponding to a ribose sugar).

Note that in our analysis on these standard molecules, the inferred Mass2Motifs were characterised first without any prior knowledge on the identities of standard compounds, but we still observe a high level of agreements between the identifications of standard compounds and the independent characterisation of Mass2Motifs which explains identified spectra. This suggests an alternative to the usual procedure where identification is performed first and the common substructures, shared by the small set of identified compounds, are deduced. In the complementary approach, characterised Mass2Motifs can be used as a starting point for analysis. The large number of spectra that can be explained by Mass2Motifs are further examined and their putative identities deduced through collaborating multiple evidences, such as substructure annotation, matching against standard database, MS2 spectral library, etc.

Comparison to Spectral Clustering

Spectral clustering approaches (e.g. Molecular Networking) can also help in molecular annotation by propagating identifications through the network. For example, if one spectrum can be identified, it can be used to putatively annotate the spectrums neighbours in the network. MS2LDA differs from this approach in three key ways. Firstly, MS2LDA does not require any complete spectra to be identified (they can be putatively annotated from Mass2Motifs).

Mass2Motif	Annotation	Degree	Fragment or Loss Features	Elemental Formula
115	[phenylalanine-CHOOH]-based substructure.	28	fragment_120.0808, fragment_103.0546, fragment_91.0541	C8H10N, C8H7, C7H7
156	[ribose (pentose, C5-sugar)-H2O]-related loss.	22	loss_132.0421	C5H8O4
202	[tryptophan-NH3]-related substructure.	15	fragment_118.0654, fragment_117.0571, fragment_91.0541, fragment_130.0645, fragment_188.0706	C8H8N, C8H7N, C7H7, C9H8N, C11H10NO2
211	N-acetylputrescine substructure.	24	loss_59.0370, fragment_114.0912, fragment_72.0447, fragment_60.0448	C2H5NO, C6H12NO, C3H6NO, C2H6NO
214	amine loss.	57	loss_17.0247	NH3
220	adenine substructure.	32	fragment_136.0629, fragment_119.0351	C5H6N5, C5H3N4
241	histidine substructure.	21	fragment_110.0718, fragment_156.0769, fragment_93.0450, fragment_95.0608	C5H8N3, C6H10N3O2, C5H5N2, C5H7N2
262	combined loss of H2O and CO – indicative for free carboxylic acid group (COOH).	90	loss_46.0053	CH2O2

Table 7.3: Annotations of the Mass2Motifs associated to the fragmentation spectra of the peaks generated by the standard molecules shown in Figure 7.7. The degree of a Mass2Motif indicates the number of MS2 fragmentation spectra in Beer3 positive ionization mode data having the fragment or loss features that can be explained by the Mass2Motif.

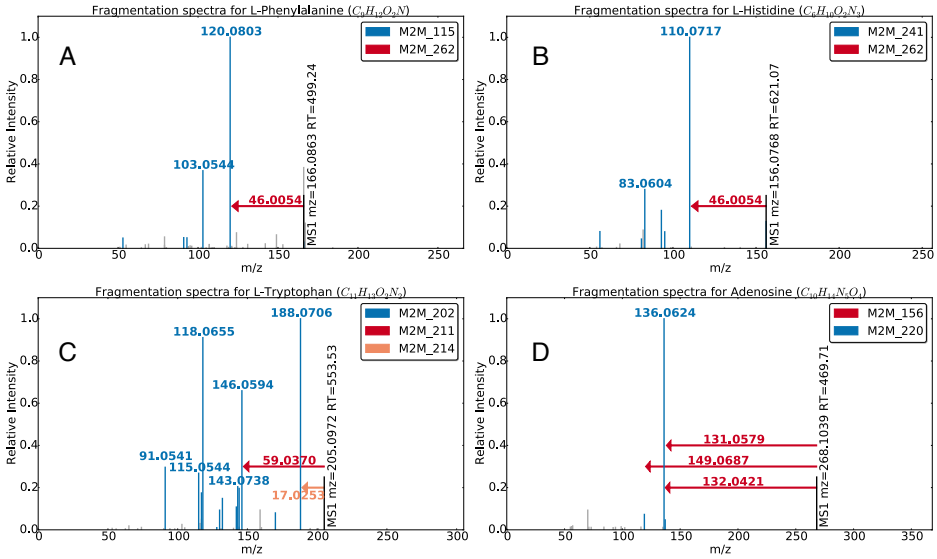


Figure 7.7: Mass2Motif spectra of identified standard molecules A) L-histidine, B) L-phenylalanine, C) L-tryptophan, and D) adenosine, with their characterized motifs (see Table 7.3) indicated by colours.

Secondly, MS2LDA does not require a high degree of total spectral similarity to allow spectra to share annotations; it just relied on the presence of a shared Mass2Motif. Finally, because spectra can include multiple Mass2Motifs, they can be given multiple annotations while in spectral clustering, each spectrum can only belong to one cluster. A key characteristic of MS2LDAis the ability to decompose MS2 spectra into multiple (potentially biochemically relevant) components. For example, in each of Figures 7.7A to 7.7D, we observe the spectra being decomposed into 2 or more Mass2Motifs. To our knowledge, no other methods can do this in an unsupervised manner without training spectra consisting of known structures or *a priori* knowledge of interesting combinations of fragment and/or loss features.

Similarly in MS2LDA, a fragmentation spectrum can now be described by one or more Mass2Motifs. Figure 7.8 demonstrates this with an example of a subset of the network produced by MS2LDAVis, consisting of spectra explained by two Mass2Motifs characterised as ferulic acid and ethylphenol. All but one spectrum can be explained by just one of the Mass2Motifs but one spectrum is generated by a molecule that contain both substructures and can therefore be explained by both Mass2Motifs. In the Molecular Networking analysis by JvdH, this spectrum is placed into the ethylphenol cluster, but its relationship with ferulic acid is lost. This results in a much less specific annotation of that spectrum. In contrast, the knowledge on the presence of both Mass2Motifs in the spectrum allows JvdH to assign it a putative compound identification of feruloyltyramine ([C18H20NO4]+) despite spectral matching producing no relevant hits. In general, the more Mass2Motifs present in a particular spectrum, the more specific our annotations can potentially become.

The same spectral clustering result is also reproduced in Figure 7.9 where a matrix of cosine

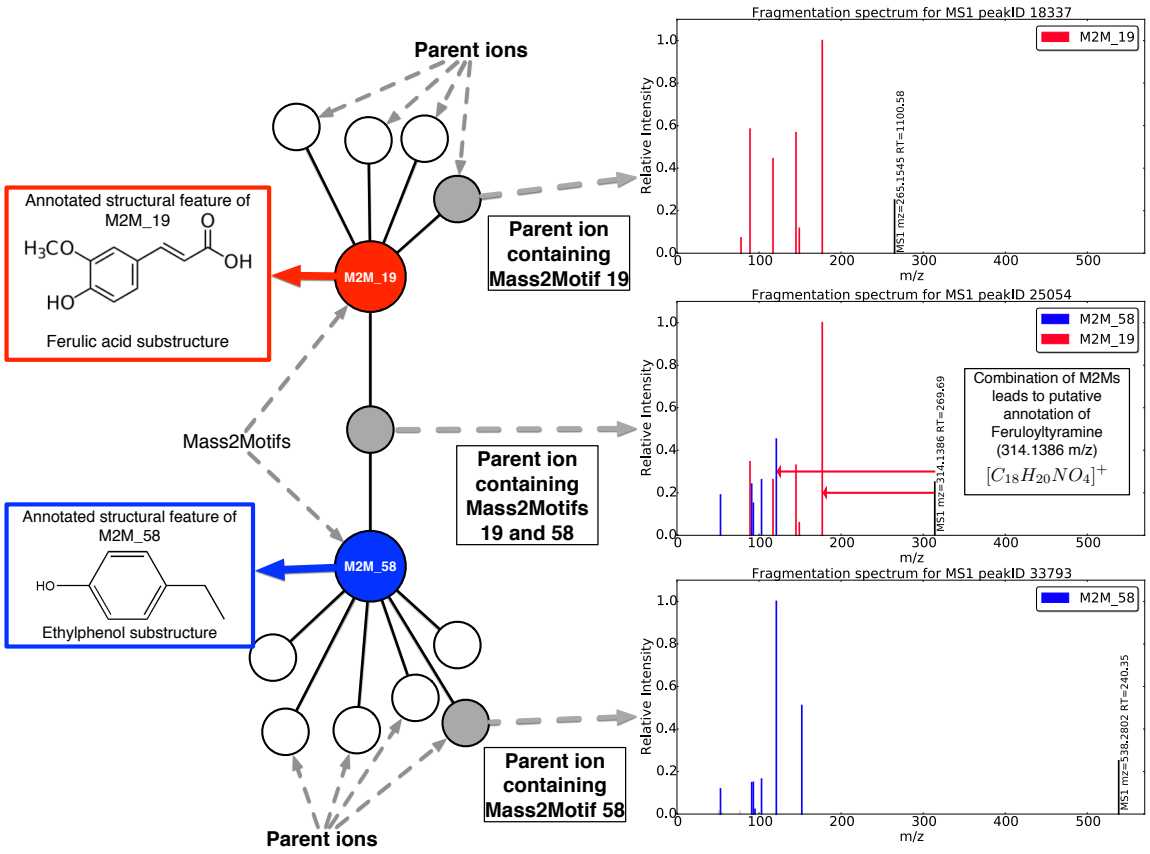


Figure 7.8: Mass2Motifs 19 and 58 were found to be representative of ferulic acid and ethylphenol, respectively. 11 fragmentation spectra can be explained by M2M_19, while 42 spectra can be explained by M2M_58. However, one spectra (shown as a gray node in the Figure) can be explained by both Mass2Motifs, but this is not possible in spectral clustering.

similarities of the spectra, placed in the ferulic acid based cluster and the ethylphenol based cluster (from Molecular Networking). Two distinct groups of spectra, based on their cosine similarities, can be seen — corresponding to each cluster. Members of each cluster can also be explained by a single Mass2Motif (the ferulic acid cluster by M2M_19, and the ethylphenol cluster by M2M 58). However, one spectrum (the last row in Figure 7.9) can also be jointly explained by the two Mass2Motifs. In cosine clustering, this spectrum would have to go into one cluster or the other based on its cosine similarity and valuable information is lost. Since a compound consists of multiple substructures, allowing each spectra to be explained by multiple Mass2Motifs naturally results in a greater potential of producing a more comprehensive characterisations of the substructures of a compound.

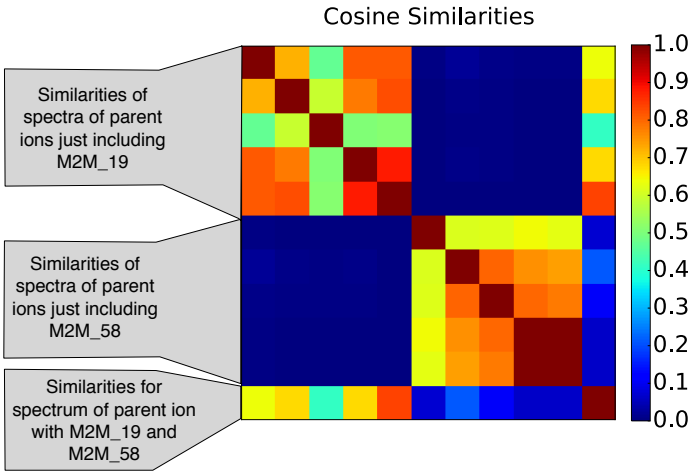


Figure 7.9: Cosine clustering results of spectra drawn from the ferulic acid based cluster and the ethylphenol based cluster (similar to M2M_19 and M2M_58). The last row represents a fragmentation spectrum that contains both substructures, but in the clustering approach, the spectra will be placed into one of the clusters based on its cosine similarity. In LDA, this spectrum can be explained by Mass2Motifs that characterise both substructures.

Differential Analysis of Mass2Motifs

We have shown that MS2LDA analysis can group molecules according to a shared Mass2Motif. As spectra can include multiple Mass2Motifs, so molecules can belong to multiple functional groups. In transcriptomic studies, it is common to consider the shared differential expression (DE) of a group of transcripts that are related through the sharing of the same Gene Ontology classification. The equivalent case in metabolomics are metabolites that share the same functional substructures and can potentially be mapped onto related pathways. The presence of the same functional substructure across these metabolites naturally suggest that their spectra can be described by the same Mass2Motif. If all metabolites sharing the same substructure are differentially expressed across samples, hypothesis can be generated as to

the underlying biochemical significance causing the expression changes. From performing differential analyses on the expressions (intensity values) of metabolites having spectra explained by the same Mass2Motif, it is therefore possible to assess the biochemical changes of groups of metabolites across samples. Note that this does not depend on the small number of metabolites having spectra that can be identified through spectral matching, instead it relies on the much larger sets of MS1 peaks having spectra that can be jointly explained by a Mass2Motif.

Using PLAGE [141], we assessed the DE of each Mass2Motif based on the intensity changes of the relevant MS1 peaks between beers 2 and 3. Figure 7.10 shows MS1 intensities of metabolites explained by two Mass2Motifs (characterised as guanine and pentose loss) with high PLAGE scores. In each case, the change in intensity across the two beer extracts are very clear (note that PLAGE considers changes in both directions when scoring). Within the molecules having spectra explained by the guanine Mass2Motif, we could annotate 5-guanine containing metabolites and identify 2 to through matching to reference standards (Figure 7.10A). For the pentose Mass2Motif, we could annotate 8 and identify 5 pentose-containing metabolites from the Mass2Motif (Figure 7.10B). These biochemically relevant metabolites show interesting patterns in the DE between the two beers. As an example of how MS2LDA differential analysis can support hypothesis generation for an expert, JvdH noted that in Beer3, the free guanine is present more often, whereas in Beer2, the conjugates of guanine are more abundant (Figure 7.10A). This reflects the differences in the chemical components of the two beers. Similarly, as metabolites can include multiple Mass2Motifs, JvdH observed that the four spectra (in Figures 7.10) annotated as guanine-related metabolites (i.e., guanosine, two methyl-guanosine isomers, and a pentosyl-hexosylguanine) are also connected to the pentose loss Mass2Motif, which itself was also differentially expressed between the two beers. Indeed, the structures of those metabolites all share both a guanine and a pentose substructure. A comparison made by JvdH to Molecular Networking results revealed that in the standard spectral similarity approach, these spectra were distributed over 10 spectral clusters. In other words, the interesting structural and intensity similarity between these molecules exposed by MS2LDA would not be found via spectral clustering.

7.7 Substructure Discoveries Across Many Fragmentation Files

Metabolomics dataset consist of fragmentation spectra in multiple input files, where each file is generated from measurements of a technical or biological replicate. Manual inspection of the results revealed that many Mass2Motifs, related to the same substructures, are consistently present in two or more beers. This is despite each file being processed inde-

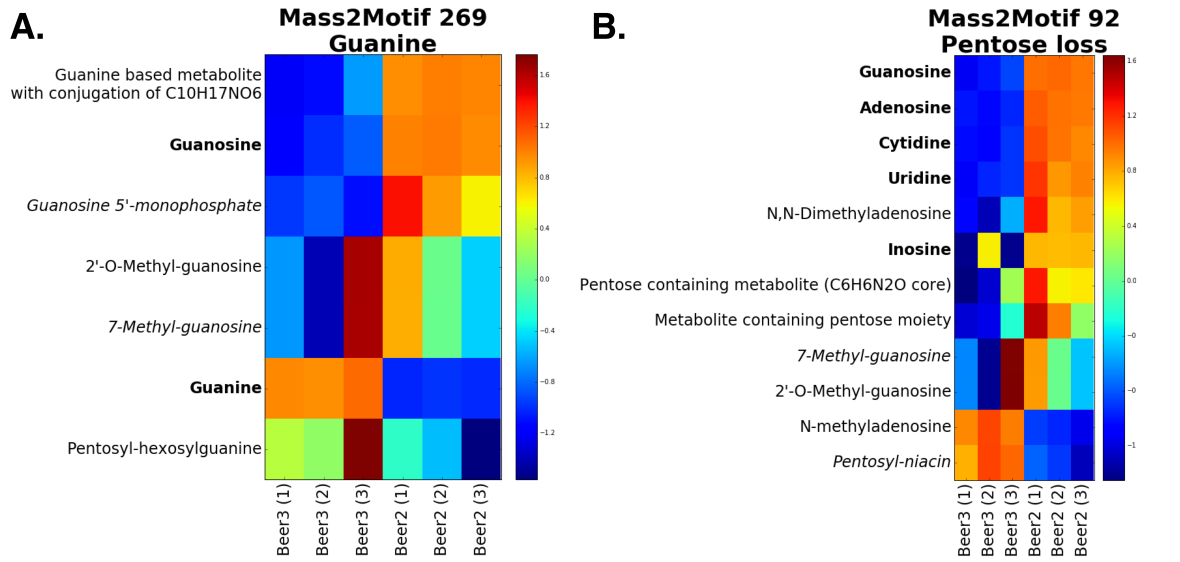


Figure 7.10: Log fold change heat-maps for the A) guanine and B) pentose loss Mass2Motifs. Each row is an annotated parent MS1 peak and columns represent different beer extracts. Bold names for parent MS1 peaks could confidently be matched to reference compounds, while italic names are for those that are annotated at a lower degree of confidence.

pendently through MS2LDA. For example, the hexose-related Mass2Motifs are present in all positive ionization mode beer files with degrees from 58 - 100 in each beer. Other larger Mass2Motifs, such as histidine, ferulic acid, etc., are also found across the beer files. The results suggest that we can jointly model the presence or absence of Mass2Motifs across many input files at once.

While our initial motivation for modelling Mass2Motifs across files is to eliminate the tedious matching of Mass2Motifs from one file to another, modelling the presence of Mass2Motifs across multiple files also opens interesting avenues for research. For instance, from such a model, spectra from different files but explained by the same Mass2Motifs can be rapidly flagged for validations to obtain their substructure characterisation. The posterior probability of observing a Mass2Motif in each file can also be compared, providing an estimate as to the presence or absence of certain substructures in each file. This information might be useful for further analysis. In [129], substructure analysis (through Molecular Networking) is applied in a clinical setting to discover patterns of fragment peaks, which correspond to substructures shared by drug metabolites. The results from [129] revealed that substructure information help in the identification of many drug metabolites. As discussed before, Molecular Networking groups related spectra by their cosine similarities, and the manual validation to extract key fragment peaks that explain *why* certain spectra are grouped together is time-consuming. This is a limitation not shared by our proposed model (that jointly models Mass2Motifs across mutiple files) if we were to apply the model to this kind of dataset.

7.7.1 Multi-file LDA Model

Here we introduce an extension of the standard LDA model that allows for Mass2Motifs, the distributions over fragment and loss features, to be shared across files. Within each file, fragmentation spectra have their own file-specific probabilities of observing certain Mass2Motifs. When only a single input file is provided, the proposed extended model reduces to the standard LDA model. The conditional dependences of this model, which we call the multi-file LDA model, is shown in Figure 7.11 and described below.

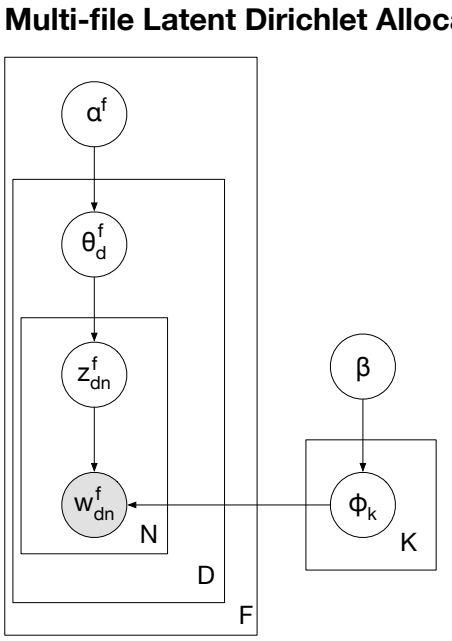


Figure 7.11: Graphical model of the multi-file LDA model. The addition to the standard LDA model is the plate on F that denotes an index over the files, $f = 1, ..., F$. Circles denotes random variables, while the shaded node denotes the observed word value.

We follow the notations used for the standard LDA model in Section 7.4, but expand them slightly to index the different files. w^f_{dn} refers to the n-th fragment or loss feature in the d-th fragmentation spectra in file f. Each w^f_{dn} is assigned to the k-th Mass2Motif, a multinomial distribution over the entire vocabulary of fragment and loss features, through the indicator variable z^f_{dn}, so $z^f_{dn} = k$ if feature n from fragmentation spectra d in file f is assigned to the k-th Mass2Motif. The probability of seeing certain Mass2Motifs for each d-th fragmentation spectra in file f is then drawn from a multinomial distribution with a parameter vector $\boldsymbol{\theta}^f_d$. This parameter vector $\boldsymbol{\theta}^f_d$ is in turn drawn from a prior Dirichlet distribution having the parameter vector $\boldsymbol{\alpha}^f$. Note that unlike the standard LDA model, each file now has its own prior Dirichlet distribution parameterised by $\boldsymbol{\alpha}^f$ and all documents in the same file has their

document-to-topic distributions drawn from the same prior Dirichlet specific to the file.

$$z^f_{dn}|\boldsymbol{\theta}^f_d \sim Multinomial(\boldsymbol{\theta}^f_d) \tag{7.5}$$

$$\boldsymbol{\theta}^f_d|\boldsymbol{\alpha}^f \sim Dir(\boldsymbol{\alpha}^f) \tag{7.6}$$

As in the case of standard LDA, the k-th multinomial distribution for a Mass2Motif is still characterised by the parameter vector $\boldsymbol{\phi}_{\boldsymbol{z}^f_{dn}}$, with $\boldsymbol{\phi}_{\boldsymbol{z}^f_{dn}}$ drawn from a prior Dirichlet distribution that is global to all files, parameterised by the vector $\boldsymbol{\beta}$.

$$w^f_{dn}|\boldsymbol{\phi}_{\boldsymbol{z}^f_{dn}} \sim Multinomial(\boldsymbol{\phi}_{\boldsymbol{z}^f_{dn}}) \tag{7.7}$$

$$\boldsymbol{\phi}_k|\boldsymbol{\beta} \sim Dir(\boldsymbol{\beta}) \tag{7.8}$$

Inference in the multi-file LDA model is again performed via a collapsed Gibbs sampling scheme. The conditional probability of $P(z^f_{dn} = k|w^f_{dn}, ...)$ of the assignment of feature n in spectra d file f to Mass2Motif k is given by eq. (7.9).

$$P(z^f_{dn} = k|w^f_{dn}, ...) \propto P(w^f_{dn}|z^f_{dn} = k, ...)P(z^f_{dn} = k|...) \tag{7.9}$$

where ... denotes any other parameters being conditioned upon but not explicitly listed. Similar to the derivation of standard LDA, we can marginalise over all $\boldsymbol{\phi}_k$ parameters in the likelihood term, $P(\boldsymbol{w}^f_{dn}|z^f_{dn} = k, ...)$ of eq. (7.9), to obtain:

$$P(w^f_{dn}|z^f_{dn} = k, ...) \propto \frac{\sum_f c^f_{kn} + \beta_n}{\sum_n \sum_f c^f_{kn} + \beta_n} \tag{7.10}$$

where $\sum_f c^f_{kn}$ is the total number of the n-th feature from all files currently assigned to Mass2Motif k (this count excludes the current feature being sampled in the current iteration of Gibbs sampler). For the prior term $P(z^f_{dn} = k|...)$, marginalising over all $\boldsymbol{\theta}^f_d$ parameters produces as in the standard LDA:

$$P(z^f_{dn} = k|...) \propto c^f_{dk} + \alpha^f_k \tag{7.11}$$

with c^f_{dk} the number of features from document n in file f currently assigned to Mass2Motif k, excluding the current feature being sampled. Putting the prior and likelihood terms together, the following predictive distribution is obtained for the assignment of feature n from document d file f to Mass2Motif k:

$$P(z^f_{dn} = k|w^f_{dn}, ...) \propto (c^f_{dk} + \alpha^f_k) \cdot \frac{\sum_f c^f_{kn} + \beta_n}{\sum_n \sum_f c^f_{kn} + \beta_n} \tag{7.12}$$

In each iteration of the Gibbs sampling, the information on the current feature n in spectra d file f being sampled is removed. Reassignment of the feature to a Mass2Motif is then performed by sampling z_{dn}^{f} from the distribution specified by eq. (7.12). Given $\boldsymbol{z}$, the predictive distribution for the d-th spectrum over the Mass2Motifs, $\boldsymbol{\theta}_{d}^{f}$, is obtained from the expectation of the Dirichlet-Multinomial distribution defined in eqs. (7.5)-(7.6):

$$\theta_{dk}^{f} = \frac{c_{dk}^{f} + \alpha_{k}^{f}}{\sum_{k} c_{dk}^{f} + \alpha_{k}^{f}} \tag{7.13}$$

where c_{dk}^{f} is the count of features from spectra d in file f assigned to Mass2Motif k.

For each spectra, the multinomial count vector $\mathbf{c}_{d}^{f}$, of features from the spectra that are assigned to the different Mass2Motifs, is a sample from the Dirichlet-Multinomial distribution defined in eqs. (7.5)-(7.6). Given all the $\mathbf{c}_{1}^{f}, \mathbf{c}_{2}^{f}, ...\mathbf{c}_{D}^{f}$ vectors in the file, the parameter $\boldsymbol{\alpha}^{f}$ of the Dirichlet-Multinomial distribution of spectra-to-Mass2Motifs in file f can be estimated by maximizing the log likelihood, $log \prod_{d=1}^{D} p(\boldsymbol{c}_{d}^{f}|\boldsymbol{\alpha}^{f})$. An iterative procedure to approximate this is described in [143].

In a similar manner to standard LDA, each k-th Mass2Motif, the predictive distribution over features, $\boldsymbol{\phi}_{k}$, can be obtained as the expectation of the Dirichlet-Multinomial distribution defined in eqs. (7.7)-(7.8):

$$\phi_{kn} = \frac{c_{kn} + \beta_{n}}{\sum_{n} c_{kn} + \beta_{n}} \tag{7.14}$$

where c_{kn} is the count of the n-feature from all files that are assigned to Mass2Motif k.

7.7.2 Results & Discussion

On the dataset of four Beer extracts in positive ionisation mode processed through multi-file LDA using the same hyperparameters as the individual LDA. For data interpretation, initially, the same threshold values on t_{θ} and t_{ϕ} were selected as the previous single-file analysis (0.05 and 0.01 respectively). Table 7.4 shows the results of five global Mass2Motifs that could be matched to the individual LDA results in Section 7.6.2. The results in Table 7.4 shows that multi-file LDA produces comparable results on the Mass2Motifs composition. This is entirely expected given that the four Beer extracts used for evaluation share similar metabolic profiles and correspondingly, have many substructures in common.

Information from all files now contribute to the inference of global Mass2Motifs. The fact that global Mass2Motifs that are consistent with our previous characterisation in Section 7.6.2 still emerge suggests the same underlying patterns of fragment and loss features to be present in each Beer extract. Figure 7.12 shows four example fragmentation spec-

Mass2Motif	Annotation	Top Features Above Threshold
M2M_17	Ferulic acid substructure	**fragment_177.05478**, **fragment_89.03865**, **fragment_145.02844**, **fragment_117.03319**, loss_58.98941, fragment_163.03887, **fragment_149.05998**, loss_88.09967
M2M_155	Histidine substructure	**fragment_110.07161**, **fragment_156.07687**, fragment_83.06041, **fragment_93.04511**, fragment_82.05246, fragment_209.10558, **fragment_95.06057**, loss_167.08663, fragment_81.04494, loss_191.0615
M2M_115	Leucine substructure	**fragment_86.09653**, **fragment_132.10165**, fragment_69.07013, fragment_332.112, fragment_143.11763
M2M_95	Water loss substructure	**loss_18.01031**, fragment_314.0859, fragment_296.07259
M2M_232	Asparagine substructure	**fragment_136.06231**, loss_162.03459, **fragment_119.0354**, loss_162.00534, fragment_137.04623

Table 7.4: Five global Mass2Motifs inferred from multi-file LDA. For each Mass2Motif, the top features above threshold are listed. Features characterised as key to the substructure from the previous individual LDA analyses are shown in bold.

tra originating from different Beer extracts — jointly inferred by multi-file LDA as containing the Mass2Motif characterised as the ferulic acid substructure. While this can be achieved from independently running LDA on each file, the tedious matching process of common Mass2Motifs across files can now be eliminated. Inspections on the degree (the number of spectra associated to a Mass2Motif above the user-defined threshold t_θ) of the five Mass2Motifs in Table 7.4 revealed that with a minor adjustment to t_θ, the same sets of fragmentation spectra previously associated to the listed Mass2Motifs can all be recovered.

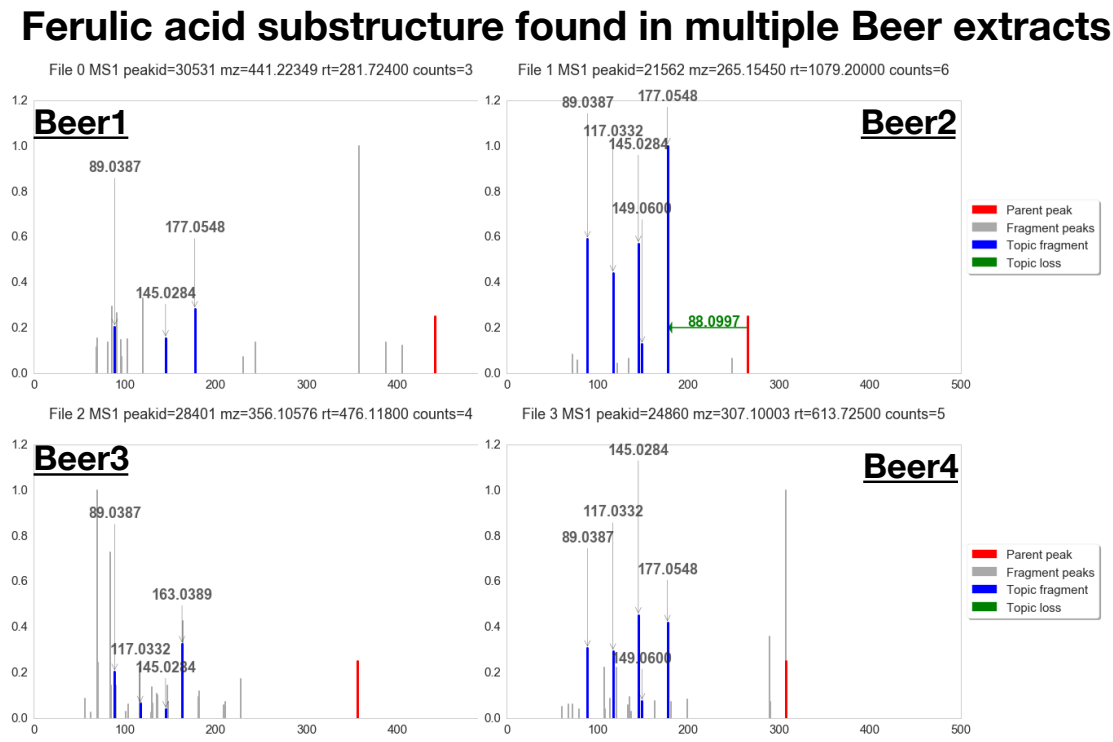


Figure 7.12: Fragmentation spectra from different Beer extracts found by multi-file LDA to contain the same Mass2Motif 17 characterised as the ferulic acid substructure.

From each posterior sample, we can also obtain the updated $\boldsymbol{\alpha}^f$ for the different Mass2Motif across all files. As $\boldsymbol{\alpha}^f$ is the asymmetric parameter that serves as the pseudo-count in the Dirichlet-Multinomial distribution of spectra-to-Mass2Motifs, a high value of $\boldsymbol{\alpha}^f_k$ for a particular k means that a specific Mass2Motif is more likely for each spectra in file f. Figure 7.13 shows the plot of posterior alpha values for the Mass2Motifs characterised as the ferulic acid, histidine and leucine substructures. Inspections of the comparisons in Figure 7.13 may lead to interesting biological hypothesis that explains e.g. why the ferulic acid substructure is more likely for the spectra in the third beer file compared to the others.

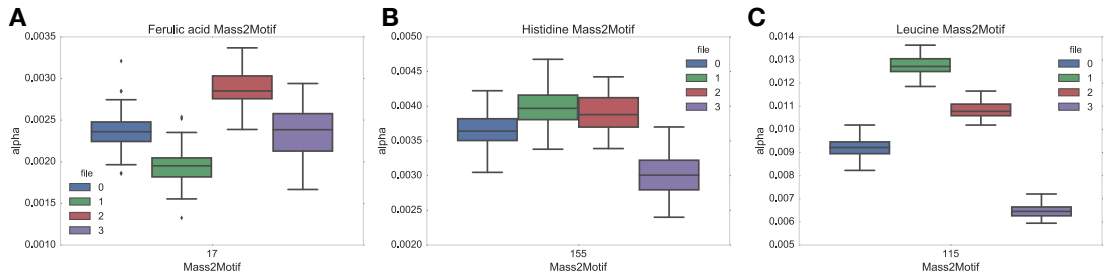


Figure 7.13: Posterior alpha values for the **A)** ferulic acid, **B)** histidine and **C)** leucine Mass2Motifs across the different beer files.

7.8 Conclusion

We have introduced MS2LDA, a pipeline that simplifies fragmentation data by exploiting the parallels between MS fragmentation data and text documents. The pipeline performs all steps required in the analysis: the preparation of a co-occurrence matrix of fragment and loss features in fragmentation spectra, the LDA analysis, and the graphical visualization of the resulting output. Evaluation of the workflow on beer extracts result in numerous informative patterns of concurrent mass fragmental and neutral loss, termed Mass2Motifs, which we could annotate as biochemically-relevant substructures. The MS2LDA approach is markedly different from other advanced spectral analysis tools as multiple Mass2Motifs can be associated with one metabolite, and determination of the key mass fragments or neutral losses that are part of a conserved structural motif is unsupervised. The application of LDA to modelling the fragmentation spectra produced by mass spectrometry instrument is exhaustively explored in this chapter. We have shown how spectra comprise of multiple substructures which can be explained by characterised Mass2Motifs. Through comparison to Molecular Networking, we demonstrated through examples how MS2LDA allows us to explain parts of a spectrum, producing a better functional annotation in contrast to spectral clustering where a spectrum can only be placed in one cluster. The differential analysis of parent ions having fragments sharing Mass2Motifs introduces the possibility of assessing changes in the expression levels of metabolites — sharing substructures explained by a characterised Mass2Motif — despite the identities of the metabolites unknown. This is particularly useful in the case of untargeted metabolomics experiments.

As future work, we envision developing a larger library of characterised Mass2Motifs from data sets produced on a diverse range of analytical platforms and different sample types. A challenge to this approach lies in the fact that mass spectrometry instruments have varying accuracy and therefore require different binning thresholds. One possible solution is define a common space of chemical vocabulary; rather than using binned fragment and loss features; a Mass2Motif can now be defined as the distribution over chemical formulae words. Such an approach is hampered by the fact that *de novo* elemental formula assignment itself is a

difficult problem, with large uncertainties as to the correctness of annotated formulae of a fragment or loss feature. A probabilistic model of formula annotation that can offers confidence values on the formulae annotation of a fragment or loss feature might be useful in this scenario as formula annotation uncertainties can then be incorporated into Mass2Motif formation in MS2LDA. Non-parametric model such as the Hierarchical Dirichlet Process [96] can also be applied for topic discovery by letting the number of Mass2Motifs to be learned from the data itself. This allows for a truly flexible system of substructure annotation where Mass2Motifs can be obtained from training the model on large public fragmentation databases, such as HDMB or MassBank. In a similar manner as our analysis in this chapter, the resulting Mass2Motifs can be characterised. New and unseen fragmentation spectra can be run using the pre-trained models with these characterised Mass2Motifs, allowing for the rapid identification of the substructure that comprise a fragmentation spectra.

An extension of the standard LDA model, in form of the multi-file LDA model, is also proposed in this chapter to handle Mass2Motif inference from multiple data sets. Such a model can be used in large-scale clinical and metabolomic studies. In this model, the prior information on which prior Mass2Motifs the user expects to see can be included into the MS2LDA workflow, allowing the LDA inference on certain known Mass2Motifs that are expected to be present in the sample while allowing others to be inferred from the data.

Other LDA-based techniques developed for text (e.g. hierarchical LDA [144]) are also likely to offer benefits as we hypothesise that Mass2Motifs can be defined in a hierarchy. For instance, generic patterns such as the loss of CO2 may lie at the top of the hierarchy of Mass2Motifs, while the more specific Mass2Motifs are formed at the bottom. It is anticipated that visualisation and the meaningful presentation of inference results will be a challenging task in such a model.

In general, we anticipate that the approach of applying topic modelling techniques to fragmentation spectra data to be particularly useful in research areas such as clinical metabolomics, pharmacometabolomics, environmental analysis, natural products research and nutritional metabolomics, as it can quickly and in an unsupervised manner recognize substructure patterns related to drugs, pollutants, and food-derived molecules, respectively.

Chapter 8

Conclusion

LC-MS based omics, such as proteomics and in particular metabolomics, play a major role in modern systems biology. However, there are many challenges in data pre-processing steps before LC-MS data can be analysed. In particular, the information from the peak grouping step is often not used in the alignment and identification stage. More broadly, the presence of a grouping structure means that a set of peaks can be structurally related. Generative models can be used to induce the clustering on the peak data, revealing the latent structures that exist.

In this thesis, we have shown that using this structural information can help in alignment and identification. Our tools are generative modelling. Using this, we showed that grouping can be used to improve alignment (matching). In particular, we improve a direct matching method by by incorporating grouping information. IP clusters, corresponding to groups of peaks that are related through being the ionisation product peaks of the same metabolite, can also be matched directly – either via a direct matching scheme or through a second-stage clustering method. A generative model also can be constructed that models all peaks across all files at once, producing alignment as a result and also useful latent structures. From fragmentation data, identification can be enhanced by taking into consideration the grouping of fragmentation peaks that potentially correspond to substructures.

8.1 Summary of Contributions

This thesis makes a number of contributions, motivated by our thesis statement in Section 1.1, which is restarted in the following:

Untargeted liquid chromatography mass spectrometry data pre-processing is a challenging task that is often subjected to errors and inaccuracies. Much of this can be attributed to the complexity of the LC-MS data itself and also to the lack of knowledge as to which compounds are present in the sample. However, the structural dependencies in the observed

peak data means that through generative modelling, we can explain the relationships between peaks, allowing us to produce groups of related peaks that can be used to improve or enhance the alignment and identification steps of LC-MS data pre-processing.

The thesis statement is then supported by the following contributions:

1. Chapter 4 presented a method to perform the grouping of related peaks by RT and combine this grouping information with a direct-matching method. We demonstrated on benchmark datasets with alignment ground truth how this information can be used to improve alignment.

2. Chapter 5 expands upon the grouping process in Chapter 4, where only the RT information is used, and proposes a model that takes into consideration the mass information as well when grouping related peaks. Through a set of transformation rules (specific to metabolomics data at the moment), our model produces IP clusters, where member peaks can be explained by their ionisation product transformations. We showed in Chapter 5 that IP clusters can be matched directly in place of peak features, and this produces an improved alignment performance. Additionally, uncertainties in the matching can also be quantified through a second-stage clustering of the IP clusters.

3. Chapter 6 expands upon the work in Chapter 5. Instead of having to fix the MAP cluster assignment of peaks to local clusters in the same file, we introduce the notion of a hierarchical model that allows for peaks across multiple files to be grouped. We show that modelling the data generatively in this manner and performing grouping allows us to produce alignment (matching). From the model, highly confident matched peaksets can be extracted, which may be useful in some analytical cases.

4. Chapter 7 looks at fragmentation data, produced from tandem mass spectrometry process. We show that by thinking generatively, we can explain fragment peaks by how they relate to substructures shared by metabolites. This aids in exploratory data interpretation during the identification of compounds in metabolomics data.

8.2 Future Work

There are a number of interesting future work that could follow from the results in this thesis. They are:

8.2.1 Improved Generative Models to Cluster Related Peaks

Generative modelling of peaks are demonstrated in Chapters 4 and 5 where we build a model to cluster peaks in the same file by their RT values and explainable mass transformations. However, there is more information present in the LC-MS data that is not used in our model. In particular, peaks elute from liquid chromatography and produce chromatographic profiles (the retention time value of a peak is a point along the chromatographic profile). The chromatographic profiles of related peaks should be similar, and in [49], a mixture model is proposed to cluster using chromatographic profiles. This is shown to produce improvements over the greedy approach of clustering peaks. We might also want to incorporate this information into our models, for example by changing the PrecursorCluster model from Chapter 5 and adding another likelihood term for the correlation of the chromatographic profiles. Following [49], we might we use a two-component mixtures to describe this likelihood: the first component corresponds to the likelihood of peaks to be in the same cluster, while another component describes the likelihood of peaks to be in different clusters based on their chromatographic correlations. The proposed implementation in [49] uses Gibbs sampling, and we foresee that modifying our inference procedure to accommodate this new likelihood term to be straightforward.

The proposed PrecursorCluster model in Chapter 5 also makes a fairly strong assumption that the most intense peak in the cluster must be the $M + H$ peak. This assumption may not always hold as we have seen cases where valid clusters do not have its most intense peak as the $M + H$ peak. Relaxing this assumption means more clusters may be obtained, but depending on the data, we might also see more false assignments of peaks to clusters. Performing validations on the results with and without this constraint will be challenging and require a close collaboration with a life scientist who possesses the necessary expert knowledge to validate the data. This however might point to a more flexible method where peaks can be clustered without having to make such a strong assumption.

8.2.2 Using the Generative Models for Identification

The proposed models in this thesis are generally validated against the alignment ground truth, i.e. we consider that the models produce a sensible clustering of related peaks if we can take the resulting groups and use them to obtain a good alignment performance. However, that is not the only use of the output from the models. In particular, the set of related peaks that have been grouped together and can be explained as being generated from the same latent variable (corresponding to a compound) might be used for identification. We have explored a preliminary form of this idea in Section 5 where we hand-pick clusters that correspond to Cysteic acid and melatonin, and also in Section 6 where we take some global RT clusters

and annotate them by their putative compound identities. Note that once we have assigned a putative compound identity to a clustering object, being able to annotate the entire peaks that are members of that cluster is a natural consequence of the clustering output of the model. It is worth investigating whether such an approach might bring an improved discriminative power to identification compared to identifying peaks one-by-one, as what is conventionally done at the moment. However, the lack of gold standard for identification means that this will be an extensive endeavour that again requires a close collaboration with a life scientist.

8.2.3 Data Visualisation and Interpretation

As the MS2LDAVis module in Chapter 7 shows, the interpretation of complex inference results can be daunting to the average user. Having an easy-to-use visualisation interface that displays the most pertinent information in a user-friendly manner shifts this burden of interpretation from the user to the system, and it is important when producing tools that we hope will be used and adopted by the community at large. One of the problems with the probabilistic matching results returned by the Cluster-Cluster method in Chapter 5 and also the HDP-Align method in Chapter 6 is that the result does not lend itself to easy interpretation. The conventional way of presenting a list of aligned peaksets is in the form of a table, where each row corresponds to a consensus peak (derived from the aligned peakset) and the columns are the observed intensities in the different LC-MS runs. From our output, we now obtain aligned peaksets at varying probabilities, but how about other information that we obtain from inference? From the inferred clustering structures, we obtain more than just alignment as we can also extract for e.g. the inferred ionisation product types from PrecursorCluster, the entire top-level global RT cluster from the HDP model, etc.. Displaying this information in a manner that is useful to the user requires careful considerations. For instance, we might decide to supplement the usual tabular view of peaklist with a graphical visualisation showing how peaks are explained through which ionisation product transformations and their probabilities. Integration with external database services, such as PubChem [122], is also useful in this kind of visualisation systems to obtain additional meta-data that may enhance interpretation.

8.2.4 Topic Modelling of Fragmentation Data

In our study, the multi-file LDA model proposed in Chapter 7 is applied to a metabolomics dataset containing four beer LC-MS runs. However a larger dataset (up to 30 LC-MS runs) from a clinical experiment involving drug studies are available from our collaborators. This dataset has been run through the multi-file MS2LDA pipeline and can be used to validate that indeed we can find useful Mass2Motifs that correspond to substructures shared by drug

metabolites. The proposed inference procedure in Chapter 7 relies on Gibbs sampling, which has difficulties scaling to a large number of files, so we performed variational inference [98] instead.

Our LDA models (both the single- and multi-file version) assumes that the number of Mass2-Motifs K is known and has to be defined by the user or estimated through a cross-validation procedure (as what we have done in Chapter 7). Setting K that is too large may lead to overfitting with many small, overly specific Mass2Motifs, while setting a value for K that is too small leads to underfitting with large and generic Mass2Motifs. Hierarchical Dirichlet Process has been used as the prior in a non-parametric topic model [96] and provides a principled mechanism to let the number of Mass2Motifs to be learned from the data. This can be implemented next. Along this line, we have also seen that Mass2Motifs form *hierarchies*, with generic substructures, such as the loss of CO that is shared by multiple Mass2Motifs. This suggest that a hierarchical extension of the LDA model can be considered to model the data [144].

The problem of transferring Mass2Motifs that we have learned from one dataset to another is also something we need to consider as this will allow inferred and characterised Mass2Motifs to be stored in a database and applied to new, unseen data, allowing for rapid explorations of the unknowns . One way we can do this is by fixing the topic-to-word probabilities for the selected Mass2Motifs and using them when running LDA on the new data. This approach, however, is rather *ad-hoc*, and more principled approach such as [145] can be considered. For the transferring of Mass2Motifs to work, a common vocabular space over the words have to be defined on the existing data used for training and the new data. Rather than using the discretised fragment and loss features (as what we do now) that heavily depend on the mass accuracy of a particular instrument, we may explore alternative binning procedures that use the elemental formulae as the 'words' in the LDA system . The MS2LDAVis module can also be extended to allow for Mass2Motifs expressions in the different files to be compared easily. All these are the necessary building blocks that contribute towards the development of an online interactive system that the community can use to submit validated topics and apply them new dataset for a rapid exploration of the 'fragmentome' on new and unseen fragmentation data.

8.3 Summary and Conclusions

Data pre-processing is a challenging task in LC-MS preprocessing pipeline. In this thesis, we have shown how generative models can be used to explain the relationships between related peaks, allowing for groups of related peaks to be extracted. We have shown how starting from this premise, we could propose new methods that improve on alignment and

enhance identification. In alignment, this is accomplished through the grouping of related peaks into ionisation product (IP) clusters, whether based on retention time (RT) alone or by considering the mass relationships between peaks as well. More accurate alignment can be constructed by taking into account this information on the IP clusters — as opposed to the normal case of matching by peak features alone and not taking the structural dependencies of these peaks into account. Modelling LC-MS runs hierarchically also allows us to group IP peaks within and across runs at once, and while in this thesis, we use the results to produce a probabilistic matching of peaks, the inferred latent structures might be used for other steps in the data pre-processing pipeline as well. And finally, we look at fragmentation data and demonstrate that by modelling the structural dependencies of fragment peaks, we produce a method that aid in data interpretation and the characterisation of complex fragmentation spectra. The latter represents an active research that directly addresses the primary bottleneck of data pre-processing of metabolomics data in an untargeted manner.

Although there is a lot of work to be done still, we believe that the thesis presents a compelling case to the benefit of generative modelling of peak data. The structural information that is present in mass spectrometry data is often neglected in alignment and identification via fragmentation data. Our results show that this results in useful information that can be used to improve the quality of the preprocessing pipeline.

Bibliography

[1] T.-H. Tsai, M. G. Tadesse, C. Di Poto, L. K. Pannell, Y. Mechref, Y. Wang, and H. W. Ressom, "Multi-profile bayesian alignment model for LC-MS data analysis with integration of internal standards," *Bioinformatics*, vol. 29, no. 21, pp. 2774–2780, 2013.

[2] T. S. Lee, Y. S. Ho, H. C. Yeo, J. P. Y. Lin, and D.-Y. Lee, "Precursor mass prediction by clustering ionization products in LC-MS-based metabolomics," *Metabolomics*, vol. 9, no. 6, pp. 1301–1310, 2013.

[3] M. Mann and O. N. Jensen, "Proteomic analysis of post-translational modifications," *Nature Biotechnology*, vol. 21, no. 3, pp. 255–261, 2003.

[4] A. D. Panopoulos, O. Yanes, S. Ruiz, Y. S. Kida, D. Diep, R. Tautenhahn, A. Herrerías, E. M. Batchelder, N. Plongthongkum, M. Lutz *et al.*, "The metabolome of induced pluripotent stem cells reveals metabolic changes occurring in somatic cell reprogramming," *Cell Research*, vol. 22, no. 1, pp. 168–177, 2012.

[5] M. Y. Hirai, M. Yano, D. B. Goodenowe, S. Kanaya, T. Kimura, M. Awazuhara, M. Arita, T. Fujiwara, and K. Saito, "Integration of transcriptomics and metabolomics for understanding of global responses to nutritional stresses in Arabidopsis thaliana," *Proceedings of the National academy of Sciences of the United States of America*, vol. 101, no. 27, pp. 10 205–10 210, 2004.

[6] O. Fiehn, "Metabolomics–the link between genotypes and phenotypes," *Plant Molecular Biology*, vol. 48, no. 1-2, pp. 155–171, 2002.

[7] M. L. Metzker, "Sequencing technologiesthe next generation," *Nature Reviews Genetics*, vol. 11, no. 1, pp. 31–46, 2010.

[8] A. Alonso, S. Marsal, and A. Julià, "Analytical methods in untargeted metabolomics: state of the art in 2015," *Frontiers in Bioengineering and Biotechnology*, vol. 3, p. 23, 2015.

[9] E. L. Ulrich, H. Akutsu, J. F. Doreleijers, Y. Harano, Y. E. Ioannidis, J. Lin, M. Livny, S. Mading, D. Maziuk, Z. Miller, E. Nakatani, C. F. Schulte, D. E. Tolmie, R. Kent Wenger, H. Yao, and J. L. Markley, "Biomagresbank," *Nucleic Acids Research*, vol. 36, no. suppl 1, pp. D402–D408, 2008.

[10] Q. Cui, I. A. Lewis, A. D. Hegeman, M. E. Anderson, J. Li, C. F. Schulte, W. M. Westler, H. R. Eghbalnia, M. R. Sussman, and J. L. Markley, "Metabolite identification via the madison metabolomics consortium database," *Nature biotechnology*, vol. 26, no. 2, pp. 162–164, 2008.

[11] Z. Pan and D. Raftery, "Comparing and combining NMR spectroscopy and mass spectrometry in metabolomics," *Analytical and Bioanalytical chemistry*, vol. 387, no. 2, pp. 525–527, 2007.

[12] J. Hao, W. Astle, M. De Iorio, and T. M. Ebbels, "BATMANan R package for the automated quantification of metabolites from nuclear magnetic resonance spectra using a Bayesian model," *Bioinformatics*, vol. 28, no. 15, pp. 2088–2090, 2012.

[13] X. Song, B.-L. Zhang, H.-M. Liu, B.-Y. Yu, X.-M. Gao, and L.-Y. Kang, "IQMNMR: Open source software using time-domain NMR data for automated identification and quantification of metabolites in batches," *BMC Bioinformatics*, vol. 12, no. 1, p. 1, 2011.

[14] R. Smith, A. D. Mathis, D. Ventura, and J. T. Prince, "Proteomics, lipidomics, metabolomics: a mass spectrometry tutorial from a computer scientist's point of view," *BMC Bioinformatics*, vol. 15, no. 7, p. 1, 2014.

[15] M. Cao, K. Fraser, J. Huege, T. Featonby, S. Rasmussen, and C. Jones, "Predicting retention time in hydrophilic interaction liquid chromatography mass spectrometry and its use for peak annotation in metabolomics," *Metabolomics*, vol. 11, no. 3, pp. 696–706, 2015.

[16] P. G. Pedrioli, J. K. Eng, R. Hubley, M. Vogelzang, E. W. Deutsch, B. Raught, B. Pratt, E. Nilsson, R. H. Angeletti, R. Apweiler *et al.*, "A common open representation of mass spectrometry data and its application to proteomics research," *Nature Biotechnology*, vol. 22, no. 11, pp. 1459–1466, 2004.

[17] L. Martens, M. Chambers, M. Sturm, D. Kessner, F. Levander, J. Shofstahl, W. H. Tang, A. Römpp, S. Neumann, A. D. Pizarro *et al.*, "mzML – a community standard for mass spectrometry data," *Molecular & Cellular Proteomics*, vol. 10, no. 1, pp. R110–000 133, 2011.

[18] R. Tautenhahn, C. Böttcher, and S. Neumann, "Highly sensitive feature detection for high resolution LC/MS," *BMC Bioinformatics*, vol. 9, no. 1, p. 1, 2008.

[19] T. Pluskal, S. Castillo, A. Villar-Briones, and M. Orešič, "Mzmine 2: modular framework for processing, visualizing, and analyzing mass spectrometry-based molecular profile data," *BMC Bioinformatics*, vol. 11, no. 1, p. 1, 2010.

[20] M. Katajamaa and M. Orešič, "Data processing for mass spectrometry-based metabolomics," *Journal of Chromatography A*, vol. 1158, no. 1, pp. 318–328, 2007.

[21] S. Castillo, P. Gopalacharyulu, L. Yetukuri, and M. Orešič, "Algorithms and tools for the preprocessing of LC-MS metabolomics data," *Chemometrics and Intelligent Laboratory Systems*, vol. 108, no. 1, pp. 23–32, 2011.

[22] A. Chokkathukalam, D.-H. Kim, M. P. Barrett, R. Breitling, and D. J. Creek, "Stable isotope-labeling studies in metabolomics: new insights into structure and dynamics of metabolic networks," *Bioanalysis*, vol. 6, no. 4, pp. 511–524, 2014.

[23] K. Podwojski, A. Fritsch, D. C. Chamrad, W. Paul, B. Sitek, P. Mutzel, C. Stephan, H. E. Meyer, W. Urfer, K. Ickstadt *et al.*, "Retention time alignment algorithms for lc/ms data must consider nonlinear shifts," *Bioinformatics*, p. btp052, 2009.

[24] C. Christin, A. K. Smilde, H. C. Hoefsloot, F. Suits, R. Bischoff, and P. L. Horvatovich, "Optimized time alignment algorithm for lc- ms data: correlation optimized warping using component detection algorithm-selected mass chromatograms," *Analytical Chemistry*, vol. 80, no. 18, pp. 7012–7021, 2008.

[25] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE transactions on acoustics, speech, and signal processing*, vol. 26, no. 1, pp. 43–49, 1978.

[26] N.-P. V. Nielsen, J. M. Carstensen, and J. Smedsgaard, "Aligning of single and multiple wavelength chromatographic profiles for chemometric data analysis using correlation optimised warping," *Journal of Chromatography A*, vol. 805, no. 1, pp. 17–35, 1998.

[27] W. Windig and W. F. Smith, "Chemometric analysis of complex hyphenated data. Improvements of the component detection algorithm." *Journal of Chromatography A*, vol. 1158, no. 1-2, pp. 251–7, 2007.

[28] a. M. van Nederkassel, M. Daszykowski, P. H. C. Eilers, and Y. V. Heyden, "A comparison of three algorithms for chromatograms alignment." *Journal of Chromatography A*, vol. 1118, no. 2, pp. 199–210, 2006.

[29] J. Listgarten, R. M. Neal, S. T. Roweis, and A. Emili, "Multiple alignment of continuous time series," in *Advances in Neural Information Processing Systems*, 2004, pp. 817–824.

[30] C. a. Smith, E. J. Want, G. O'Maille, R. Abagyan, and G. Siuzdak, "XCMS: processing mass spectrometry data for metabolite profiling using nonlinear peak alignment, matching, and identification." *Analytical Chemistry*, vol. 78, no. 3, pp. 779–87, 2006.

[31] E. Lange, C. Gröpl, O. Schulz-Trieglaff, A. Leinenbach, C. Huber, and K. Reinert, "A geometric approach for the alignment of liquid chromatographymass spectrometry data," *Bioinformatics*, vol. 23, no. 13, pp. i273–i281, 2007.

[32] M. A. Fischler and R. C. Bolles, "Random Sample Consensus : A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Communications of the ACM*, vol. 24, pp. 381–395, 1981.

[33] E. Lange, R. Tautenhahn, S. Neumann, and C. Gröpl, "Critical assessment of alignment procedures for {LC}-{MS} proteomics and metabolomics measurements," *BMC Bioinformatics*, vol. 9, p. 375, 2008.

[34] N. Hoffmann, M. Keck, and H. Neuweger, "Combining peak-and chromatogram-based retention time alignment algorithms for multiple chromatography-mass spectrometry datasets," *BMC Bioinformatics*, vol. 13, p. 214, 2012.

[35] R. Ballardini, M. Benevento, and G. Arrigoni, "MassUntangler: A novel alignment tool for label-free liquid chromatography-mass spectrometry proteomic data," *Journal of Chromatography A*, vol. 1218, no. 49, pp. 8859–68, 2011.

[36] B. Voss, M. Hanselmann, B. Y. Renard, M. S. Lindner, U. Köthe, M. Kirchner, and F. a. Hamprecht, "SIMA: simultaneous multiple alignment of LC/MS peak lists." *Bioinformatics*, vol. 27, no. 7, pp. 987–93, 2011.

[37] a. L. Duran, J. Yang, L. Wang, and L. W. Sumner, "Metabolomics spectral formatting, alignment and conversion tools (MSFACTs)," *Bioinformatics*, vol. 19, no. 17, pp. 2283–2293, 2003.

[38] J. Wang and H. Lam, "Graph-based peak alignment algorithms for multiple liquid chromatography-mass spectrometry datasets." *Bioinformatics*, vol. 29, no. 19, pp. 2469–2476, 2013.

[39] H. Lin, L. He, and B. Ma, "A Combinatorial Approach to the Peptide Feature Matching Problem for Label-Free Quantification." *Bioinformatics*, pp. 1–7, 2013.

[40] R. Smith, J. T. Prince, and D. Ventura, "A coherent mathematical characterization of isotope trace extraction , isotopic envelope extraction , and LC-MS correspondence," *BMC Bioinformatics*, vol. 16, no. Suppl 7, p. S1, 2015.

[41] R. Smith, D. Ventura, and J. T. Prince, "Novel algorithms and the benefits of comparative validation." *Bioinformatics*, vol. 29, no. 12, pp. 1583–5, 2013.

[42] ——, "LC-MS alignment in theory and practice: a comprehensive algorithmic review," *Briefings in Bioinformatics*, 2013.

[43] H. P. Benton, E. J. Want, and T. M. Ebbels, "Correction of mass calibration gaps in liquid chromatography–mass spectrometry metabolomics data," *Bioinformatics*, vol. 26, no. 19, pp. 2488–2489, 2010.

[44] R. K. Snider, "Efficient calculation of exact mass isotopic distributions." *Journal of the American Society for Mass Spectrometry*, vol. 18, no. 8, pp. 1511–5, 2007.

[45] B. O. Keller, J. Sui, A. B. Young, and R. M. Whittal, "Interferences and contaminants encountered in modern mass spectrometry." *Analytica chimica acta*, vol. 627, no. 1, pp. 71–81, 2008.

[46] R. Scheltema, S. Decuypere, and J. Dujardin, "Simple data-reduction method for high-resolution LC-MS data in metabolomics," *Bioanalysis*, 2009.

[47] C. Kuhl, R. Tautenhahn, C. Böttcher, T. R. Larson, and S. Neumann, "CAMERA: an integrated strategy for compound spectra extraction and annotation of liquid chromatography/mass spectrometry data sets." *Analytical Chemistry*, vol. 84, no. 1, pp. 283–9, 2012.

[48] R. a. Scheltema, A. Jankevics, R. C. Jansen, M. a. Swertz, and R. Breitling, "PeakML/mzMatch: a file format, Java library, R library, and tool-chain for mass spectrometry data analysis." *Analytical Chemistry*, vol. 83, no. 7, pp. 2786–93, 2011.

[49] S. Rogers, R. Daly, and R. Breitling, "Mixture model clustering for peak filtering in metabolomics," in *Ninth International Workshop on Computational Systems Biology, WCSB 2012, June 4-6, Ulm, Germany*, 2012, p. 71.

[50] L. W. Sumner, A. Amberg, D. Barrett, M. H. Beale, R. Beger, C. A. Daykin, T. W.-M. Fan, O. Fiehn, R. Goodacre, J. L. Griffin *et al.*, "Proposed minimum reporting standards for chemical analysis," *Metabolomics*, vol. 3, no. 3, pp. 211–221, 2007.

[51] T. Kind and O. Fiehn, "Metabolomic database annotations via query of elemental compositions: mass accuracy is insufficient even at less than 1 ppm." *BMC Bioinformatics*, vol. 7, p. 234, 2006.

[52] W. Dunn, A. Erban, R. Weber, and D. Creek, "Mass appeal: metabolite identification in mass spectrometry-focused untargeted metabolomics," *Metabolomics*, 2012.

[53] R. R. da Silva, P. C. Dorrestein, and R. A. Quinn, "Illuminating the dark matter in metabolomics," *Proceedings of the National Academy of Sciences*, vol. 112, no. 41, p. 201516878, 2015.

[54] D. J. Creek, A. Jankevics, R. Breitling, D. G. Watson, M. P. Barrett, and K. E. V. Burgess, "Toward global metabolomics analysis with hydrophilic interaction liquid chromatography–mass spectrometry: improved metabolite identification by retention time prediction," *Analytical Chemistry*, vol. 83, no. 22, pp. 8703–8710, 2011.

[55] J. Stanstrup, S. Neumann, and U. Vrhov??ek, "PredRet: Prediction of Retention Time by Direct Mapping between Multiple Chromatographic Systems," *Analytical Chemistry*, vol. 87, no. 18, pp. 9421–9428, 2015.

[56] S. Rogers, R. a. Scheltema, M. Girolami, and R. Breitling, "Probabilistic assignment of formulas to mass peaks in metabolomics experiments," *Bioinformatics*, vol. 25, no. 4, pp. 512–518, 2009.

[57] R. Daly, S. Rogers, J. Wandy, A. Jankevics, K. E. V. Burgess, and R. Breitling, "MetAssign: probabilistic annotation of metabolites from LC-MS data using a Bayesian clustering approach." *Bioinformatics*, vol. 30, no. 19, pp. 2764–2771, 2014.

[58] F. Hufsky, K. Scheubert, and S. Böcker, "Computational mass spectrometry for small-molecule fragmentation," *TrAC - Trends in Analytical Chemistry*, vol. 53, pp. 41–48, 2014.

[59] M. Kotera, M. Hirakawa, T. Tokimatsu, S. Goto, and M. Kanehisa, "The kegg databases and tools facilitating omics analysis: latest developments involving human diseases and pharmaceuticals," *Next Generation Microarray Bioinformatics: Methods and Protocols*, pp. 19–39, 2012.

[60] H. Horai, M. Arita, S. Kanaya, Y. Nihei, T. Ikeda, K. Suwa, Y. Ojima, K. Tanaka, S. Tanaka, K. Aoshima *et al.*, "Massbank: a public repository for sharing mass spectral data for life sciences," *Journal of Mass Spectrometry*, vol. 45, no. 7, pp. 703–714, 2010.

[61] H. E. Pence and A. Williams, "Chemspider: an online chemical information resource," *Journal of Chemical Education*, vol. 87, no. 11, pp. 1123–1124, 2010.

[62] J. Xia and D. S. Wishart, "MetPA: a web-based metabolomics tool for pathway analysis and visualization." *Bioinformatics*, vol. 26, no. 18, pp. 2342–4, 2010.

[63] J. Krumsiek, K. Suhre, T. Illig, J. Adamski, and F. J. Theis, "Gaussian graphical modeling reconstructs pathway reactions from high-throughput metabolomics data," *BMC Systems Biology*, vol. 5, no. 1, p. 1, 2011.

[64] J. Krumsiek, K. Suhre, A. M. Evans, M. W. Mitchell, R. P. Mohney, M. V. Milburn, B. Wägele, W. Römisch-Margl, T. Illig, J. Adamski, C. Gieger, F. J. Theis, and G. Kastenmüller, "Mining the unknown: a systems approach to metabolite identification combining genetic and metabolic information." *PLoS Genetics*, vol. 8, no. 10, p. e1003005, 2012.

[65] "Genetics meets metabolomics: a genome-wide association study of metabolite profiles in human serum." *PLoS Genetics*, vol. 4, no. 11, p. e1000282, 2008.

[66] R. C. H. De Vos, S. Moco, A. Lommen, J. J. B. Keurentjes, R. J. Bino, and R. D. Hall, "Untargeted large-scale plant metabolomics using liquid chromatography coupled to mass spectrometry." *Nature Protocols*, vol. 2, no. 4, pp. 778–791, 2007.

[67] M. Mamas, W. B. Dunn, L. Neyses, and R. Goodacre, "The role of metabolites and metabolomics in clinically applicable biomarkers of disease," *Archives of Toxicology*, vol. 85, no. 1, pp. 5–17, 2011.

[68] "Metabolomics in human nutrition: opportunities and challenges." *The American Journal of Clinical Nutrition*, vol. 82, no. 3, pp. 497–503, 2005.

[69] D. B. Kell, "Systems biology, metabolic modelling and metabolomics in drug discovery and development." *Drug Discovery Today*, vol. 11, no. 23-24, pp. 1085–92, 2006.

[70] M. Kanehisa, S. Goto, M. Furumichi, M. Tanabe, and M. Hirakawa, "Kegg for representation and analysis of molecular networks involving diseases and drugs," *Nucleic Acids Research*, vol. 38, no. suppl 1, pp. D355–D360, 2010.

[71] R. Caspi, H. Foerster, C. A. Fulcher, P. Kaipa, M. Krummenacker, M. Latendresse, S. Paley, S. Y. Rhee, A. G. Shearer, C. Tissier *et al.*, "The MetaCyc Database of metabolic pathways and enzymes and the BioCyc collection of Pathway/Genome Databases," *Nucleic Acids Research*, vol. 36, no. suppl 1, pp. D623–D631, 2008.

[72] L. Cottret, D. Wildridge, F. Vinson, M. P. Barrett, H. Charles, M.-F. Sagot, and F. Jourdan, "Metexplore: a web server to link metabolomic experiments and genome-scale metabolic networks," *Nucleic Acids Research*, vol. 38, no. suppl 2, pp. W132–W137, 2010.

[73] M. Sandin, J. Teleman, J. Malmström, and F. Levander, "Data processing methods and quality control strategies for label-free LC-MS protein quantification." *Biochimica et biophysica acta*, vol. 1844, no. 1 Pt A, pp. 29–41, 2014.

[74] A. Chawade, M. Sandin, J. Teleman, J. Malmström, and F. Levander, "Data processing has major impact on the outcome of quantitative label-free LC-MS analysis." *Journal of Proteome Research*, vol. 14, no. 2, pp. 676–87, 2015.

[75] E. de Hoffmann and V. Stroobant, *Mass spectrometry: Principles and applications*, 3rd ed., L. John Wiley & Sons, Ed., West Sussex, England, 2007.

[76] J. H. Gross, *Mass Spectrometry: A Textbook*. Springer Science & Business Media, 2006.

[77] H. G. Gika, G. A. Theodoridis, R. S. Plumb, and I. D. Wilson, "Current practice of liquid chromatography-mass spectrometry in metabolomics and metabonomics," *Journal of Pharmaceutical and Biomedical Analysis*, vol. 87, no. March 2016, pp. 12–25, 2014.

[78] D. A. Megger, T. Bracht, H. E. Meyer, and B. Sitek, "Label-free quantification in clinical proteomics," *Biochimica et Biophysica Acta - Proteins and Proteomics*, vol. 1834, no. 8, pp. 1581–1590, 2013.

[79] R. Xu and D. Wunsch, "Survey of Clustering Algorithms," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.

[80] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern recognition letters*, vol. 31, no. 8, pp. 651–666, 2010.

[81] D. P. De Souza, E. C. Saunders, M. J. McConville, and V. a. Likić, "Progressive peak clustering in GC-MS Metabolomic experiments applied to Leishmania parasites." *Bioinformatics*, vol. 22, no. 11, pp. 1391–6, 2006.

[82] A. M. Frank, "Clustering Millions of Tandem Mass Spectra," *Journal of Proteome Research*, vol. 7, pp. 113–122, 2007.

[83] "ProbMetab: an R package for Bayesian probabilistic annotation of LC-MS-based metabolomics." *Bioinformatics*, pp. 1–2, 2014.

[84] F. Allen, R. Greiner, and D. Wishart, "Competitive fragmentation modeling of ESI-MS/MS spectra for putative metabolite identification," *Metabolomics*, vol. 11, no. 1, pp. 98–110, 2014.

[85] F. Allen, A. Pon, R. Greiner, and D. S. Wishart, "Computational prediction of electron ionization mass spectra to assist in GC-MS compound identification," *Analytical Chemistry*, vol. 88, pp. 7689–7697, 2016.

[86] M. Heinonen, H. Shen, N. Zamboni, and J. Rousu, "Metabolite identification and molecular fingerprint prediction through machine learning," *Bioinformatics*, vol. 28, no. 18, pp. 2333–2341, 2012.

[87] K. Dührkop, H. Shen, M. Meusel, J. Rousu, and S. Böcker, "Searching molecular structure databases with tandem mass spectra using csi: Fingerid," *Proceedings of the National Academy of Sciences*, vol. 112, no. 41, pp. 12 580–12 585, 2015.

[88] C. E. Rasmussen, "The infinite Gaussian mixture model," vol. 12, 2000, pp. 554–560.

[89] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian Data Analysis*. Chapman & Hall, 2014, vol. 2.

[90] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[91] N. L. Hjort, C. Holmes, P. Müller, and S. G. Walker, *Bayesian Nonparametrics*. Cambridge University Press, 2010, vol. 28.

[92] T. S. Ferguson, "A Bayesian Analysis of Some Nonparametric Problems," *The Annals of Statistics*, pp. 209–230, 1973.

[93] Y. W. Teh, "Dirichlet Process," in *Encyclopedia of Machine Learning*. Springer, 2011, pp. 280–287.

[94] H. Ishwaran and L. F. James, "Gibbs sampling methods for stick-breaking priors," *Journal of the American Statistical Association*, 2011.

[95] D. J. Aldous, "Exchangeability and related topics," in *École d'Été de Probabilités de Saint-Flour XIII1983*. Springer, 1985, pp. 1–198.

[96] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Hierarchical dirichlet processes," *Journal of the American Statistical Association*, vol. 101, no. 476, pp. 1566–1581, 2006.

[97] S. Kim and P. Smyth, "Hierarchical dirichlet processes with random effects," in *Advances in Neural Information Processing Systems*, 2006, pp. 697–704.

[98] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," *Journal of Machine Learning Research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[99] S. Rogers, M. Girolami, C. Campbell, and R. Breitling, "The latent process decomposition of cdna microarray data sets," *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, vol. 2, no. 2, pp. 143–156, 2005.

[100] D. Weinshall, G. Levi, and D. Hanukaev, "LDA Topic Model with Soft Assignment of Descriptors to Words," in *Proceedings of the 30th Annual International Conference on Machine Learning*, vol. 28.

[101] R. Das, M. Zaheer, and C. Dyer, "Gaussian LDA for Topic Models with Word Embeddings," in *Proceedings of the 53nd Annual Meeting of the Association for Computational Linguistics*, 2015.

[102] B. Carpenter, "Integrating out multinomial parameters in Latent Dirichlet Allocation and Naive Bayes for collapsed Gibbs sampling," Tech. Rep., 2010. [Online]. Available: https://lingpipe.files.wordpress.com/2010/07/lda3.pdf

[103] "Retention time alignment algorithms for LC/MS data must consider non-linear shifts." *Bioinformatics*, vol. 25, no. 6, pp. 758–64, 2009.

[104] J. Wandy, R. Daly, R. Breitling, and S. Rogers, "Incorporating peak grouping information for alignment of multiple liquid chromatography-mass spectrometry datasets," *Bioinformatics*, vol. 31, no. 12, pp. 1999–2006, 2015.

[105] D. Gusfield and R. Irving, *The stable marriage problem: structure and algorithms*, 1989.

[106] H. Kuhn, "The Hungarian method for the assignment problem," *Naval research logistics quarterly*, 1955.

[107] R. Duan, S. Pettie, and H. Su, "Scaling algorithms for approximate and exact maximum weight matching," *arXiv preprint arXiv:1112.0790*, pp. 1–36, 2011.

[108] E. Melamud, L. Vastag, and J. D. Rabinowitz, "Metabolomic analysis and visualization engine for LC-MS data." *Analytical Chemistry*, vol. 82, no. 23, pp. 9818–26, 2010.

[109] L. Brodsky, A. Moussaieff, N. Shahaf, A. Aharoni, and I. Rogachev, "Evaluation of peak picking quality in LC-MS metabolomics data." *Analytical Chemistry*, vol. 82, no. 22, pp. 9177–87, 2010.

[110] G. Landan and D. Graur, "Characterization of pairwise and multiple sequence alignment errors." *Gene*, vol. 441, no. 1-2, pp. 141–7, 2009.

[111] J. D. Thompson, D. G. Higgins, and T. J. Gibson, "CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice." *Nucleic Acids Research*, vol. 22, no. 22, pp. 4673–80, 1994.

[112] C. Notredame, D. G. Higgins, and J. Heringa, "T-Coffee: A novel method for fast and accurate multiple sequence alignment." *Journal of Molecular Biology*, vol. 302, no. 1, pp. 205–17, 2000.

[113] O. Penn, E. Privman, G. Landan, D. Graur, and T. Pupko, "An alignment confidence score capturing robustness to guide tree uncertainty." *Molecular Biology and Evolution*, vol. 27, no. 8, pp. 1759–67, 2010.

[114] B. D. Redelings and M. a. Suchard, "Joint Bayesian estimation of alignment and phylogeny." *Systematic Biology*, vol. 54, no. 3, pp. 401–18, 2005.

[115] R. K. Bradley, A. Roberts, M. Smoot, S. Juvekar, J. Do, C. Dewey, I. Holmes, and L. Pachter, "Fast statistical alignment." *PLoS Computational Biology*, vol. 5, no. 5, p. e1000392, 2009.

[116] J. Jeong, X. Shi, X. Zhang, S. Kim, and C. Shen, "Model-based peak alignment of metabolomic profiling from comprehensive two-dimensional gas chromatography mass spectrometry." *BMC Bioinformatics*, vol. 13, no. 1, p. 27, 2012.

[117] M. Ghanat Bari, X. Ma, and J. Zhang, "PeakLink: a new peptide peak linking method in LC-MS/MS using wavelet and SVM." *Bioinformatics*, vol. 30, no. 17, pp. 2464–70, 2014.

[118] M. Sandin, A. Ali, K. Hansson, O. Månsson, and E. Andreasson, "An Adaptive Alignment Algorithm for Quality-controlled Label-free LC-MS," *Molecular & Cellular Proteomics*, pp. 1407–1420, 2013.

[119] R. Tibshirani, T. Hastie, B. Narasimhan, S. Soltys, G. Shi, A. Koong, and Q.-T. Le, "Sample classification from protein mass spectrometry, by peak probability contrasts," *Bioinformatics*, vol. 20, no. 17, pp. 3034–3044, 2004.

[120] V. Perera, M. D. T. Zabala, H. Florance, N. Smirnoff, M. Grant, and Z. R. Yang, "Aligning extracted lc-ms peak lists via density maximization," *Metabolomics*, vol. 8, no. 1, pp. 175–185, 2012.

[121] C. Wang and D. M. Blei, "A split-merge mcmc algorithm for the hierarchical dirichlet process," *arXiv preprint arXiv:1201.1657*, 2012.

[122] E. E. Bolton, Y. Wang, P. A. Thiessen, and S. H. Bryant, "Pubchem: integrated platform of small molecules and biological activities," *Annual Reports in Computational Chemistry*, vol. 4, pp. 217–241, 2008.

[123] T. Kind and O. Fiehn, "Seven Golden Rules for heuristic filtering of molecular formulas obtained by accurate mass spectrometry." *BMC Bioinformatics*, vol. 8, p. 105, 2007.

[124] C. a. Smith, G. O'Maille, E. J. Want, C. Qin, S. a. Trauger, T. R. Brandon, D. E. Custodio, R. Abagyan, and G. Siuzdak, "METLIN: a metabolite mass spectral database." *Therapeutic Drug Monitoring*, vol. 27, no. 6, pp. 747–51, 2005.

[125] K. Varmuza and W. Werther, "Mass Spectral Classifiers for Supporting Systematic Structure Elucidation," *Journal of Chemical Information and Modeling*, vol. 36, no. 2, pp. 323–333, 1996.

[126] J. Hummel, N. Strehmel, J. Selbig, D. Walther, and J. Kopka, "Decision tree supported substructure prediction of metabolites from GC-MS profiles," *Metabolomics*, vol. 6, no. 2, pp. 322–333, 2010.

[127] J. Y. Yang, L. M. Sanchez, C. M. Rath, X. Liu, P. D. Boudreau, N. Bruns, E. Glukhov, A. Wodtke, R. de Felicio, A. Fenner *et al.*, "Molecular networking as a dereplication strategy," *Journal of Natural Products*, vol. 76, no. 9, pp. 1686–1699, 2013.

[128] D. D. Nguyen, C.-H. Wu, W. J. Moree, A. Lamsa, M. H. Medema, X. Zhao, R. G. Gavilan, M. Aparicio, L. Atencio, C. Jackson *et al.*, "MS/MS networking guided analysis of molecule and gene cluster families," *Proceedings of the National Academy of Sciences*, vol. 110, no. 28, pp. E2611–E2620, 2013.

[129] J. J. Van Der Hooft, S. Padmanabhan, K. E. Burgess, and M. P. Barrett, "Urinary antihypertensive drug metabolite screening using molecular networking coupled to high-resolution mass spectrometry fragmentation," *Metabolomics*, 2016.

[130] Y. Ma, T. Kind, D. Yang, C. Leon, and O. Fiehn, "Ms2analyzer: A software for small molecule substructure annotations from accurate tandem mass spectra," *Analytical Chemistry*, vol. 86, no. 21, pp. 10 724–10 731, 2014.

[131] D. L. Sweeney, "A Data Structure for Rapid Mass Spectral Searching," *Mass Spectrometry*, vol. 3, no. Special_Issue_2, pp. S0035–S0035, 2014.

[132] D. R. Scott, "Pattern recognition/expert system for identification of toxic compounds from low resolution mass spectra," *Chemometrics and intelligent laboratory systems*, vol. 23, no. 2, pp. 351–364, 1994.

[133] X. Chen, X. Hu, X. Shen, and G. Rosen, "Probabilistic topic modeling for genomic data interpretation," in *Bioinformatics and Biomedicine (BIBM), 2010 IEEE International Conference on*. IEEE, 2010, pp. 149–152.

[134] R. Zhang, Z. Cheng, J. Guan, and S. Zhou, "Exploiting topic modeling to boost metagenomic reads binning," *BMC Bioinformatics*, vol. 16, no. Suppl 5, p. S2, 2015.

[135] S. Böcker, M. C. Letzel, Z. Lipták, and A. Pervukhin, "SIRIUS: decomposing isotope patterns for metabolite identification." *Bioinformatics*, vol. 25, no. 2, pp. 218–24, 2009.

[136] M. a. Stravs, E. L. Schymanski, H. P. Singer, and J. Hollender, "Automatic recalibration and processing of tandem mass spectra using formula annotation," *Journal of Mass Spectrometry*, vol. 48, no. 1, pp. 89–99, 2013.

[137] C. Sievert and K. Shirley, "LDAvis: A method for visualizing and interpreting topics," *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*, pp. 63–70, 2014.

[138] S. Bocker and Z. Liptak, "A Fast and Simple Algorithm for the Money Changing Problem," *Algorithmica*, vol. 48, no. 4, pp. 413–432, 2007.

[139] H. M. Wallach, I. Murray, R. Salakhutdinov, and D. Mimno, "Evaluation methods for topic models," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 1105–1112.

[140] T. L. Griffiths and M. Steyvers, "Finding scientific topics." *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101 Suppl, pp. 5228–5235, 2004.

[141] J. Tomfohr, J. Lu, and T. B. Kepler, "Pathway level analysis of gene expression using singular value decomposition," *BMC Bioinformatics*, vol. 6, no. 1, p. 1, 2005.

[142] A. L. Tarca, G. Bhatti, and R. Romero, "A comparison of gene set analysis methods in terms of sensitivity, prioritization and specificity," *PloS One*, vol. 8, no. 11, p. e79217, 2013.

[143] T. P. Minka, "Estimating a Dirichlet distribution," *Annals of Physics*, vol. 2000, no. 8, pp. 1–13, 2003.

[144] D. Griffiths and M. Tenenbaum, "Hierarchical topic models and the nested chinese restaurant process," *Advances in Neural Information Processing Systems*, vol. 16, p. 17, 2004.

[145] J.-H. Kang, J. Ma, and Y. Liu, *Transfer Topic Modeling with Ease and Scalability*, ch. 48, pp. 564–575.