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

MARCH 2016

Abstract

This is a dissertation outline using the style guidelines defined by the University of Glasgow.

Acknowledgements

ACK

Dedication. (Is what you need.)

Table of Contents

List of Tables

List of Figures

Todo list

Chapter 1

Introduction

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

1.1 Thesis Statement

1.2 Overview of Thesis and Research Contributions

Chapter 2

Computational Biology Background

This chapter provides the background knowledge necessary to understand the basic principles of mass-spectrometry-based analysis as applied to large-scale untargeted biological studies. A particular emphasis is given to the application of mass spectrometry techniques to the field of metabolomics. For a further reading on mass spectrometry, the reader is directed to a more comprehensive textbooks such as [1] and [2]. Reviews on the necessary data preprocessing steps of mass spectrometry data can be found for e.g. in proteomics [3, 4, 5] and metabolomic [6, 7, 8].

2.1 Computational Biology

Since the discovery of the deoxyribonucleic acid (DNA) as the basic storage of genetic information, the same fundamental principle is found to govern the transmission of hereditary information common to all life on Earth. Following the central dogma of molecular biology:

DNA is transcribed into RNA, which is translated into proteins.

In the central dogma, genetic materials are coded in the DNA, a strand of which consists of a series of nucleotides. The backbone of a nucleotide comprises sugar and phosphate groups attached to one of the four nitrogenous bases of **A**denine, **T**hymine, **G**uanine, and **C**ytosine, forming the four well-known alphabets of the DNA. The start of the transcription process begins with the unwounding of the double strand of the DNA into single strands. The single DNA strand is transcribed by a protein complex (RNA polymerase) into messenger ribonucleic acid (RNA), which can be be thought as nearly identical to DNA, with the crucial difference that the base **U**racil is used in place of thymine. The substitution of thymine to uracil allows RNA to perform its important function as the messenger RNA, which as the name suggests is the intermediate mechanism of messaging for the information contained in

the chemically inert DNA. Messenger RNA is read by the ribosome, a part of the translational apparatus of the cell, and translated into amino acids, which are the building blocks of proteins.

Since its initial proposal, this simple central dogma model has been challenged and expanded to acknowledge other factors that can influence the transcription and translation processes. Nevertheless, the central dogma serves to illustrate the flow of genetic information in a biological system. Different sub-fields of computational biology predominantly study the different entities and processes involved in the central dogma. Genomics is concerned with the large-scale study of the genome (the entire DNA in the organism) and how the genes encoded in the genome interact with each other. 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 [9]. Transcriptomics focuses on understanding the transcriptome (the complete set of messenger RNA) and their measurement. Transcriptome relies on DNA micro-array technologies and more recently, have been increasingly performed by NGS sequencing as well. Proteins and their large-scale identifications and quantifications are studied in proteomics, while the complete set metabolites present in the sample and their expressions are the focus of metabolomics. Each successive layer of the -omics hierarchy, which comes closer to the actual physical expression of observable traits (phenotypes), introduces more complexity due to the increased number of ways of putting the building blocks in that layer together (Figure 6.1).

Draw this

Figure 2.1: The building blocks of the genome are the DNA nucleotides. In the transcriptome, the building blocks are the nitrogenous bases that comprise the RNA. In proteomics, the 20 amino acids residues make up the polypeptide comprising a protein molecule. In contrast, the building blocks of metabolites are the atoms (usually CHNOPS: carbon, hydrogen, nitrogen, oxygen, phospor and sulphur) that comprise a large range of compounds, such as lipids, amino acids, vitamins, etc., with varying physical and chemical properties

2.2 Mass spectrometry-based omics

The two -omics closest to the phenotype in Figure 6.1 rely on mass spectrometry, usually coupled to a separation intrument such as chromatography, to perform measurements and quantification of the biological entities of interest. Proteomics and metabolomics are briefly expanded in this section, while details on the set-up of the instruments can be found in Section 2.3.

Proteins, considered the building blocks of life, serve critical roles in an organism by performing cellular maintenance, catalysing chemical reactions, carrying molecules across cell membranes and many other essential functions. The primary building blocks of a protein are amino acids, which results from the translation of messenger RNA. An amino acid consists of the amine group (-NH_2), a carboxylic group (-COOH) and a side chain. Through the loss of water molecule, amino acids can be chained to each other through the peptide bonds, collectively forming a peptide. Each amino acid can be described by a unique letter drawn from a set of 20 chemical alphabets, and consequently a peptide can be succinctly described as a string of letters corresponding to its constituent amino acids. While the sequence of amino acids comprising a protein is largely coded by genes in the genome, this process is far from deterministic. In a process called post-translational modification [10], proteins can be chemically modified after synthesis in a way that completely alters its structure and folding stability, e.g. through phosphorylation, methylation or glycosylation. This results in a large variety of protein diversity present in the biological system, and it is the large-scale characterisation of identities and quantities of proteins that is of particular interest to **proteomics**.

Apart from proteins, numerous other chemical reactions essential for sustaining life also happen inside a cell. 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 the molecules 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, and vitamins (e.g. B3 and B12) and minerals (e.g. phosphorus, iron and zinc). The overall set of metabolites that can be found within an organism is collectively called the metabolome. **Metabolomics** studies the metabolome on a large scale, usually for the purpose of identifying and quantifying their differences in the particular organisms or tissues under various experimental or physiological conditions. As metabolomics as a study is considered to be the closest to the phenotype, changes to the metabolome often result in physically observed properties, and indeed changes in the metabolite composition of an organism may be

caused by responses to environmental and genetic factors [11]. Studying the metabolome provides us with an instantaneous 'snapshot' of the chemical activities that are occurring in the cell at that moment.

2.3 Mass Spectrometry

Atoms are 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 hold together 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. The term 'mono-isotopic' refers to the most abundant isotope species of an element. The exact mass of a compound can therefore be calculated from the formula sum of the masses of its constituent mono-isotopes. The nominal mass of a compound is similarly calculated by summing the integer masses of the constituent mono-isotopes (e.g. the nominal mass of $H_2O = 1 + 1 + 16 = 18$).

Mass spectrometer (MS) coupled to liquid chromatography, forming the set-up of liquid chromatography mass spectrometry (LC-MS), is the preferred measurement platform for determining the elemental composition and the abundance of the analytes (proteins or metabolites) in proteomics or metabolomics studies. 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 $= 1e6 * \frac{\text{(observed m/z}-\text{exact m/z)}}{\text{exact m/z}}$.

2.3.1 Metabolomics

In recent years, the combination of liquid chromatography coupled to mass spectrometry (LC-MS) has emerged as one of the most widely used techniques in untargeted metabolomic studies. Metabolites in the extracted sample cannot be introduced at once as direct injection into MS due to ion supression effect [12], where compounds 'compete' for charges during the ionisation process inside the MS. Due to this ion supression effect, metabolites present in low abundance might not be ionised and therefore not detected in the resulting mass spectra. As a result, it is necessary for metabolites to be separated before being introduced gradually into the inlet of the ionisation source. Separation techniques such as liquid chromatography LC coupled to MS is commonly used for this purpose. In liquid chromatography, the mobile solvent containing the analytes (metabolites) is introduced and pumped into the stationary phase of the chromatographic column. Metabolites elutes at different time through their interactions with the capillary, based on the hydrophobicity, charge and other chemical properties of the metabolites. The time it takes for these metabolites to elute through the stationary phase of the LC column is called the retention time (RT). LC-MS tends to be easier to automate and suitable for high throughput experiments. Sample preparations for an LC-MS set-up also tends to be simpler compared to the alternative of separation via gas chromatography, while compounds across a wide range of polarity can be separated [1].

Metabolites that elute from liquid chromatography are then vaporised and ionised inside the mass spectrometer. This is usually accomplished through soft-ionisation methods such as atmospheric pressure ionisation or electrospray ionisation (ESI). The distinction between soft- and hard- ionisation methods come from how 'soft' methods do not break the chemical bonds of the compound during the ionisation process, which stands in contrast to hard-ionisation methods, such as the electron impact ionisation, that breaks the chemical bonds in the neutral molecules of compounds. ESI can be directly coupled to LC, so often, it is the preferred method of ionisation. In ESI, the sample analyte is dissolved into a solvent and sprayed through an electrospray. It is the resulting charged aerosol that enters the vacuum of the mass spectrometer, 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. The result of this process is a mass spectrum: a two dimensional representation of m/z values to signal intensities. The final raw data produced by an LC-MS setup is called the ion chromatograms: a collection of mass spectra over the range of elution time. The entire raw data can therefore be characterised by a set of vector of m/z, intensity and retention time, and for every slice on the ion chromatogram sharing the same RT value (a scan), a mass spectrum is produced from metabolites that elute at that same retention time. A mass spectrum is the observed m/z and intensity (abundance) values of the peaks that result

from fragmentations of the metabolites during the scan. When the MS instrument is run on the full-scan mode, the entire m/z range is selected for fragmentation.

Redraw this to be simpler.

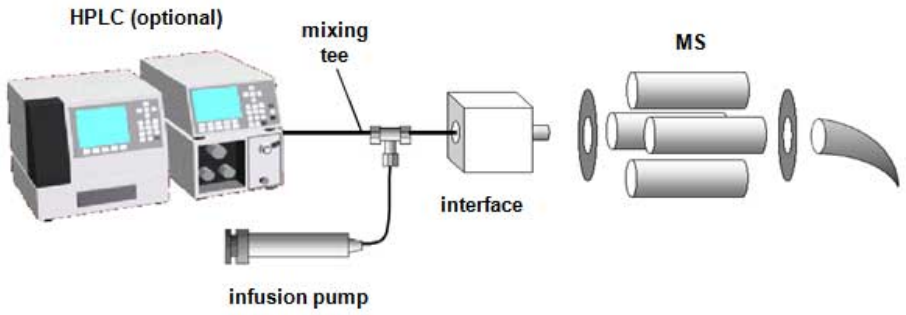

Figure 2.2: A typical LC-MS set-up. High performance liquid chromatography instruments are used to separate metabolites (by their chemical properties) in the sample before they are gradually introduced into the mass spectrometer.

Redraw this to illustrate the point better.

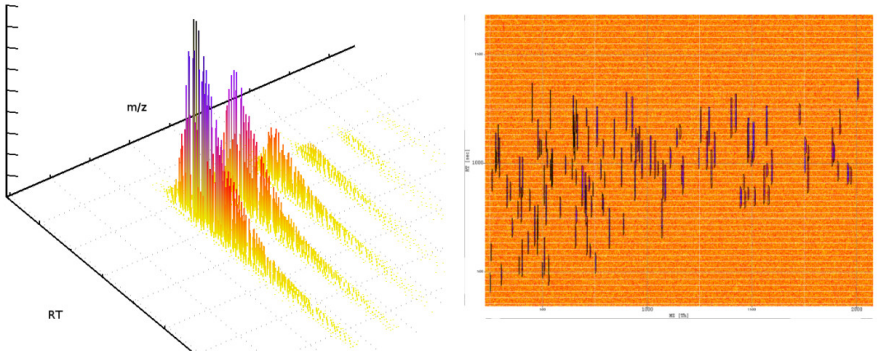

Figure 2.3: The resulting raw data (ion chromatograms) produced from an LC-MS experiment. We can view the data as a 3D profile (left) and as a 2D profile seen from the top (right). A slice of the data on the m/z axis is the mass spectrum. Each mass spectrum is produced by a scan of the mass spectrometer. A collection of mass spectra is produced over the whole range of retention time. A point in the raw data is thus characterised by its intensity value on the m/z and retention time axes.

2.3.2 Proteomics and Glyomics

For 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. Different types of samples will demand the appropriate sample handling protocol in the sample preparation stage. 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. An initial separation process (prefactionation) can also be performed on the digested peptides using liquid chromatography, resulting in different fractions, which can then be ran separately through the hypenated set-up of LC-MS (Figure 2.2) for mass fragmentation analysis in a manner similar to metabolomics analysis (described in the following paragraphs in Section 2.3.1). This yields the peptide mass fingerprint, which although challenging, can generally be used to match the resulting peptide fingerprints against a database of reference fingerprints for identification of the peptides and correspondingly the entire protein.

2.3.3 Fragmentation

Fragmentation through tandem MS or MS^n instruments can be used to provide further fragmentation information for metabolite 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. This is described further in Section 2.4.3. 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 peptides/metabolites being fragmented together in the same m/z window, and require

sophisticated analysis strategy to deconvolve the signals from the noise.

2.4 Metabolomics Pipeline: From Raw Data to Biological Hypothesis

The raw LC-MS data is noisy, so pre-processing has to take place before analysis can be performed and biological conclusion drawn. The raw LC-MS data has to go through successive pre-processing and transformations along the data pre-processing pipeline before it can be analysed. The main steps of LC-MS data preprocessing generally involve peak detection and the filtering of noise, the matching of identical peaks across samples (alignment), identifications of peaks and lastly, data normalization and visualization. 2.4 shows these key preprocessing steps in the typical LC-MS data processing pipeline, which is elaborated further next.

Redraw this to make it look nicer..?

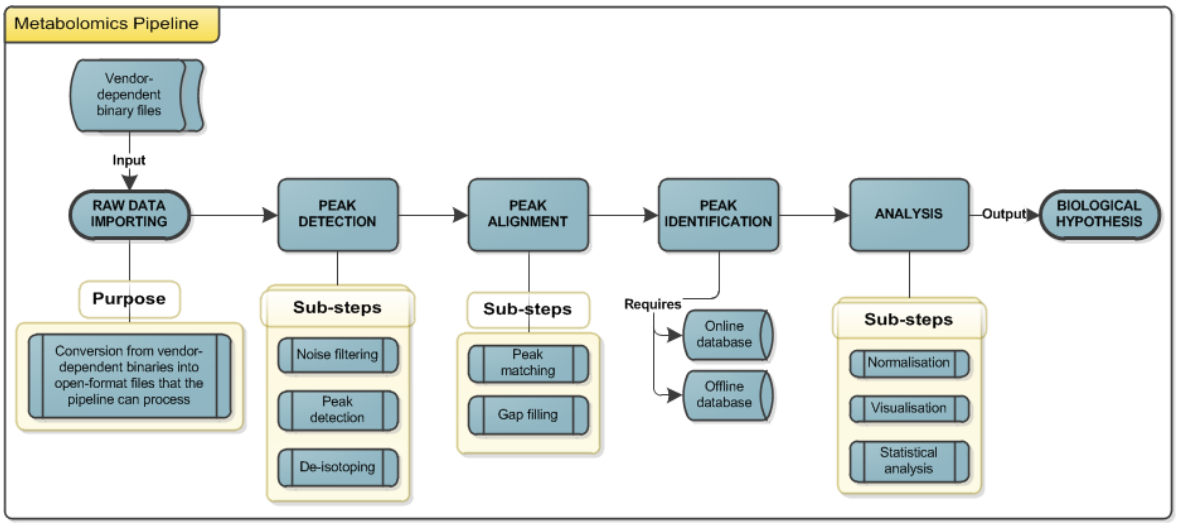

Figure 2.4: Preprocessing pipeline of LC-MS metabolomics data.

2.4.1 Peak Detection

The raw LC-MS data is imported into the pipeline. Beginning as a vendor-proprietary format, the raw data is converted into open XML-based format for storing mass spectrometry data, such as the mzXML or mzML format [13]. Noise filtering is performed as a preliminary filtering to remove noises and artifact signals due to the various chemical noises that also occur during the ionisation process. Peak detection is then performed to identify areas and intensities of peaks. A survey of the different approaches towards peak detection can be

found in [14], but what is important to note is at this stage, additional peaks can potentially be introduced due to peaks falsely detected from chemical noises, e.g. as a result of the contaminants present in the sample, while on the other hand, peaks that should be detected can instead be missing as a consequence of setting incorrect parameters for the detection step, e.g. by setting threshold values that are too low.

Additionally, not all observed peaks would correspond to true precursor ions of the metabolites, since peaks could also be generated by other entities sharing the same identifying mass value, due to the presence of isotopes, contaminants, adducts and other signal artifacts in the sample ([15]). In particular, due to the presence of naturally occurring isotopes (e.g. ^{13}C) and the formation of adducts (the addition of a molecule ion to another), one precursor ion corresponding to a single metabolite alone can produce many observed peaks in the mass spectrum, forming a distribution of isotopic peaks at different m/z values but having similar chromatographic peak shapes in their elution time profiles. As one of the main challenges of peak detection comes from the presence of these isotope peaks, the de-isotoping step is often performed as an integral part of the peak detection step. The presence of multiple peaks that can be traced back to a single metabolite

Write more about related peaks stuff.

2.4.2 Peak Alignment

The next step in the LC-MS data processing pipeline is the peak alignment step, where peaks from different LC-MS runs have to be matched. 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. The organism studied are usually exposed to different factors (e.g. treatment or no treatment) controlled throughout the course of the experiment. Biological replicates are necessary to determine entities that are differentially expressed across samples. In contrast, technical replicates are obtained from the same samples analysed multiple times. Technical replicates are necessary to account for variability and measurement errors throughout the experiment. Since experiments in biology usually involve a comparison of multiple samples, it is necessary to align the LC-MS data produced from multiple samples in order to compare them. Alignment methods attempt to match peaks in correspondence across replicates.

An initial approach towards alignment of replicates would be to spike a known amount of internal standards into each sample before running them through the LC-MS instruments. The peaks generated from the standards can be used as 'landmark' peaks to linearly shift the retention time in each sample, usually against a reference sample. Alternatively, labelling experiment can also be done by chemically labelling metabolites in two samples with isotopic reagents. The samples are then mixed before the LC-MS experiment and run through a single LC-MS run. The same metabolites from two samples would generally appear at close

retention time, making alignment easy. 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 LC-MS experiments 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 caused by the poor reproducibility of retention time, with potentially large non-linear shifts and distortions across LC-MS runs produced from different analytical platforms. Consequently, most alignment methods correct for those shifts and distortions by finding – either explicitly or implicitly – a mapping function f that maps time t in one replicate to $f(t)$ in another. The mapping function f should be a monotonically smooth and increasing function, since elution orders of peaks that come out from the liquid chromatography instrument are generally preserved across replicates, at least for the data produced from the same LC-MS instruments. Alignment methods can therefore be broadly divided into two categories: warping and direct matching methods [16]. Warping methods perform RT correction of peak features before establishing their correspondences across replicates. Warping methods attempt to correct the RT drifts present across runs, by fitting an RT correction function (typically a regression model), using either the full LC-MS profile data or the peak feature data alone. Early warping approaches, such as dynamic time warping [17], correlation optimised warping [18] and parametric time warping [19], are predominantly based on dynamic programming, and use only the time information present in the Total Ion Chromatogram, although more recent warping approaches have started to consider the m/z dimension as well [20]. Once the time warping resulting in RT shifts have been corrected, the correspondence of peaks can be found through any method that matches peak features across runs.

The alternative approach towards alignment is the direct-matching methods, where the warping step is skipped and peak features are directly matched across replicates to establish their correspondences. Direct approaches therefore require that the peak (i.e. feature) extraction step has already been completed. Direct matching methods can be preferred in certain cases due to their simplicity, while still offering good performance [21]. The majority of direct matching approaches consist of two stages: computing feature similarity and using this similarity to match the features. 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 [22], cosine similarity [23], Euclidean distance [24], and Mahalanobis distance [25]. Once similarity has been computed, feature matching can be established through either a greedy or combinatorial matching method.

Many approaches have been proposed for direct matching of peak features. 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 [26] merges all runs and greedily groups features into aligned peaksets within a user-defined RT tolerance. Join Aligner [22] in MZmine2 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 [24] 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) [25] 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. [27] explores the application of the classical Hungarian algorithm to find the maximum weighted bipartite matching. BIPACE [23] establishes correspondence by finding the maximal cliques in the graph. SMFM [28] uses dynamic programming to compute a maximum bipartite matching under a relaxed bijective mapping assumption for time mapping.

Alignment 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 [22]. 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, described in [25]) – or by pooling features across runs and grouping similar peaks in the combined input simultaneously (e.g. the *group()* function of XCMS in [**?**]).

Label-free experiments pose many challenges in analysing replicates from different LC-MS runs. In particular, peaks from different runs can experience a potentially non-linear shift in retention time across chromatograms [29]. 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 [20]) 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, a single peak from one replicate can have several potential matches peaks in another replicate, whilst having no matches in another replicate. This is exacerbated by the uncertainties introduced due to parameter selections in the preceding steps of the pipeline. As a result, replicates produced by different LC-MS platforms or from different laboratories cannot be easily aligned to each other. In particular, the non-linear variation in retention time makes aligning technical replicates (which contains the same composition of metabolites) difficult and aligning biological replicates (which may not contain the same composition of metabolites) even more challenging.

Since large-scale untargeted metabolomics study can generate a huge number of samples (see [30, 31]), having a reliable and accurate peak alignment step during data preprocessing 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 [32]. Errors or uncertainties inadvertently produced in any sub-step before identification would be carried forward forward in the pipeline. Improper preprocessing steps can also introduce variabilities that obscure important biological variations of metabolites themselves.

2.4.3 Peak Identification

The problem of identification of LC-MS data from peptides is referred to as peptide mass fingerprinting. 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. Identification is more difficult for metabolites due to the inherent complexity in metabolomics samples. A complete characterisation of the entire metabolome of any species is very difficult, while identifications of the metabolites present in the sample are challenging, particularly in untargeted studies, due to the large number of unknown metabolites present in the sample and the numerous ways the building blocks of metabolites (the CHNOPS atoms), can be arranged in a single molecule alone. Similar to proteomics, the primary metabolite identification techniques relies on matching the accurate mass information of compounds to the set of chemical alternatives in a mass spectral database. The goals of the identification process are to distinguish between (in increasing levels of difficulty): **(1)** metabolites with different nominal masses, **(2)** metabolites with the same nominal masses, but different formula and monoisotopic masses, and finally **(3)** metabolites with the same nominal and monoisotopic masses, but different chemical structures (including chirals and isomers, such as leucine and isoleucine) [33].

Having a high mass accuracy is crucial here as it reduces the size of possible alternatives. However, 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 [34]. Identification is particularly difficult for metabolites present in low abundance in the samples. Consequently, widely-used metabolomics analysis tools like mzMine [22] employ sophisticated heuristics (such as the Seven Golden Rules) to narrow the formulae space based on various chemical constraints. Additional information such as the isotope patterns of com-

pounds, and their fragmentation patterns (obtained from tandem MS), can also be used to help in accurate metabolite identification. Identification can also be performed on the basis of groups of peaks that have been gathered together in the ionisation product clustering step. For instance, the software tool CAMERA (Collection of Algorithms for MEtabolite pRole Annotation, [35]), can be used to perform the annotations of ionisation product species on groups of peaks, based on constructing a similarity graph and detecting highly-connected subgraphs in the graph. The same principle is used in the more recent probabilistic approach of MetAssign [36] that performs identifications of metabolites based on how well observed peaks fit to the relationship between theoretical distributions of adduct and isotopes.

A fragmentation spectrum produced from tandem MS can also be used for identification through matching against **(1)** a database of reference spectra obtained from either prior measurements on a similar platform or **(2)** a database of theoretical spectra generated in an *in-silico* manner [37]. Frequently, a combination of a database of reference spectra and *in-silico* theoretical spectra is used to ensure the largest coverage of compounds during matching. The actual matching results can then be 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. In recent years, a growing number of databases for metabolomic-related data and mass spectral libraries have been available online, including for instance, the METLIN database [38], ChemSpider [39] and MassBank [40]. The size of such specific tandem spectra libraries is limited in comparison to general chemistry databases, such as PubChem [41], which may contain up to 30 million compounds. However, mass spectral libraries are not comprehensive and contain only a small number of known metabolites. For metabolites not in the libraries, an internal standard is still required [33]. Another approach towards assigning compound identity for MS2 spectrum that do not involve matching against a reference database is to predict the substructure and compound classes of the compounds in the data [37]. Classification method, such as support vector machine, decision tree and neural networks, have been applied to the problem of predicting compound identity from some training MS2 spectra [42, 43, **?**]. Such classification-based approaches often do not generalise well to new dataset generated from different analytical platform.

2.4.4 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 ([44, 45]) or in relation to other -omics studies ([46, 47]).

While targeted metabolomics focuses on a handful of specific metabolites, untargeted studies (such as in [30] and [31]) 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 [48]. 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 [49], drug discovery [50] and even in an integrative approach that combines genomics and metabolomics to obtain a more comprehensive picture of living organisms [47].

2.5 Conclusion

Software toolkit that deals with metabolomics data usually operate in a modular manner, where successive transformation of the raw LC-MS data happen by particular modules in the pipeline. However, it is important to highlight that despite the (apparently) serial preprocessing manner shown in 2.4, the actual workflow employed by life scientists is often iterative. For example, it is often the case that there are some peaks of low intensities that should be present but are found to be missing from a replicate. This requires the life scientist to go back to the peak detection stage, reduce the threshold used for noise filtering and

repeat the pre-processing stages again from that point onwards. Another challenge common in bioinformatics data analysis in general is the lack of interoperability of different toolkits that deal with different parts of the pipeline. This often requires the user to 'hack' together an ad-hoc solution to perform data preprocessing that suits the needs of the research purpose. However, despite its many challenges, metabolomics is an exciting field with many open research problems.

Chapter 3

Machine Learning Background

Note:*[Machine learning stuff, around 10 pages ..?]*

3.1 Probabilities

Random variable

Marginalisation

Inference

3.2 Mixture model clustering

In the clustering problem, we are presented with a list of feature vectors as input, and our goal is to separate those data points (features) into groups. Clustering is an instance of unsupervised learning where the learning algorithm tries to find hidden structure in unlabeled data – in constrast to supervised learning, where each data point comes with a class label. Many clustering algorithms exist, the simplest of which is k-means clustering. In k-means, we assume that the data contains a fixed set of K clusters. Features are then assigned to the nearest cluster centroids based on some distance function. Each cluster centroid is updated by computing the average of all the features assigned to it. This process is repeated until convergence.

An alternative way to cluster data is through statistical model-based clustering. In mixture model clustering, each cluster is represented by a statistical distribution. The normal (Gaussian) distribution is commonly used to model continuous data, while the multinomial distribution is frequently used to model discrete data (for e.g. as topics in a document). The entire dataset can therefore be modeled by a finite mixture of mixture of several probability

distributions, e.g. a Gaussian mixture model of two components can be used to model the distribution of heights in males and females from the sampled data. To illustrate with an example, here we construct a one-dimensional Gaussian mixture model on the retention time (RT) of our peak features. Each mixture component ideally corresponds to a metabolite, since peaks that share close RT values should originate from the same metabolite. This can be represented as the weighted finite sum of its K component distributions

$$p(\mathbf{y}|\boldsymbol{\mu}, \boldsymbol{s}, \boldsymbol{\pi}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mu_k, s) \tag{3.1}$$

where $\boldsymbol{\mu} = \{\mu_1, ..., \mu_k\}$ are the component means, s is a fixed variance common to all components, and $\boldsymbol{\pi} = \{\pi_1, ..., \pi_k\}$ are the mixing proportions where $\pi_k > 0$ and $\sum_{k=1}^{K} \pi_k = 1$. The data points are represented as $\boldsymbol{y} = \{y_1, y_2, ..., y_n\}$ where y_n is the RT value of a peak feature. In this model, each retention time value is 'generated' by its k-th component Gaussian (a crucial modelling assumption here is all peaks are generated by a metabolite, which might not be true in the presence of noisy signals, ionisation products and other artefacts in the data). We denote this by the indicator variable z_{nk} where $z_{nk} = 1$ if feature n is assigned to component k, and 0 otherwise. Collectively, the indicator variable z_{nk} for all $n \in N$ and $k \in K$ can be stored inside the matrix $\mathbf{Z}$ of size N by K. Let $\theta = \{\boldsymbol{\mu}, \boldsymbol{\pi}, \mathbf{Z}\}$ denotes all the parameters of interest in the model. We now have $p(\boldsymbol{y}|\lambda)$ and we would like to infer $p(\lambda|\boldsymbol{y})$, in particular all the indicator variables in $\mathbf{Z}$ that tells us which data point goes into which mixture component (cluster), i.e. the cluster memberships. We get this by applying Bayes' rule:

$$p(\lambda|\boldsymbol{y}) = \frac{p(\lambda)p(\boldsymbol{y}|\lambda)}{\int p(\lambda)p(\boldsymbol{y}|\lambda)d\lambda} \tag{3.2}$$

The aim of inference here is to estimate model parameters from the posterior joint distribution $p(\lambda|\boldsymbol{y})$ of model parameters λ given the data $\boldsymbol{y}$. For non-trivial models, this often involves solving the complex integration on the denumerator on the right hand side of the equation above, which can be difficult (it's not analytically tractable). Instead, parameter estimations can be done through maximum likelihood estimation (MLE), usually through the Expectation-Maximization algorithm, which finds model parameters maximising the likelihood of the model given the data. Alternatively, parameter estimations can also be done through Markov chain Monte Carlo (MCMC) methods. MCMC sampling allows us to approximate a target distribution via random walks obeying the Markovian property, where the current state in the random walk depends only on the previous state. When direct sampling of the posterior distribution is difficult but the full conditional distribution of each model parameter (e.g. $p(\mu_k|...)$ where ... denotes every other model parameter and the data) is easier to sample from, Gibbs sampling, an instance of MCMC methods, is often used.

3.3 Markov chain Monte Carlo methods

In Gibbs sampling, each model parameter is sampled in turn from its full conditional distribution until the random walk converges to the target distribution. The advantage of MCMC methods is that we obtain distributions over the model parameters, which allow us to quantify our uncertainties on them – as opposed to MLE method that provides only the most likely parameter values.

3.4 Dirichlet Process mixture model clustering

We can avoid specifying the number of cluster K *a-priori* by assuming that the data is generated by a mixture of infinite number of components (taking the limit as K goes to ∞). Dirichlet Process is a stochastic process that describes a distribution of probabilty distributions, and is often used in Bayesian non-parameteric models – particularly as a prior distribution in Dirichlet Process (DP) mixture model. In non-parametric models, the model structure (e.g. the number of mixture components) is not fixed in advance *a priori*, but is instead determined based on the observed data. To do this, we place a Dirichlet process (DP) prior on the component parameters. Let θ_k denotes the component parameter of the k-th cluster. The DP can be viewed as an infinite dimensional generalisation of the Dirichlet distribution, where draws from the DP is itself a probability distribution. Following the example above, the RT data points can thus be explained by the following generative model:

$$G|\alpha, H \sim DP(\alpha, H) \tag{3.3}$$

$$\mu_k|G \sim G \tag{3.4}$$

$$y_n|\mu_k \sim N(\mu_k, s) \tag{3.5}$$

Similar to the finite case, $N(\mu_k, s)$ denotes the distribution of the data point y_n,which is a Gaussian distribution parameterised by mean μ_k and variance s (which is fixed, so we will not infer). The component parameter (the μ_ks) are conditionally independent given G, which is a discrete distribution drawn from the Dirichlet Process, and the data point y_n are conditionally independent given a component parameter μ_k. H is the base distribution that provides the prior on the μ_ks, while the parameter α can be seen as the inverse variance, with larger values of α producing smaller variance in the distributions drawn from the GP from H. The DP prior induces a partitioning on the data points, where the probability of a newly arriving data point to join an existing cluster is proportional to the number of data points already in that cluster. However, with a probability proportional to α, the data point will form a new cluster on its own. Additional details on Dirichlet process mixture model clustering can be found in [51].

3.5 Hierarchical Dirichlet Process mixture model clustering

While the DP mixture model allows us to cluster related peaks together within each run, we would also like such clusterings to be shared across runs. This is reasonable to expect because if a cluster represents related peaks derived from a metabolite / compound, then we can expect to discover the same clusters across similar runs. The idea here is that: (1) peaks put together in the same 'global' clusters are basically aligned, and (2) we can define a model with entities that are meaningful in the biological sense, and thus discover insights from such models.

Suppose we have J runs to align. A Hierarchical Dirichlet process (HDP) is a distribution over a set of random probability measures, where each replicate has its associated random probability measure G_j. The global measure G_0 is distributed as a Dirichlet process, and the random measures G_j for each replicate is also distributed according to a DP, conditionally independent on G_0

$$G_0|\alpha, H \sim DP(\alpha, H) \tag{3.6}$$

$$G_j|\alpha_0, G_0 \sim DP(\alpha_0, G_0) \tag{3.7}$$

$$\mu_k|G_j \sim G_j \tag{3.8}$$

$$y_n|\mu_k \sim N(\mu_k, s) \tag{3.9}$$

Notice that the difference between DP mixture and HDP mixture is the fact that we are adding another level of hierarchy to the model, where the probability measure G_j for each replicate j is in turn drawn from the global measure G_0. A draw from the DP is a discrete probability measure, so G_0 and G_j are discrete distributions of point masses from the base distribution H. By drawing G_j, the j-th file specific measure, from a common global measure G_0, this makes it possible for us to share clustering parameters across different runs. In the popular Chinese Restaurant Franchise analogy described in [**?**], we have a Chinese restaurant franchise with a menu of dishes shared across all its restaurants (the global measure G_0). At each table (replicate cluster parameter) in a restaurant, one dish (global cluster parameter) is ordered from the menu by the first customer (data point) who sits there. The dish is shared by all customers who sit on that table. Newly arriving customer joins existing tables with a probability proportional to the number of people already sitting there, or sits on a new table by himself with a probability proportional to α_0. Existing dishes are also ordered based on its popularity across the franchise (the number of tables ordering it), or a new dish is created with a probability proportional to α. In this hierarchical DP process, cluster parameter values are shared across runs and also within run.

3.6 Latent Dirichet Allocation

Latent Dirichlet Allocation (LDA), proposed in [**?**], is a probabilistic topic model widely used for unsupervised topic discovery. In the standard LDA model applied to text mining, documents comprise of some topics, each of which may produce the observed words in that document. Given a corpus of documents, the goal of inference in LDA is to approximate the posterior distributions of documents to topics and words to topics.

For the purpose of substructure discovery in MS2 data, a topic – explained as the set of recurring words shared in many documents – can be seen as corresponding to a substructure shared by many metabolites. Each topic then produces the observed MS2 fragment/loss words in an MS1 document. We assume the bag-of-word word model, where within each MS1 document, the observed MS2 fragment/loss word features are exchangeable. i.e. their ordering do not matter, only their observed counts matter. The input to LDA is therefore a matrix of the counts of occurences of MS2 word for each MS1 document. This can produced by concatenating the count matrices of the fragment words and the loss words produced in section **??** row-wise, e.g. if there are N_f unique fragment words and N_l unique loss words, both of which are shared across D MS1 peaks, the input matrix to LDA is a D-by-$(N_f + N_l)$ matrix. Entries in the matrix are the observed counts of words in the document, so they are the discretised intensity values of the fragment and loss words for each MS1 peak – produced according to Section **??**. We restrict the input to the standard LDA to take into account only the fragment and loss words because the counts of both fragment and loss words are derived from the normalised intensity values of the MS2 peaks.

The standard LDA model – as applied to substructure discovery – is now briefly described here. Given K predefined topics (indexed by $k = 1, ..., K$) corresponding to metabolite substructures, the observation of the n-th MS2 fragment/loss word in the d-th document (MS1 peak) can be described by the following generative process.

$$w_{dn}|\boldsymbol{\phi}_{z_{dn}} \sim Multinomial(\boldsymbol{\phi}_{z_{dn}}) \tag{3.10}$$

$$z_{dn}|\boldsymbol{\theta}_d \sim Multinomial(\boldsymbol{\theta}_d) \tag{3.11}$$

$$\boldsymbol{\theta}_d|\alpha \sim Dirichlet(\alpha) \tag{3.12}$$

$$\boldsymbol{\phi}_k|\beta \sim Dirichlet(\beta) \tag{3.13}$$

In other words: observation on the n-th MS2 word in the d-th MS1 peak (w_{dn}) is conditioned on the assignment of word w_{dn} to some known k-th multinomial distribution (corresponding to a substructure). This assignment is denoted by the indicator variable z_{dn}, so $z_{dn} = k$ if w_{dn} is assigned to a k-th multinomial. The k-th multinomial distribution that an MS2 word is assigned to is characterised by the parameter vector $\boldsymbol{\phi}_{z_{dn}}$. However, $\boldsymbol{\phi}_{z_{dn}}$ is itself drawn

from a prior Dirichlet distribution having a symmetric parameter β. The probability of seeing certain substructures (topics) for each d-th MS1 peak is then drawn from a multinomial distribution with a parameter vector $\boldsymbol{\theta}_d$. This parameter vector $\boldsymbol{\theta}_d$ is in turn drawn from a prior Dirichlet distribution having a symmetric parameter α. Figure 5.2 is the plate diagram of the standard LDA model, which shows the conditional dependencies between the random variables in the model.

Chapter 4

Incorporating Clustering Information into Peak Alignment

Note:*[Around 25 pages?]*

4.1 Introduction

None of the alignment tools surveyed in Section 2.4.2 [16] take into account the structural dependencies between co-eluting peaks that are related to the same metabolite when solving the correspondence problem. Such information could potentially be used to improve the alignment process, since a set of co-eluting peaks (derived from the same compound/peptide fragment) in one run should generally be aligned to another set of co-eluting peaks in the other run. As described in Section 2.4.1, related peaks are defined to be all those peaks that appear in a run due to the presence of one compound (peptide/metabolite) in the sample being analysed. Examples of related peaks are isotope peaks, multiple adduct and deduct peaks, and fragment peaks [**?**]. Such peaks should co-elute from the column and have similar chromatographic shapes and RT values. The related peak information can come from any peak grouping (e.g. clustering via RT) method, but one key assumptions is that groups of co-eluting peaks will be preserved across runs.

In this chapter, we propose using an infinite Gaussian mixture model to pull related peaks sharing similar RT values together (Section 4.2). The information from the clustering process is then used to modify the similarity score matrix used for the alignment (matching) of peaks across runs (Section 4.3). This idea is illustrated in Figure 4.1. In the Figure, 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 related 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.

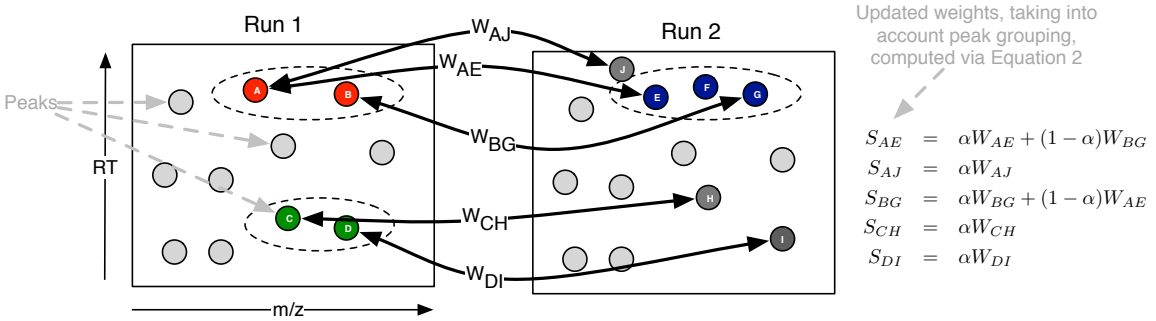

Figure 4.1: Illustrative example of the incorporation of grouping information into the similarity score. Each node in the figure is a peak feature, and dotted ovals represent groups of related peaks, e.g. isotopes, fragments, etc. Initially weights (e.g. W_{AE}) are computed for pairs of peaks (one from 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 related 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).

Statement of Original Work

The idea of constructing alignment via approximate maximum weighted matching was proposed by the author. Simon Rogers then conceived the idea of using the clustering information of related peaks to modify the similarity matrix used for matching. Code implementation, the construction of alignment ground truth and performance evaluation was carried out by the author.

4.2 Clustering of related peaks

A peak feature refers to a tuple of $(m/z, RT)$ produced as output after the pre-processing of LC-MS data, where m/z is the mass-to-charge value and RT the retention time value of a peak feature. We can group related peaks together by RT. Our observation consists of a vector of

N observed peak's RT values $\mathbf{y} = (y_1, y_2, ..., y_n)$. Our aim is to partition each set of peaks into K groups of related peaks (clusters) by their RT values. We used a Gaussian mixture model with Dirichlet Process prior [51] 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 related peaks. Let the indicator $z_{nk} = 1$ denotes the assignment of peak n to RT cluster k. Then:

$$\begin{aligned} \boldsymbol{\pi}|\alpha &\sim GEM(\gamma) && (4.1)\\ z_{nk} = 1|\boldsymbol{\pi}_k &\sim \boldsymbol{\pi}_k && (4.2)\\ \mu_k|\mu_0, \tau_0 &\sim \mathcal{N}(\mu_k|\mu_0, \tau_0^{-1}) && (4.3)\\ y_n|z_{nk} = 1, \mu_k &\sim \mathcal{N}(\mu_k, \delta^{-1}) && (4.4) \end{aligned}$$

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:

$$\begin{aligned} \beta_k &\sim Beta(1, \gamma) && (4.5)\\ \boldsymbol{\pi}_k &= \beta_k \prod_{l=1}^{k-1}(1 - \beta_l) && (4.6) \end{aligned}$$

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 $p(z_{nk} = 1, ...)$ 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:

$$p(z_{nk} = 1|\mathbf{y}_n, \ldots) \propto \begin{cases} c_k \cdot p(\mathbf{y}_n|z_{nk} = 1, ...) \\ \gamma \cdot p(\mathbf{y}_n|z_{nk^*} = 1, ...) \end{cases} \quad (4.7)$$

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

$$p(\mathbf{y}_n|z_{nk} = 1...) = \mathcal{N}(\mathbf{y}_n|\mu_k, \lambda_k^{-1}) \quad (4.8)$$

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:

$$p(\mathbf{y}_n|z_{nk^*}=1...) \;=\; \mathcal{N}(\mathbf{y}_n|\mu_0,\lambda_{k^*}^{-1}) \tag{4.9}$$

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. For each sample, our primary interest is the marginal posterior of the probability of peak-vs-peak to be in the same cluster k. We obtain this 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). The result of this is a matrix of probabilities for any two peaks in the same run to be in the same cluster k, averaged over all samples.

4.3 Direct Matching

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. In the following sections, we describe each step in more detail.

4.3.1 Feature Similarity

Suppose we wish to align run A containing N_A peaks with run B containing N_B peaks. We follow SIMA [25] 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. 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:

$$W(\boldsymbol{p}_i,\boldsymbol{p}_j)=1-\frac{D(\boldsymbol{p}_i,\boldsymbol{p}_j)}{D_{max}}, \tag{4.10}$$

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 Incorporating Related Peak Groups

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

$$\boldsymbol{S} = \alpha \boldsymbol{W} + (1 - \alpha)\boldsymbol{L} \tag{4.11}$$

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:

$$\boldsymbol{L} = \boldsymbol{C}^A \cdot \boldsymbol{W} \cdot \boldsymbol{C}^B. \tag{4.12}$$

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 4.12 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 related 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 demonstrate the performance improvement from incorporating related peaks information using two different clustering algorithms: a greedy RT clustering approach (Maximum-Weighted-Greedy (MWG)) and a statistical mixture model (Maximum-Weighted-Mixture (MWM)). MWG starts with the most intense peak in the

dataset and clusters it 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 $\mathbf{C}^B$. MWM uses an infinite Gaussian mixture model on RT [**?**, see e.g.]]Rasmussen2000. Analytical inference is not possible in this model, so a Gibbs sampling procedure is used to sample clusterings used to compute the probability of two features (peaks) to be in the same cluster. 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. More details of the mixture model and sampling procedure are provided in the Supplementary document.

4.3.3 Feature Matching

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 [25], the Gale-Shapley algorithm [**?**] 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 [27] 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 [**?**] 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 [**?**].

4.4 Evaluation Study

Performance evaluation of alignment methods themselves is difficult due to the lack of gold standard and evaluation criteria for benchmarking [6]. Relatively few works, such as [21], exists that provide a comprehensive ground truth for evaluation. In fact, despite the numerous alignment methods that exist, most methods remain unevaluated, evaluated against a small number of alternatives or evaluated based on highly subjective criteria [16]. In particular, 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. Precision and Recall has been used for performance evaluation of other alignment tools **[REF]**.

In this chapter, the performance of the proposed methods and other benchmark methods is evaluated on LC-MS datasets from proteomic, metabolomic and glycomic experiments. The proteomic datasets are obtained from [21] while the glycomic dataset comes from [**?**]. These datasets provide the ground truth for alignment and have used to benchmark alignment performance in other evaluation studies [21, 22, 24, 25, **?**]. Additionally, we also introduce a metabolomic dataset generated from the standard runs used for the calibration of chromatographic columns [31]. The runs were produced from different LC-MS analyses separated by weeks, representing a challenging alignment scenario.

4.4.1 Construction of Ground Truth

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, 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 section for each dataset.

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 P1 and P2 datasets

4.4.2 Proteomic Datasets

[21] introduces two benchmark LC-MS proteomic sets (P1, P2) constructed to evaluate the ability of alignment tools in dealing with large retention time drifts, available from http://msbi.ipb-halle.de/msbi/caap and our site. 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 [21] 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 and **??** show 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 [21].

4.4.3 Metabolomic Datasets

We use a metabolomic dataset generated from a mixture of 104 standard metabolites used for the calibration of chromatographic columns (details in [31]). 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, which were then converted into a suitable format using the mzMatch suite [**?**]. Both the original PeakML files and the converted text files can be found in our site.

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 **??**). 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.

Standard Run	# features	Standard 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

The full metabolomic dataset comes in 11 runs in total. 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. The following tables show the number of features in each run and the pairs of files selected as training and testing sets in our Metabolomic experiment.

Filename	# features	Filename	# features
std1-file1.txt	4999	std1-file7.txt	6319
std1-file2.txt	4986	std1-file8.txt	4101
std1-file3.txt	6836	std1-file9.txt	5485
std1-file4.txt	9752	std1-file10.txt	5034
std1-file5.txt	7076	std1-file11.txt	5317
std1-file6.txt	4146		

Table 4.3: No. of features in the full metabolomic dataset

4.4.4 Glycomic Dataset

[**?**] provides a glycomic dataset containing 23 runs, available from http://omics.georgetown.edu/alignL and our site. The glyomic dataset were produced from untargeted LC-MS study for identifying N-glycan disease biomarkers. 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 [**?**] 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 [**?**], 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 [**?**]

4.4.5 Experimental setup

The alignment tools evaluated have in common user-defined mass-to-charge ratio (m/z) and 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 [16]. 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 [16], and there is no reason to believe that these parameters should remain constant across different experiments.

Previous studies on the proteomic and metabolomic datasets presented here [21, 24, 25] 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 paper, *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 [16] by independently reproducing key results from [21] and [25] for the Join and SIMA alignment methods. Our evaluation studies were performed on datasets selected in section 5.3.1 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 ($\frac{\boldsymbol{TP}}{\boldsymbol{TP}+\boldsymbol{FP}}$) is the fraction of aligned pairs in the output that are correct with respect to the ground truth, while recall ($\frac{\boldsymbol{TP}}{\boldsymbol{TP}+\boldsymbol{FN}}$) 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 [27]. Collectively for the purpose of performance evaluation, the set of Precision, Recall and F_1 values is referred to as a 'measurement'.

4.4.6 Other Alignment Tools For Comparison

Our proposed approach was benchmarked against MZmine2's Join Aligner [22] and SIMA [25]. These 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.

MZmine2's Join Aligner

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.

Simultaneous Multiple Alignment (SIMA)

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.

4.5 Results

We conducted several experiments on the proteomic, metabolomic and glycomic datasets, each designed to test a different aspect of alignment tools' performance. Details on the parameter optimisations for evaluated tools are provided in the Supplementary document.

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. 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 the Supplementary document).

The results are shown in Tables 4.5 and 4.6 (full results, including precision and recall values, can be found in the Supplementary document). 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. Figure **??** shows 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. The full results for all fractions, including computed precision and recall values, are available in the Supplementary document.

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.

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 (full results in the Supplementary document). 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. The implication is discussed in section 5.5.

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 4.2 and 4.3 summarise the results from the experiments (detailed full results and parameter range selections are described in the Supplementary document). 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. 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. Note that unlike in the Proteomic and Glycomic experiments, the results for MWG shown here (also referred to as MWG(RT+PS) in section 3.4 of the Supplementary document) takes into account the Pearson correlations of the chromatographic shapes between peak features during the clustering process. Results for MWG that consider only the RT values (referred to as MWG(RT) in the Supplementary) for grouping of related peaks can be found in section 3.4 of the Supplementary document.

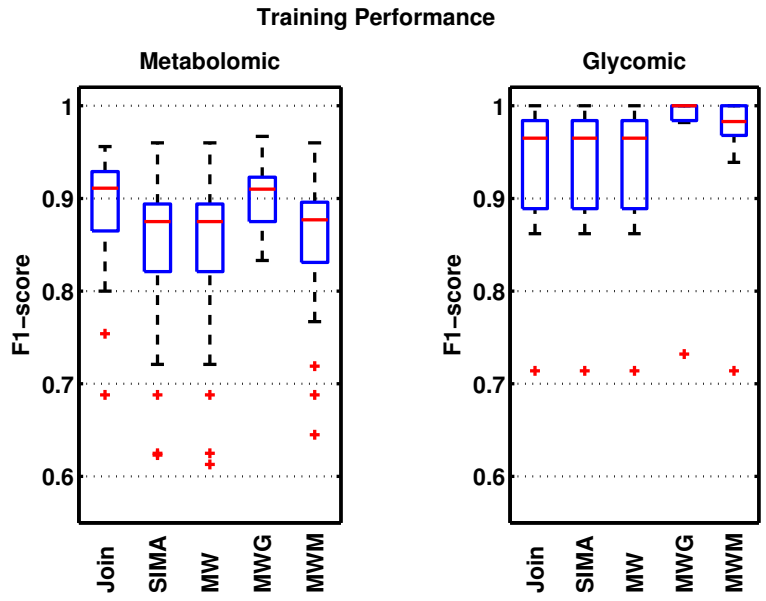

Figure 4.2: Training performance shows the best F_1 scores obtained by each method on 30 pairs of randomly-selected metabolomic and glycomic training sets.

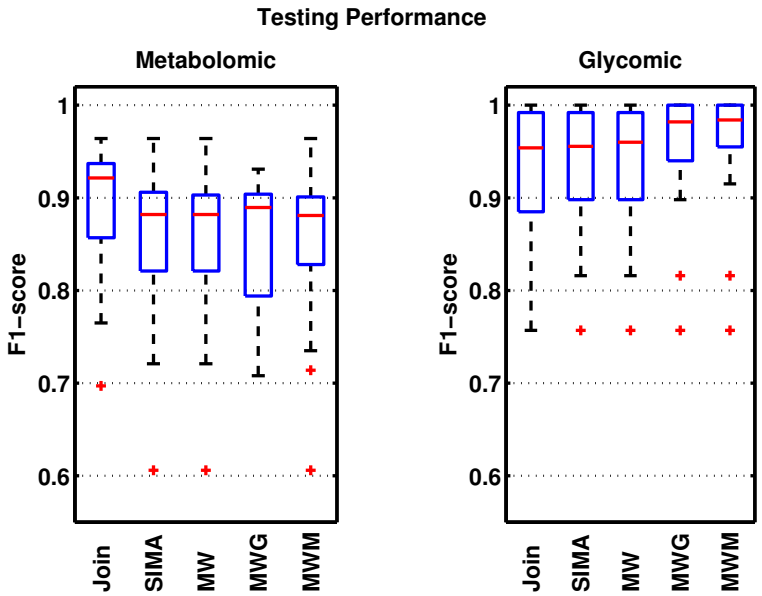

Figure 4.3: Testing performance shows how well each method generalise on the 30 different testing sets, each evaluated using the optimal training parameters from its corresponding training set.

4.6 Discussion and Conclusion

In this paper, 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 category of *direct matching* approaches – those alignment approaches 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 related peaks. In the case of the noisiest data (dataset P2 fraction 000), we observe some combinations of parameters that result in training points with reduced precision and recall values. These are likely due to the difficulty of producing a high-quality grouping of related peaks with sub-optimal parameters especially when only the RT information is used. Comparisons of matching performances on the metabolomic dataset for the clustering of related peaks with and without chromatographic peak shape correlations (see section 3.4 of the Supplementary document) shows that for best performance during the clustering stage, additional information, such as chromatographic peak shapes, should be used whenever available.

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. 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. Using MW as an example, the optimal RT window parameter σ_t is 90 seconds for training fraction 000 and 275 seconds for training fraction 080. We also observe that in the multiple-fractions experiment for P2, 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 fractions of P2. The results suggest the dependence of our methods on the quality of groupings of related peaks in order to generalise well on different runs. The same conclusion can be obtained from the training and testing performances on the metabolomic dataset as well, where we see significant improvements in the training performance but none in the testing performance. On datasets with lower RT variation, such as the P1 and the glycomic data, we see evidence of improvements in both the training and testing performances, suggesting that incorporating clustering information in the proposed manner can indeed improve alignment performance and generalise well to different runs even with less than optimal parameter settings.

Note that our method relies on grouping of related 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 related 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 related 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 [**?**] that uses a mixture model of peak shape correlations or [36] that considers the dependencies between adduct and isotopic peaks when clustering. Exploring alternative approximate matching algorithms (such as the scaling algorithm in [**?**], 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 paper also suffer from

variable behaviours depending on the order of the runs being aligned. 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 paper and is an item for future work.

The related-peak based similarity score that underpins our approach could be applied to many other direct matching approaches [**?**, e.g. SIMA:]]Voss2011a and similar ideas could also be incorporated into recently developed methods that take into account the presence of internal standards [**?**]. The evaluation pipeline developed over the course of our experiments can also be easily extended by algorithmic researchers to evaluate other alignment tools in future work.

Chapter 5

Providing Confidence Values in Alignment Results

Note:*[Around 20 pages?]*

5.1 Introduction

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. A preliminary form of this idea has been explored in [**?**], where hierarchical clustering is performed on the total ion chromatogram data to group peaks into within-run local clusters, which are further grouped into across-run super clusters. The highly accurate mass information available from modern LC-MS platforms is not used in [**?**], although it is highlighted as a possible future work. The choice of using a hierarchical clustering method in [**?**] also requires 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.

According to [16], the common shortcomings shared by many alignment methods include the incorrect modelling assumption that elution order of peaks is preserved across runs and the abundance of user-defined parameters, which can dramatically influence alignment results. Further uncertainties can be introduced due to the selection of a reference run and the construction of a guide tree in hierarchical methods. Since alignment is such an important part of the data preprocessing steps, it would be useful to be able to robustly identify the uncertainty or confidence in the alignment results. The subject of identifying and quantifying

uncertainty has been extensively investigated in the problem of multiple sequence alignment (MSA) for genomics and transcriptomics. [**?**] attempt to quantify the alignment uncertainty of the popular MSA tool ClustalW [**?**], based on evaluations using synthetic data, and concludes that between half to all columns in their benchmark MSA results contain alignment errors. [**?**] construct a score that reflects the consensus between all possible pairwise alignments in T-COFFEE, while [**?**] 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 [**?**] and [**?**], where the MSA results and phylogeny are constructed simultaneously, thus eliminating the need for a guide tree.

Additionally, it is also desirable for alignment methods to provide some measure of confidence in the quality of its alignment. In the absence of ground truth information, life scientists typically 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 with confidence values is a big research gap that, to our knowledge, has not been addressed at all by any of the alignment tools surveyed earlier. Some interactive analysis tools like MAVEN [**?**] can assign quality scores to individual peaks. This is accomplished by training a neural network (or a decision tree) on training data that have been manually annotated using metrics of peak quality. Other approach like [**?**] 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.

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. Bayesian methods operating on profile data [**?**, e.g.]]Listgarten2004, Kong2009, Tsai2013a and feature-based alignment methods [**?**, e.g.]]Fischer2006, Pluskal2010, Voss2011a exist to correct RT drift, but in such methods, uncertainties are not propagated from the RT regression stage to the necessary peak matching stage that follows. 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, [**?**] propose an empirical Bayes 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 [**?**] still requires a greedy search to find candidate features within m/z and RT-RT tolerances to a predetermined set of 'landmark' peaks. [**?**] describe PeakLink, a workflow for alignment that performs an initial warping using a fourth-degree polynomial. PeakLink poses the pairwise matching problem as a bi-

nary 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 [**?**] state that it would be possible to do so with the choice of a suitable reference run.

In this work, we expand upon the idea of viewing direct matching as a hierarchical clustering problem by proposing **HDP-Align**, a Bayesian non-parametric model that groups related-peaks within runs 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, which represent the various ionisation products (IPs) derived from the global compound. The proposed model allows us to infer the matching of peaks across all runs at once, without the need for any intermediate merging of pairwise runs, and the resulting posterior summaries provide us with a confidence score in the matching quality of aligned peaksets. This 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 5.1 shows an illustration of the clustering process in HDP-Align. Additionally, the latent variables of clustering structure inferred in the model can potentially have physically meaningful identities that can be used for further analysis, and using a metabolomic dataset, we demonstrate the usefulness of such clustering objects by using the mass clusters derived from the model to perform putative annotations of features based on their potential adduct types and metabolite identities.

5.2 Hierarchical Dirichlet Process Mixture Model for Alignment

5.2.1 Model Description

The proposed model for HDP-Align is framed as a Hierarchical Dirichlet Process (HDP) mixture model [**?**], with essential modifications to suit the nature of the multiple peak alignment problem. 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 peak features in total, which can be separated into K_j local clusters of related-peak features. In a j-th file, peak features are indexed by $n = 1, ..., N_j$ and local clusters are indexed by $k = 1, ..., K_j$. Across

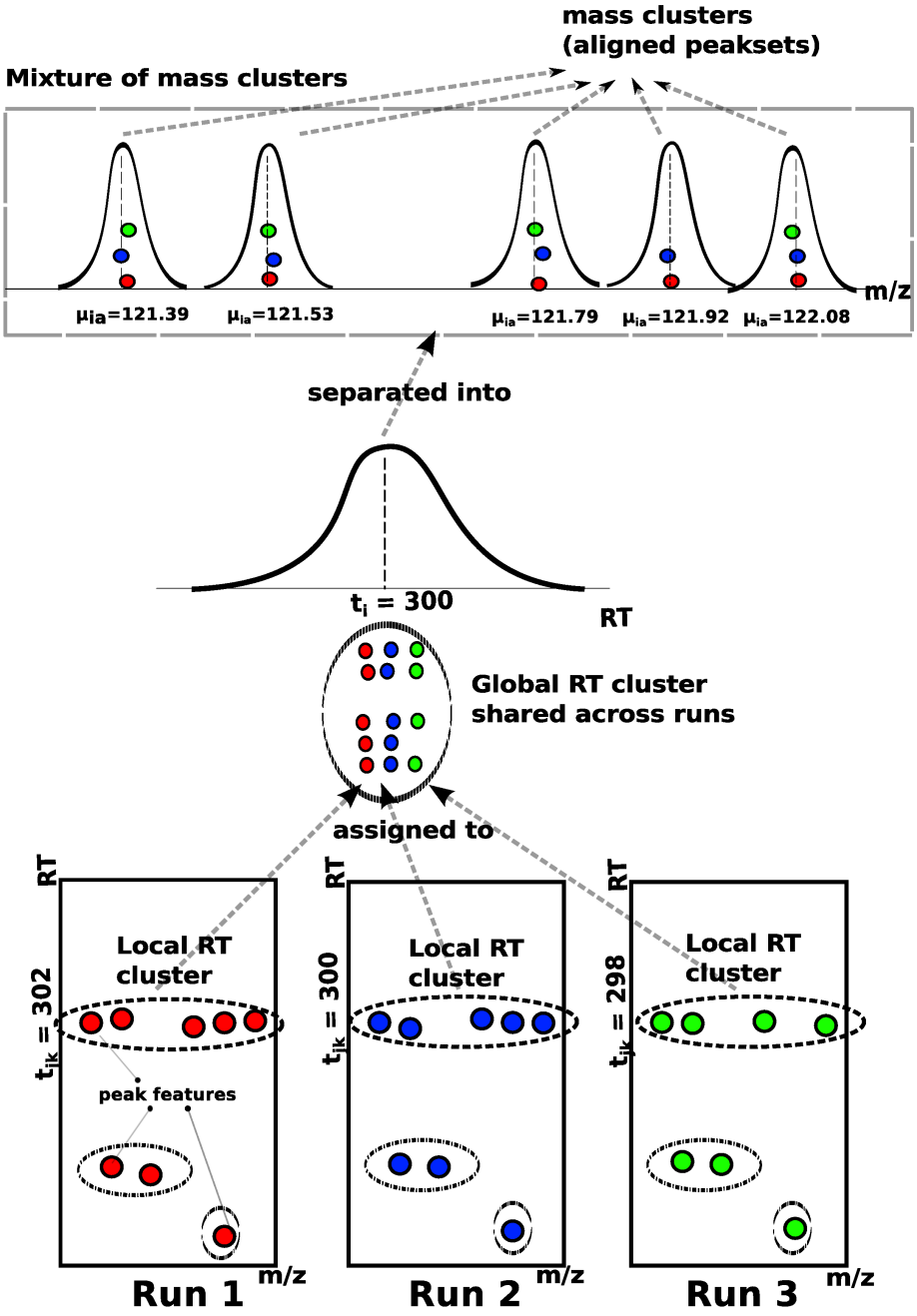

Figure 5.1: An illustrative example of how the proposed model in HDP-Align simultaneously **(1)** performs the clustering of related peak features into within-run local clusters by their RT values, **(2)** assigns the peak features to global RT clusters shared across runs, and **(3)** separates peak features into mass clusters, which correspond to aligned peaksets.

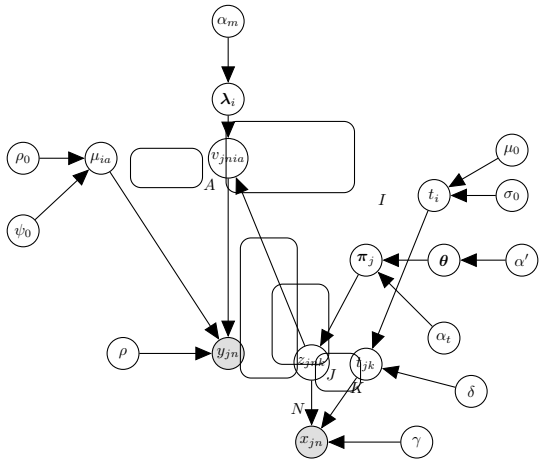

Figure 5.2: Graphical model for HDP-Align. x_{jn} is the observed RT value of peak n in file j, while y_{jn} is the observed m/z value.

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 peak features by their m/z values into A mass clusters (indexed by $a = 1, ..., A$) corresponding to the ions produced by different adduct-isotope combinations from the global compound during the MS process. We call these the ionisation products (IPs).

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. Figure 5.2 shows the conditional dependencies between random variables in the model as a Bayesian network. Let d_j be the list of observed data of peak features 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 m/z value in log space. $\boldsymbol{\theta}$ denotes the global mixing proportions and $\boldsymbol{\pi}_j$ the local mixing proportions for file j. The global mixing proportions $\boldsymbol{\theta}$ are distributed according to the Griffiths, Engen and McCloskey (GEM) distribution:

$$\boldsymbol{\theta}|\alpha' \sim GEM(\alpha') \tag{5.1}$$

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

$$\beta_i \sim Beta(1, \alpha') \tag{5.2}$$

$$\theta_i = \beta_i \prod_{l=1}^{i-1}(1 - \beta_l) \tag{5.3}$$

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

$$\boldsymbol{\pi}_j | \alpha_t, \boldsymbol{\theta} \sim DP(\alpha_t, \boldsymbol{\theta}) \tag{5.4}$$

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.

$$z_{jnk} = 1 | \boldsymbol{\pi}_j \sim \boldsymbol{\pi}_j \tag{5.5}$$

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.

$$t_i | \mu_0, \sigma_0 \sim \mathcal{N}(\mu_0, \sigma_0^{-1}) \tag{5.6}$$

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.

$$t_{jk} | t_i, \delta \sim \mathcal{N}(t_i, \delta^{-1}) \tag{5.7}$$

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.

$$x_{jn} | z_{jnk} = 1, t_{jk}, \gamma \sim \mathcal{N}(t_{jk}, \gamma^{-1}) \tag{5.8}$$

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:

$$\begin{aligned}
\boldsymbol{\lambda}_i|\alpha_m &\sim GEM(\alpha_m) && (5.9)\\
v_{jnia}=1|\boldsymbol{\lambda}_i &\sim \boldsymbol{\lambda}_i && (5.10)\\
\mu_{ia}|\psi_0,\rho_0 &\sim \mathcal{N}(\mu_{ia}|\psi_0,\rho_0^{-1}) && (5.11)\\
y_{jn}|v_{jnia}=1,\mu_{ia} &\sim \mathcal{N}(\mu_{ia},\rho^{-1})\cdot I(\mathbf{d}_{jn}) && (5.12)
\end{aligned}$$

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.

5.2.2 Inference

Inference within the model is performed via a Gibbs sampling scheme, allowing us to compute posterior probabilities over the alignment of any set of peaks across the J files via the proportion of posterior samples in which they are assigned to the same mass component (a) in the same top-level cluster. 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, peak features 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 peak features 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.

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 $p(z_{jnk} = 1|\mathbf{d}_{jn}, \ldots)$.

$$p(z_{jnk} = 1|\mathbf{d}_{jn}, \ldots) \propto \begin{cases} c_{jk} \cdot p(\mathbf{d}_{jn}|z_{jnk} = 1, ...) \\ \alpha_t \cdot p(\mathbf{d}_{jn}|z_{jnk^*} = 1, ...) \end{cases} \tag{5.13}$$

We will consider the top and bottom terms of eq. 5.13 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

$$p(\mathbf{d}_{jn}|z_{jnk} = 1, ...) = p(x_{jn}|z_{jnk} = 1, ...) \cdot p(y_{jn}|z_{jnk} = 1, ...) \tag{5.14}$$

The RT term $p(x_{jn}|z_{jnk} = 1, ...)$ in eq. 5.14 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 $p(y_{jn}|z_{jnk} = 1, v_{jni} = 1...)$

$$\begin{aligned} p(y_{jn}|z_{jnk} = 1, v_{jni} = 1...) = &\sum_a \frac{c_{ia}}{\alpha_m + \sum_a c_{ia}} p(y_{jn}|v_{jnia} = 1, ...) \\ &+ \frac{\alpha_m}{\alpha_m + \sum_a c_{ia}} p(y_{jn}|v_{jnia^*} = 1, ...) \end{aligned} \tag{5.15}$$

To compute the terms in eq. 5.15, first we consider the case for an existing mass cluster a in the global RT cluster i. Then,

$$p(y_{jn}|v_{jnia} = 1, ...) = \mathcal{N}(\mu_{ia}, \rho^{-1}) \tag{5.16}$$

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

$$p(y_{jn}|v_{jnia^*} = 1, ...) = \mathcal{N}(\psi_0, \rho^{-1} + \rho_0^{-1}) \tag{5.17}$$

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, we get

$$p(\mathbf{d}_{jn}|z_{jnk^*}=1,...) = \sum_i \left[\frac{c_i}{\alpha' + \sum_i c_i} p(\mathbf{d}_{jn}|v_{jni}=1,...)\right] + \frac{\alpha'}{\alpha' + \sum_i c_i} p(\mathbf{d}_{jn}|v_{jni^*}=1,...) \tag{5.18}$$

There are two terms to compute in eq. 5.18: 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. 5.18, $p(\mathbf{d}_{jn}|v_{jni}=1,...)$ 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

$$p(x_{jn}|v_{jni}=1,...) = \mathcal{N}(x_{jn}|t_i, \gamma^{-1}+\delta^{-1}) \tag{5.19}$$

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. 5.15). Finally, for a new global RT cluster i^* in eq. 5.18, $p(\mathbf{d}_{jn}|v_{jni^*}=1,...)$ is also assumed to factorise into its RT and mass terms. Then, we marginalise over t_{jk} and t_i to obtain

$$p(x_{jn}|v_{jni^*}=1,...) = \mathcal{N}(x_{jn}|\mu_0, \sigma_0^{-1}+\gamma^{-1}+\delta^{-1}) \tag{5.20}$$

and marginalise over μ_{ia} to get

$$p(y_{jn}|v_{jni^*}=1,...) = \mathcal{N}(y_{jn}|\psi_0, \rho^{-1}+\rho_0^{-1}) \tag{5.21}$$

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

$$p(t_i|\ldots) \propto p(t_i|\mu_0, \sigma_0^{-1}) \prod_j^J \prod_k^K p(t_{jk\in i}|t_i, \delta) = \mathcal{N}(\mu_i, \gamma_i^{-1}) \tag{5.22}$$

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.

$$p(t_{jk}|\ldots) \propto p(t_{jk}|t_i, \delta^{-1}) \prod_j^J \prod_n^N p(x_{jn\in k}|t_{jk}, \gamma) = \mathcal{N}(\mu_k, \gamma_k^{-1}) \quad (5.23)$$

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

5.2.3 Using the Inference Results

Feature Matching

The Gibbs sampling procedure produces a collection of samples from the posterior distribution over all parameters of the HDP-Align model. We can use these samples to compute various posterior summaries and more interestingly, extract the alignment of peaks from the inference results (since features assigned into the same mass cluster with the same global RT cluster are considered to be aligned). For each sample from the posterior distribution, we record the aligned peaksets of peak features put into the same mass cluster. Averaging over all samples provides a distribution over these aligned peaksets.

Note that across the returned aligned peaksets, 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 peak feature combinations are included in the output from the model only when they occur with matching probability $>t$. Varying this threshold allows user 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 other alignment methods and can potentially be useful in problem domains where high precision is required from the alignment results.

Isotopic Product and Metabolite Identity Annotations

In metabolomic studies using electrospray ionisation, a single metabolite can generate multiple ionisation products (IPs, such as isotopic variants, adducts, fragment peaks), alongside other peaks resulting from noise and artifacts introduced during mass spectrometry [**?**]. Determining and annotating these IP peaks are desirable to remove extraneous peaks and reduce the burden of subsequent downstream analysis. Additionally, deducing the precursor molecular masses that generate the IPs is often essential in order to query compound library databases before assigning putative metabolite identities.

Table 5.1: List of common adduct types in positive ionisation mode for ESI.

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			

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 IPs. To demonstrate the possibility of obtaining additional information beyond alignment from the output of HDP-Align, we follow the workflow in [**?**] that performs IPs and metabolite annotations of peak features. 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 [**?**] lies in the fact that HDP-Align is able to perform the two separate steps of peak alignment and IP clustering simultaneously, as shown in Figure 5.3.

To perform IP annotation on the metabolomic dataset used in our experiment, we take the set of clustering objects produced in a single 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. A list of common adducts (Table 5.1) in positive ionisation mode is used to compute the inverse transformation $t^{-1}(m, d, e, u) = ((e * m) - d)/u$ for a precursor mass c that generates m. Here, d is the adduct mass, e is the charge and u the number of metabolite molecules in the IP type. Following [**?**], any two mass clusters sharing the same precursor mass c (within tolerance) provide a vote on the presence of that consensus precursor mass. The respective pair of mass clusters and features within can then be annotated with the adduct type that produces the transformation t^{-1} to the shared precursor mass c. The set of precursor masses deduced in this manner can also be used to query a local KEGG database in order to assign putative identities to global compounds.

5.3 Evaluation Study

5.3.1 Evaluation Datasets

Performance of the proposed methods and other benchmark methods is evaluated on LC-MS datasets from proteomic, glycomic and metabolomic experiments (Table **??** summarises the number of features in the datasets). The Proteomic dataset is obtained from [21]. All 6 fractions from the Proteomic dataset in [21], each containing 2 runs of features having high

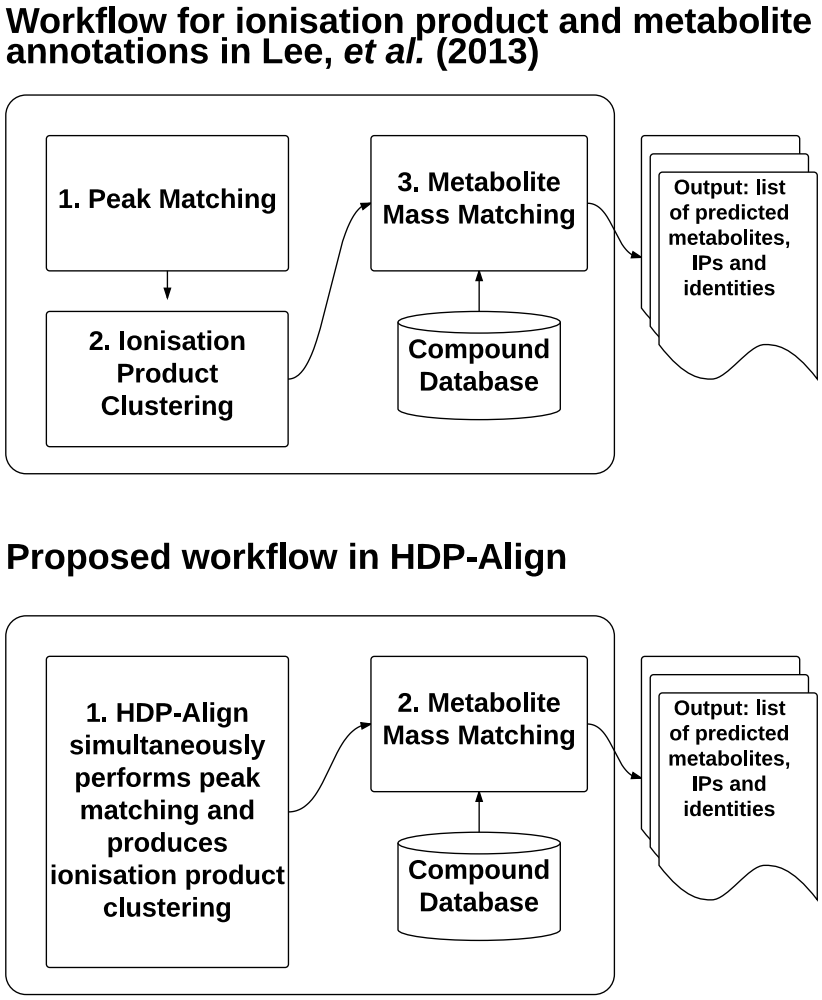

Figure 5.3: Comparisons on the workflow to assign putative annotations on isotopic products and metabolites described in [**?**] and in HDP-Align.

RT variations across runs, are used in our experiments. The Glycomic dataset is provided by [**?**]. We use the first 10 runs from the dataset in our experiment. Both of the Proteomic and Glycomic datasets provide alignment ground truth and have been used to benchmark alignment performance in other studies [21, 22, 24, 25, **?**]. Additionally, we also introduce a metabolomic dataset generated from standard runs used for the calibration of chromatographic columns [31]. The runs were produced from different LC-MS analyses separated by weeks, representing a potentially challenging alignment scenario. 6 runs were used in the experiment. Alignment ground truth was constructed from the putative identification of peaks in each of the 6 runs separately at 3 ppm using the Identify module in mzMatch [**?**], taking as additional input a database of 104 compounds known to be present and a list of common adducts in positive ionisation mode (Table 5.1). This is followed by matching of features sharing the same annotations across runs to construct the alignment ground truth. Only peaks unambiguously identified with exactly one annotation are used for this purpose; peaks with more than one annotation per run are discarded from the ground truth construction.

5.3.2 Performance Measures

Performance of the evaluated methods in our results is quantified in terms of precision and recall. These two measures are also commonly used in information retrieval, where 'precision' refers the fraction of retrieved items that are relevant, while 'recall' refers the fraction of relevant items that are retrieved [**?**].

To provide a definition of 'precision' and 'recall' suitable for evaluating alignment performance, we first enumerate all the possible q-size combinations for every aligned peakset in both the method's output and the ground truth list. For example, an alignment method returns a list of two aligned peaksets $\{a, b, c, d, \}, \{e, f, g\}$ as output. When $q = 2$, this 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 q reflects the strictness of what is considered to be a true positive item, with larger values of q demanding an alignment method that produces results spanning more runs correctly.

For a given q, 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 the context of alignment performance, precision ($\frac{\boldsymbol{TP}}{\boldsymbol{TP+FP}}$) is therefore the fraction of items in M that are correct with respect to G, while recall ($\frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}}$) is the fraction of items in G that are aligned in M. A method with a perfect alignment output would have both precision and recall values of 1.0.

5.3.3 Benchmarking Method

We benchmark HDP-Align against two established alignment methods: SIMA [25] and MZmine2's Join Aligner [22]. The performance of both methods have been evaluated in past studies [21, 22, 24, 25, **?**]. SIMA is a stand-alone program while Join aligner is an integral part of the MZmine2 suite widely used for the pre-processing of LC-MS data. The selection of SIMA and Join as the benchmark methods is motivated by the fact that both methods are direct matching methods (thus easily comparable to HDP-Align) but still differ sufficiently in how they establish the final alignment results. This is primarily due to the differences between both methods in the form of the distance/similarity function between peak features, 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 both 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 5.3.4) and report all performance values generated by each combination of the two parameters.

5.3.4 Parameter Optimisations

Table **??** describes the parameter ranges of each method during performance evaluation. For HDP-Align, 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.

For SIMA and Join, we report the results from all combinations of the mass and RT tolerance parameters within reasonable ranges. The ranges of $T_{(m/z)}$ and T_{rt} parameters used are based values reported on [21] for the Proteomic dataset and [**?**] for the Glycomic dataset. For the Metabolomic dataset, they were chosen in light of the mass accuracy and RT deviations of the data.

In HDP-Align, the mass cluster standard deviation $\sqrt{\rho^{-1}}$ 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 for the Glycomic and Metabolomic datasets, 500 posterior samples were collected for the Gibbs sampling procedure, discarding the first 500 during the burn-in period. For the Proteomic dataset with larger RT deviations, 5000 posterior samples were obtained after discarding the first 5000 samples during burn-in. The number of samples is selected to ensure convergence during inference.

5.4 Results

Precision and recall values for the evaluated methods methods on the different datasets are shown in Sections 5.4.1 and 5.4.2. 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 5.4.2. Running time of the evaluated methods are reported in Section 5.4.3.

5.4.1 Proteomic Results

Figure 5.4 shows the results from performance evaluation on the Proteomic 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

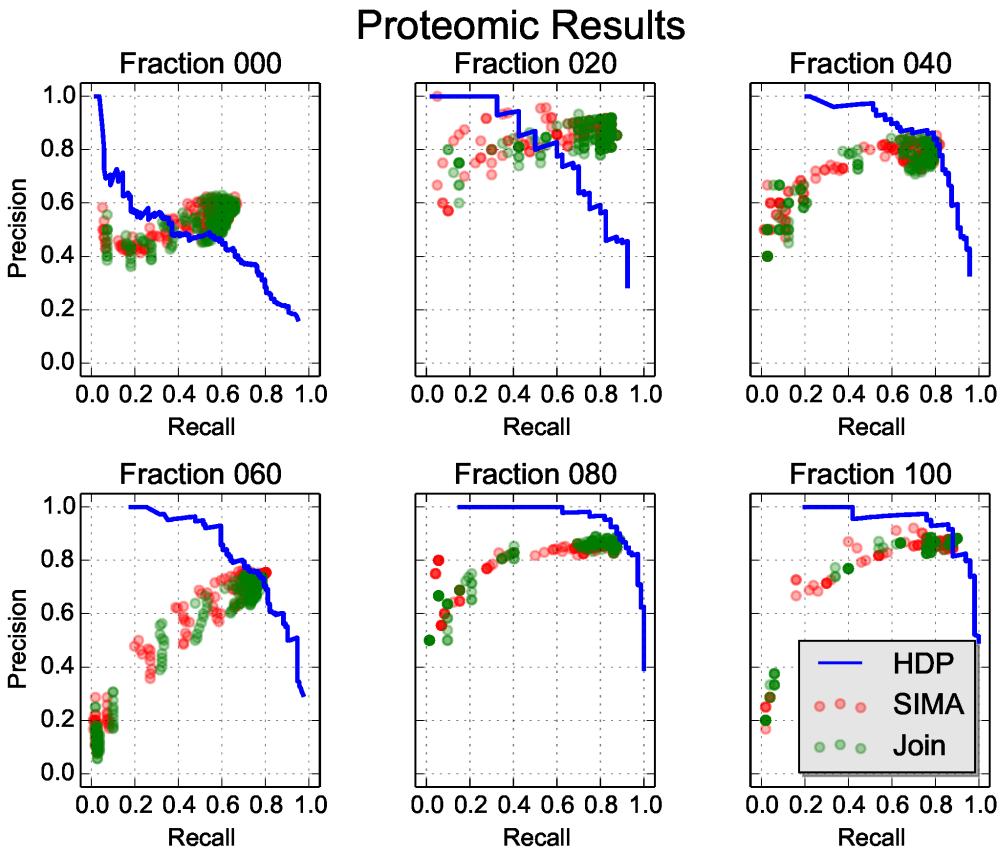

Figure 5.4: Precision-recall values on the different fractions of the Proteomic dataset.

the data and the high RT drifts present across runs (further details in [21]). 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 peak features produced 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.

5.4.2 Glycomic and Metabolomic Results

Figures 5.5 and 5.6 show the results from experiments on the Glycomic and Metabolomic datasets. Similar to the Proteomic dataset, a wide range of precision-recall values can be observed in the results for the benchmark methods on the two datasets. 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 5.5) also show some additional results on how the measured precision-recall values might change depending on the strictness of what constitutes an 'item' during performance evaluation. This is accomplished by gradually increasing the value for q (described in detail in Section 5.3.2) that determines the size of the feature combinations enumerated from a method's output. For example, q=2 considers all pairwise combinations of features from the method's output during performance evaluation, while $q = 4$ considers all combinations of size 4, and so on. Figure 5.5 shows that as q 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 qs 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 5.6. The results here appear to suggest the importance of having high mass precision during matching. Importantly, we see from Figure 5.5 that the performance of HDP-Align remains fairly consistent as q 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 [**?**], is beyond the scope of this paper and would likely necessitate using a different and more apppopriate evaluation dataset. Instead, we present an example of the further analysis performed by HDP-Align (as proposed in Section 5.2.3) on the resulting clustering objects after inference. Figure 5.7 shows a global RT cluster where peak features across runs have been grouped by their RT and m/z values. Within this global cluster, peak features are further separated into 6 mass clusters – corresponding to ionisation products produced by the global cluster during mass spectometry. In Figure 5.7, 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 great strength of our approach that this putative identification step appears very naturally from the alignment results.

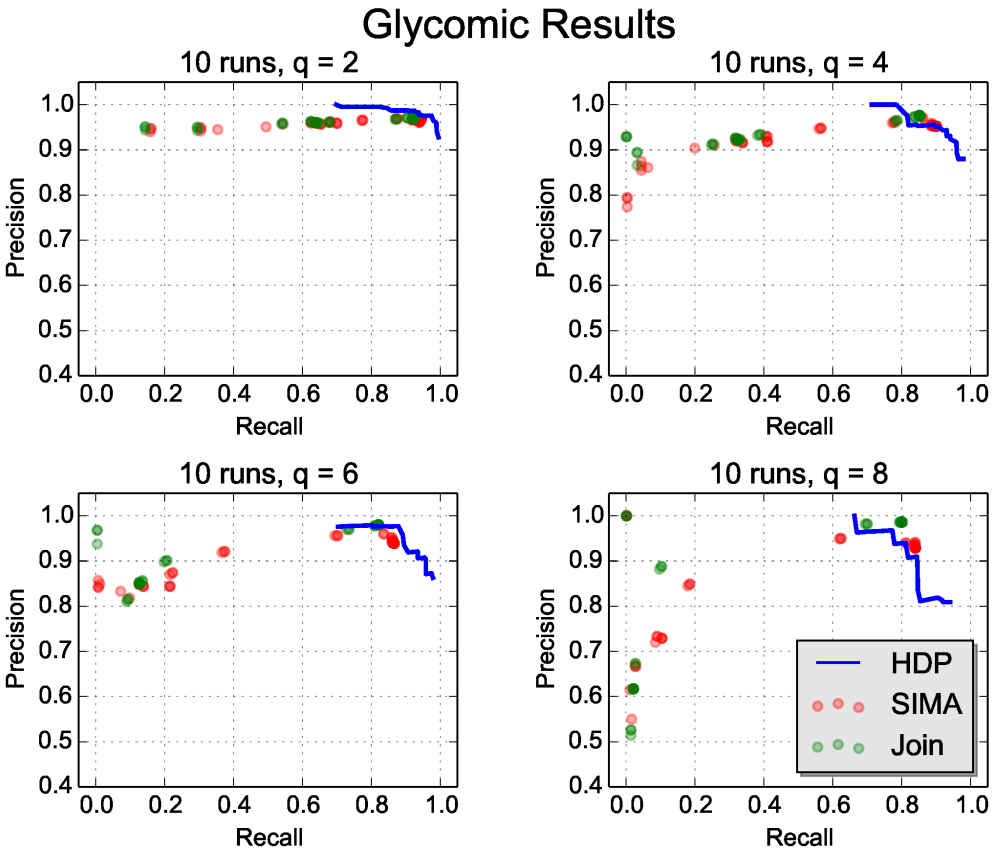

Figure 5.5: Precision-recall values on the alignment of 10 runs from the Glycomic dataset when q (the strictness of performance evaluation as described in Section 5.3.2) is gradually increased.

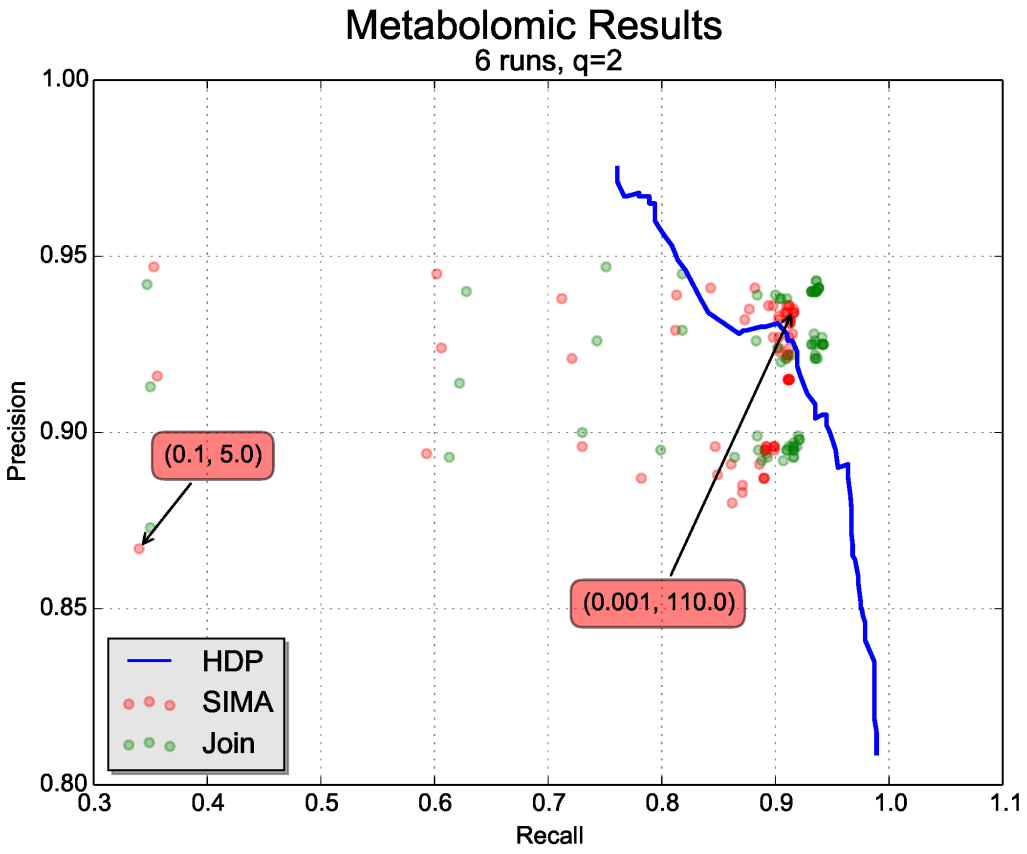

Figure 5.6: Precision-recall values on the alignment of 6 runs from the Metabolomic dataset. The parameter values $(T_{m/z}, T_{rt})$ that produce the best and worst performance in SIMA are also annotated in the Figure (red boxes).

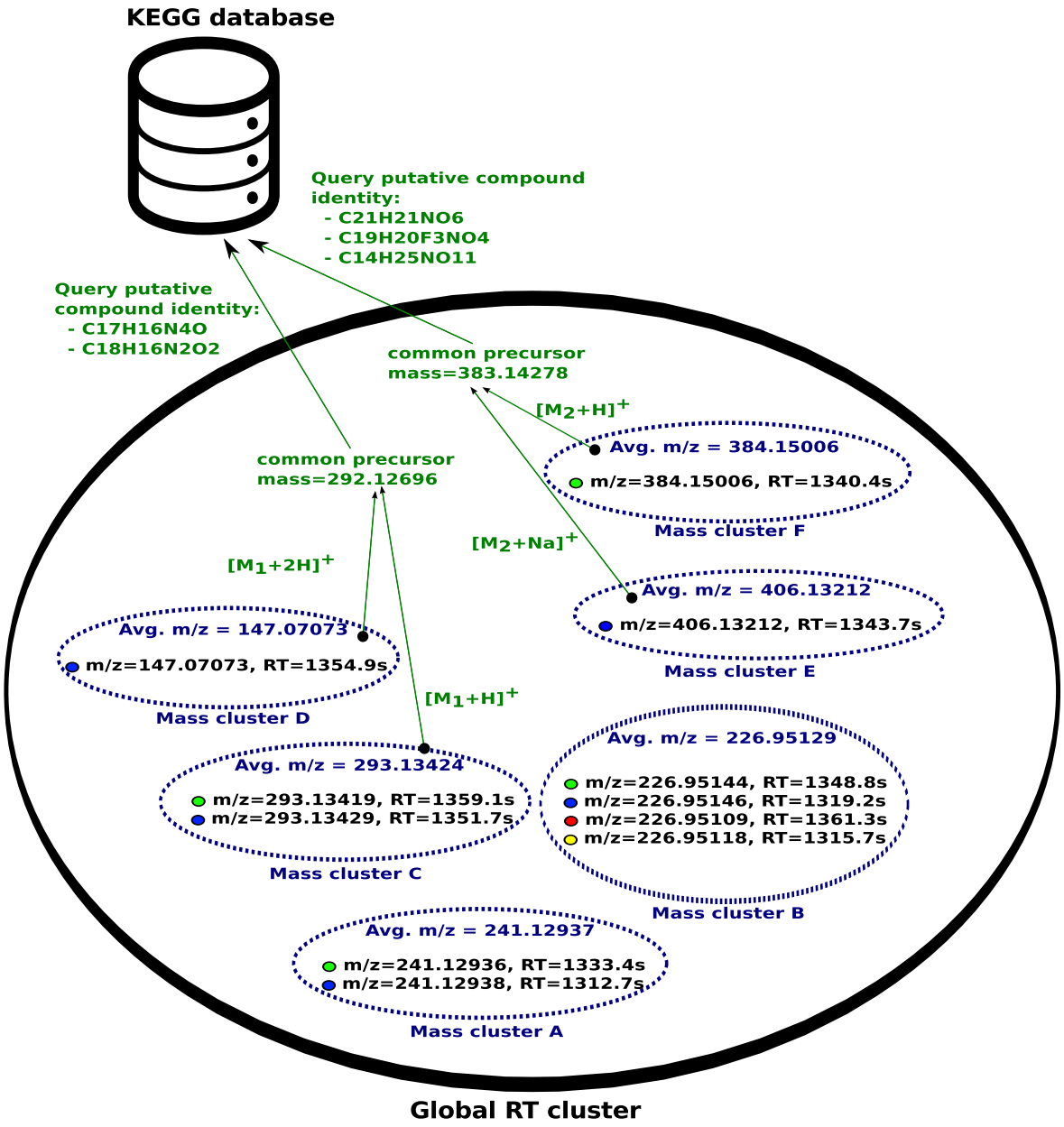

Figure 5.7: Example analysis that can be performed on the clustering objects inferred in a Gibbs sample from HDP-Align. The outer black oval denotes a global RT cluster (generally corresponding to a metabolite compound), while the smaller dotted ovals within denote mass clusters (labelled as mass cluster A, B, C, D, E, F). Peak features 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.

5.4.3 Running Time

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. The time complexity of this operation is $O(N)$, where N is the total number of peaks across all runs. In practice, additional time will also be spent on various necessary book-keeping operations, such as deleting empty clusters that are no longer required, updating internal data structures, etc.

A representative running time is given as $N = 9344$ for the Glycomic dataset. HDP-Align requires approximately 5 hours to collect 1000 samples. In comparison, both SIMA and Join perform alignment within 5 to 10 seconds. Similarly, for $N = 7477$ for the Metabolomic dataset, HDP-Align produces the results in approximately 4 hours after collecting 1000 samples, while SIMA and Join complete within seconds. The running time of HDP-Align, while being significantly longer than these two benchmark methods, is comparable to other computationally-intensive steps (e.g. peak detection) in a typical LC-MS pipeline.

5.5 Discussion and Conclusion

We have presented a hierarchical non-parametric Bayesian model that performs direct matching of peak features, a problem of significant importance in the data pre-processing pipeline of large untargeted LC-MS datasets. Unlike other direct matching methods, the novelty of our proposed approach lies in its ability of to produce well-calibrated probability scores on the matching confidence of aligned peak features (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 peak features as a hierarchical clustering problem. Matching confidence can then be obtained based on the probabilities of co-eluting peak features to be assigned under the same mass component of 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 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.

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 [16, **?**] that thorough investigations into the influence of numer-

ous 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 5.2.3 is fairly simple, it can be easily extended to more sophisticated annotation strategies, such as in CAMERA [35].

A primary weakness of HDP-Align lies in the long computational time required to produce results. Additional work will be required to reduce the computational burden of the model through various optimisation tricks and potentially by parallelising the Gibbs inference step using e.g. the method described in [**?**]. Another possibility is to adopt a different non-sampling-based inferential approach while still retaining the essence and benefits of the HDP model in this paper. The results presented in the current paper suggest the method shows enough promise to warrant the effort to speed it up.

Another aspect worthy of investigation is determining the most effective way to present and visualise the alignment probabilities produced by HDP-Align. 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.

Finally, replacing or enhancing the mixture of mass components used in HDP-Align with a more appropriate mass model, such as that in MetAssign [36] that specifically takes into account the inter-dependency structure of peaks, is an avenue for future work. This will be particularly useful when extending the proposed model in HDP-Align into a single inferential model that encompasses many intermediate steps in a typical LC-MS data processing pipeline.

Chapter 6

Precursor Clustering of Ionisation Product Peaks

Note:*[Around 30 pages?]*

6.1 Introduction

Since ionization product grouping and alignment are such important steps that occur early in the typical LC-MS data processing workflow, it is desirable to devise a method that allows for the sharing of information from one step to another. [**?**] suggests that the alignment objective function used to establish correspondence of peaks between runs can be improved by operating on groupings of related IP peaks rather than at individual peak level alone. This idea has been explored in our previous work of [**?**] that uses the grouping of related peaks to modify the similarity scores for matching with the aim of improving alignment results. However, in [**?**], the grouping of related peaks is performed based on the retention time of peak features alone. Valuable information present in the mass domain and also in the chemical relationships of related peaks is not used in the grouping process. In this work, we propose a novel Bayesian mixture model (PrecursorCluster) to perform the ionization product clustering of related peak features based on mass, retention time and a list of possible ionization transformations, bringing together peaks that share chemically meaningful relationships and that can each be related to a common precursor peak according to a set of transformation rules configurable by the user. While PrecursorCluster can potentially be used in a similar manner as our previous work of MetAssign [36] 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. Unlike CAMERA [35], which approaches the

problem of ion species annotation from a graph-theoretic point-of-view, PrecursorCluster is a fully probabilistic model that can be used to provide us with an estimate in 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.

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. Subsequently, two alternative methods are explored in this paper for establishing the correspondences of IP clusters across runs: (i) a fast direct-matching method of IP clusters (Cluster-Match) 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) a second-stage clustering model (Cluster-Cluster) that constructs alignment by means of grouping IP clusters according to their likelihood of being assigned to the same same top-level cluster (in this manner, IP clusters assigned to the same top-level cluster are considered to be matched). The two methods differ in the way the final alignment results are constructed: Cluster-Match returns a list of aligned peaksets that maximizes the weight of the matching, while Cluster-Cluster performs inference on which IP clusters should be put together into the same top-level clusters. Crucially, the resulting posterior summaries from the inference on the Cluster-Cluster model provide us with a confidence score in the matching quality of aligned peaksets, letting us robustly identify uncertainties in the alignment results.

6.2 Related Work

6.3 Methods

Figure 6.1 illustrates the entire proposed methods for the clustering of ionization product peaks and how they can be used for alignment. In the first **PrecursorCluster** stage (Section 6.3.1), we introduce a novel Bayesian clustering algorithm for grouping ionisation products based on a set of known ionization transformations. This is executed independently for each input file (where each file corresponds to a single LC-MS run), resulting in a set of IP clusters for each run. The assignment of individual peaks into their most likely IP clusters is then performed based on their maximum *a posteriori* probabilities. Once assigned, peaks are annotated based on the most likely transformation type that brings them into an IP cluster. The set of peaks assigned to an IP cluster and their respective transformations form a distinct ionization product 'fingerprint' of each IP cluster.

In the next stage, IP clusters containing ionization product peaks that have been grouped together by PrecursorCluster (according to their relationships to the precursor peaks) now

have to be matched across runs. The most straightforward way is to directly match IP clusters across runs based on the estimated m/z and RT values of the precursor peaks. We call this the **Cluster-Match** method (Section 6.3.2). Another possible approach is to perform a further clustering of the IP clusters themselves. This **Cluster-Cluster** method is introduced in Section 6.3.3. In this approach, IP clusters coming from different runs are first separated into top-level mass bins spanning across runs. IP clusters in the same top-level mass bin are then further separated into top-level clusters containing IP clusters of similar posterior m/z and RT values and ionization product fingerprints. Intuitively, the top-level clusters correspond to metabolites that are present across runs. Once top-level clusters have been constructed, the actual peak correspondence can easily established by simply matching peaks in the same top-level clusters that share the same transformation type that brings them into the IP clusters.

6.3.1 PrecursorCluster: clustering of ionization product peaks

A metabolite having a given precursor peak with mass M will be observed as multiple ionization product peaks. For example, rather than measuring the precursor peak with mass M, we might measure several IP peaks having the ion masses $[M + H]^+$, $[M + K]^+$ or $[M + ACN + H]^+$. The first-stage clustering model introduced here allows us to perform the grouping of these ionization product peaks generated from the same metabolite. The clustering model takes the form of a mixture model with finite mixture components.

Within a single run to be processed, we denote its n-th peak feature as $\mathbf{d}_n$, where $\boldsymbol{d}_n = (m_n, t_n, p_n)$ with m_n the m/z value, t_n the RT value and p_n the intensity value of that peak feature. Assuming that the data is in positive ionization mode and given a list of T transformations of commonly-known IP types (e.g. $[M + H]^+$, $[M + 2H]^+$, $[2M + Na]^+$), an observed ion peak can possibly be assigned to an IP cluster k only if the peak's ionized m/z value m_n can be transformed through a transformation u_k to a precursor mass $u_k(m_n)$ that falls within a certain user-defined tolerance γ_m parts-per-million (ppm) from the precursor mass of the IP cluster. The transformation u_k of the observed mass of peak n to the precursor mass of a cluster k takes the form of $u_k(m_n) = \frac{m_n|c|+ce-\sum_i h_i G_i}{m}$, where c is the charge, e is the mass of an electron and h_i and G_i are the atomic masses of adduct parts. Using the adduct transformation $[M + H + 2Na]^+$ for positive ionization mode data as an example, c is 3, m is 1 while $\sum_i h_i G_i$ is the total atomic mass of $H + 2Na$.

Through the $[M + H]^+$ transformation, each observed peak produces a candidate IP cluster (corresponding to a mixture component in the model) This follows from our assumption that for a candidate IP cluster to be considered valid, it must contain the $[M + H]^+$ ion peak that is also the most intense peak in that cluster. Each candidate k-th IP cluster now has a tuple of three values associated to it: (q_k, r_k, s_k) where the cluster's mass center q_k is produced by

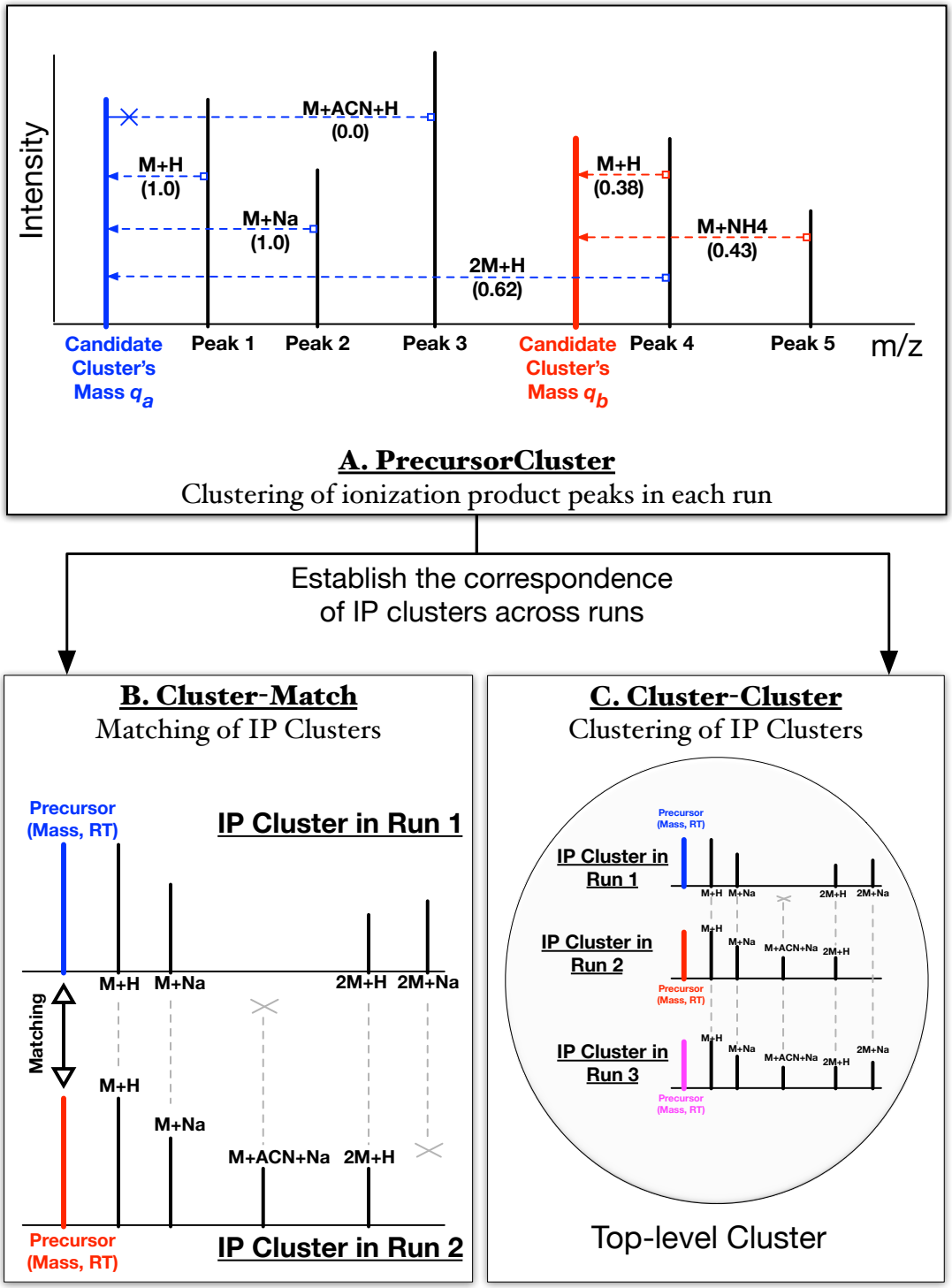

Figure 6.1: An illustration of the proposed methods. The results from inference on the PrecursorCluster model are the set of peaks and their assignments to IP clusters through any of the predefined list of transformation (Fig. 6.1A). As a starting point, each observed peak feature generates a candidate IP cluster, having the cluster's mass computed through the M+H transformation of the observed peak's m/z value. In Fig. 6.1A, this results Peak 1 generating the candidate cluster with mass q_a Peak 4 generating the candidate cluster with mass q_b (other candidate IP clusters produced by Peak 2, 3 and 5 are not shown in the figure). As a result from inference, Peak 1 and Peak 2 are clustered to q_a through the transformation types M+H and M+Na respectively with probability 1.0. Peak 3 has a valid transformation path to q_a, but it is not allowed to join that cluster since its intensity is greater than the intensity of the precursor peak. Peak 4 can either form a cluster with q_a through the 2M+H transformation (with probability 0.62) or, through the M+H transformation on itself, form its own candidate IP cluster having the precursor mass q_b (with probability 0.38.) The latter allows

computing the transformed precursor mass under the most common IP type $[M+H]^+$ from each observed ion mass in the data, while the cluster's RT center r_k and the cluster's intensity threshold s_k respectively are the RT and intensity values of the ion peak that produces this cluster through the $[M+H]^+$ transformation.

As the next step, we enumerate for each observed peak in the entire run which candidate IP clusters it can possibly be assigned to and through which chemical transformations. As constraints during the enumeration step, an assignment of peak n to candidate cluster k is possible only 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 the cluster's precursor mass q_k, **(2)** the RT value of that peak is within a certain tolerance (γ_t seconds) from the cluster's RT r_k and **(3)** the intensity of that peak is less than the cluster's intensity s_k. This follows from the modelling assumptions that ionization product peaks should co-elute together, having similar RT values to its precursor peak, and the $[M+H]^+$ peak is also be the most intense member peak of that IP cluster. From this enumeration process, it follows that an observed ion peak always satisfies all the constraints of its own IP cluster through the $[M+H]^+$ transformation and thus can belong to at least one IP cluster (its own). In dealing with multiple runs, this enumeration process can be performed iteratively for each run being processed or even run in parallel since they are completely independent of each other.

After enumeration, peaks may have multiple possible candidate IP clusters that they can join through the different transformation types. We use the indicator variable $\boldsymbol{z}_{nk}$ to denote the possible assignment of peak feature n to cluster k. Here, $\boldsymbol{z}_{nk}$ is 1 if peak n is assigned to cluster k and 0 otherwise, and each peak can only be assigned to exactly one cluster ($\sum_{k=1}^{K} \boldsymbol{z}_{nk} = 1$). We can model $\boldsymbol{z}_n$ 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 n to be assigned to a possible cluster k depends on the likelihood of that peak's transformed precursor mass and RT values under the cluster's mass and RT centers. The complete model for the clustering of ionization product peaks is therefore:

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

$$\boldsymbol{z}_n|\boldsymbol{\theta} \sim Multinomial(\boldsymbol{\theta}) \tag{6.2}$$

$$\boldsymbol{d}_n|\boldsymbol{z}_{nk} = 1, ... \sim L(\boldsymbol{d}_n|\boldsymbol{z}_{nk} = 1, ...) \tag{6.3}$$

where $L(\boldsymbol{d}_n|\boldsymbol{z}_{nk} = 1, ...)$ is the likelihod of peak n under cluster k and ... is to denote any other parameters being conditioned upon but not explicitly listed to the right of the conditioning bar. Assuming that the likelihood term in eq. (3) can be factorized into independent

mass and RT terms, we get:

$$L(\boldsymbol{d}_n|\boldsymbol{z}_{nk}=1,...) = p(u_k(m_n)|\boldsymbol{z}_{nk}=1,...)\cdot p(t_n|\boldsymbol{z}_{nk}=1,...) \tag{6.4}$$

For the mass term $p(u_k(m_n)|\boldsymbol{z}_{nk}=1,...)$ in eq. (4), the likelihood of the transformed precursor mass $u_k(m_n)$ is a product of two further terms, shown in eq. (5). The first is the indicator function $I(u_k)$, set to 1 if there are no other peaks assigned to cluster k through transformation u_k, and 0 otherwise. This allows each transformation type to only appear once in each IP cluster. Next, the transformed precursor mass $u_k(m_n)$ is distributed according to a Gaussian distribution with mean equal to the cluster mass center q_k and precision (inverse variance) of δ. We set δ to reflect the mass tolerance in parts-per-million used during the enumeration of possible assignments of peaks to potential IP clusters, such that one standard deviation ($\sqrt{\delta^{-1}}$) is set to be $\frac{\gamma_m * q_k/1e6}{3}$. The cluster mass center q_k is in turn drawn from a prior Gaussian distribution having prior mass mean μ_0 and precision δ. Since the prior value of the cluster mass mean q_k is pretty constrained and cannot deviate far from the transformed $[M+H]^+$ mass value, it makes sense to set $\mu_0 = q_k$ in the prior distribution for q_k in eq. (6).

$$p(u_k(m_n)|\boldsymbol{z}_{nk}=1,q_k,\delta,...) \sim I_k(n)\cdot\mathcal{N}(u_k(m_n)|q_k,\delta^{-1}) \tag{6.5}$$

$$p(q_k|\mu_0,\delta) \sim \mathcal{N}(q_k|\mu_0,\delta^{-1}) \tag{6.6}$$

Similarly for the RT term $p(t_n|\boldsymbol{z}_n=k,...)$ in eq. (4), the RT value t_n is distributed according to the Gaussian distribution having mean the cluster RT center q_k and precision λ set to reflect the RT tolerance used during enumeration of possible assignments, i.e. γ_t is $3\sqrt{\lambda^{-1}}$. The cluster RT center r_k is drawn from a prior Gaussian distribution with mean ψ_0 (set to be the same as r_k itself) and precision λ.

$$p(t_n|\boldsymbol{z}_{nk}=1,r_k,\lambda) \sim \mathcal{N}(t_n|r_k,\lambda^{-1}) \tag{6.7}$$

$$p(r_k|\psi_0,\lambda) \sim \mathcal{N}(r_k|\psi_0,\lambda^{-1}) \tag{6.8}$$

Given the complete specifications of the model in eq. (1-8), our goal is to infer $\boldsymbol{z}_{nk}$, the assignments of peak n to possible cluster k. A Gibbs sampling scheme [**?**] is used to approximate the joint distribution of the model during inference (details in the Supplementary document). Once Gibbs sampling has finished and enough posterior samples have been collected, peaks are then assigned to the most likely IP cluster based on their *maximum a-posteriori* (MAP) probabilities averaged across the samples obtained from Gibbs sampling after ensuring convergence. The final result from inference is the set of IP clusters, some of which may be empty and can be ignored while others consist of the observed peak that can be transformed into that cluster's precursor mass through the $[M+H]^+$ transformation, alongside with other peaks that can be related to the precursor mass through the set of

transformations defined by the user. Additionally, for each IP cluster, we also compute the precursor peak's mass and RT values from the posterior distributions of those parameters. These will be useful in the latter stages of matching IP clusters across runs.

The ionization product clustering model described in this section potentially has many uses, e.g. in the problem of annotation of related peaks and the identification of metabolites. However, in the following sections of the paper, we demonstrate its application to the problem of alignment (matching) of peak features across runs.

6.3.2 Cluster-Match: direct matching of ionization product clusters

The ionization product clustering model described in Section 6.3.1 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.

The results from inference in Section 6.3.1 is the assignment of peaks to their most likely clusters, alongside the annotation of the most likely transformation type that brings the peaks into an IP cluster. Specifically for the purpose of matching, each cluster k in file j is now associated to $\boldsymbol{c}_{jk}$, where $\boldsymbol{c}_{jk} = (\bar{q}_{jk}, \bar{r}_{jk})$. Here, $\bar{q}_{jk}$ is the posterior precursor mass value for that cluster, which takes into account the prior probability on the mass and also the transformed precursor masses of observed peaks assigned to it. Similarly, $\bar{r}_{jk}$ is the posterior RT value for cluster k in file j. The same procedure used for matching peak features across runs can now be used to match IP clusters across runs and is briefly summarized in the following paragraph.

In the direct matching alignment of two runs, the problem of establishing correspondence 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 [**?**] is a 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 [**?**] and operates as follows: add the heaviest edge e in G into the matching solution M and remove e and all other edges adjacent to e from G. Repeat this procedure until all edges in G have been removed. 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 greedy approximation has been shown in our experiments to be competitive in performance to other direct-matching methods, and more details on the MW method can be found in our previous work [**?**]. 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 correspondence 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 6.1B.)

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 [25, 22] 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.)

6.3.3 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 6.3.2). 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 [16]. 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 6.3.1. In this manner, IP clusters coming from different runs are further clustered into top-level clusters shared across runs (Figure 6.1C). The actual correspondence 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 together to establish their correspondences. 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.

As only peaks that are within a certain mass tolerance (which depends on the instrument's accuracy) should be matched across runs, we perform a preliminary partitioning of IP clusters based on their posterior precursor mass values into top-level mass bins. γ'_m is defined to be the across-run mass tolerance threshold used for the pre-grouping IP clusters coming from different runs. Across all runs $j = 1, ..., J$, IP clusters are first sorted by their posterior mass values $\{\bar{q}_{jk}\}$. The smallest mass value $min(\{\bar{q}_{jk}\})$ is used to group other IP clusters into the top-level bin. Grouping is done in successive ascending mass order until the next IP cluster to be grouped has the posterior mass value that differs by γ'_m ppm from $min(\{\bar{q}_{jk}\})$. In this case, a new top-level bin is started and the next ungrouped mass value is used for the grouping of the remaining other peaks. This procedure is repeated until all peaks have been exhausted.

Within a top-level mass bin, we now have IP clusters coming from different runs having similar posterior mass values but potentially different posterior RT values and member peaks. For each k-th IP cluster coming from file j, an adduct 'fingerprint' vector $\bar{\boldsymbol{u}}_{jk}$ of length T is created after the MAP assignment of observed peaks into the IP cluster. $\bar{\boldsymbol{u}}_{jk}$ stores the information on which adduct transformations bring member peaks into that IP cluster, with the corresponding entry in $\boldsymbol{u}_{jk}$ set to 1 if that transformation exists and 0 otherwise. Let y_i be the list of IP clusters coming from different runs that have been pre-grouped in a top-level mass bin i, i.e. $y_i = (\boldsymbol{c}_{jk}, ...)$ where $\boldsymbol{c}_{jk} = (\bar{q}_{jk}, \bar{r}_{jk}, \bar{\boldsymbol{u}}_{jk})$. Here, $\bar{q}_{jk}$ is the IP cluster's posterior mass value, $\bar{r}_{jk}$ the posterior RT value and $\bar{\boldsymbol{u}}_{jk}$) the adduct fingerprint of that IP cluster. If the top-level mass bin contains only 1 IP cluster, no possible matching can be constructed. In the case of more than 1 IP clusters present in a top-level bin, the IP clusters that have similar posterior mass and RT values and adduct fingerprints can be grouped together into a top-level cluster. To avoid specifying the number of top-level clusters *a priori*, we use an infinite Gaussian mixture model [51] to model the data.

Let the indicator $\bar{\boldsymbol{z}}_{jki} = 1$ denotes the assignment of IP cluster k coming from file j into top-level cluster i. Then:

$$\boldsymbol{\pi}|\alpha' \sim GEM(\alpha') \tag{6.9}$$

$$\bar{\boldsymbol{z}}_{jk}|\boldsymbol{\pi} \sim Multinomial(\boldsymbol{\pi}) \tag{6.10}$$

$$\boldsymbol{c}_{jk}|\bar{\boldsymbol{z}}_{jki} = 1, ... \sim L(\boldsymbol{c}_{jk}|\bar{\boldsymbol{z}}_{jki} = 1, ...) \tag{6.11}$$

where $\boldsymbol{\pi}$ is the mixing proportions, now distributed according to the GEM (Griffiths, Engen

and McCloskey) distribution (details in the Supplementary). In this model, the number of mixture components and the sparsity of each component depends on the concentration hyper-parameter α'. Next, the likelihood of $\boldsymbol{c}_{jk}$ (the k-th IP cluster from run j) to be in a top-level cluster i is assumed to be factorized into independent factors of its mass, RT and adduct signature terms:

$$L(\boldsymbol{c}_{jk}|\bar{\boldsymbol{z}}_{jki}=1,...) = p(\bar{q}_{jk}|\bar{\boldsymbol{z}}_{jki}=1,...)\cdot p(\bar{r}_{jk}|\bar{\boldsymbol{z}}_{jki}=1,...)\cdot p(\bar{\boldsymbol{u}}_{jk}|\bar{\boldsymbol{z}}_{jki}=1,...) \tag{6.12}$$

In this likelihood of eq. (12), the mass term $p(\bar{q}_{jk}|\bar{\boldsymbol{z}}_{jki}=1,...)$ is defined analogously to the first-stage clustering in Section 6.3.1. First, we have the indicator function $\bar{I}(jki)$ set to 1 if there are no other IP clusters from run j, apart from the k-th IP cluster in j, that are assigned to the i-th top-level cluster, and 0 otherwise. This ensures that there is only most one IP cluster from each run assigned to a top-level cluster. Next, 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 cluster mass center c_m is in turn drawn from a base Gaussian distribution having prior mass mean $\bar{\mu}_0$ and precision σ_m. We set $\bar{\mu}_0$ 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.

$$p(\bar{q}_{jk}|\boldsymbol{z}_{jki}=1,c_m,\bar{\delta},...) \sim \bar{I}(jki)\cdot\mathcal{N}(\bar{q}_{jk}|c_m,\bar{\delta}^{-1}) \tag{6.13}$$

$$p(c_m|\bar{\mu}_0,\sigma_m) \sim \mathcal{N}(c_m|\bar{\mu}_0,\sigma_m^{-1}) \tag{6.14}$$

Similarly, in the RT term $p(\bar{r}_{jk}|\boldsymbol{z}_{jki}=1,...)$, $\bar{r}_{jk}$ is distributed according to a Gaussian distribution with mean c_t and precision $\bar{\lambda}$. Again, the across-run RT tolerance γ'_t 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.

$$p(\bar{r}_{jk}|\boldsymbol{z}_{jki}=1,c_t,\bar{\lambda}) \sim \mathcal{N}(\bar{r}_{jk}|c_t,\bar{\lambda}^{-1}) \tag{6.15}$$

$$p(c_t|\bar{\psi}_0,\sigma_t) \sim \mathcal{N}(c_t|\bar{\psi}_0,\sigma_t^{-1}) \tag{6.16}$$

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 use of multinomial distribution here to model the binary vector $\bar{\boldsymbol{u}}_{jk}$ provides a flexibility to the Cluster-Cluster model, should the constraint of having each transformation type appear at most once in each IP cluster is removed from the first-stage clustering step.

$$\boldsymbol{\psi}|\beta \sim Dir(\beta) \tag{6.17}$$

$$\bar{\boldsymbol{u}}_{jk}|\boldsymbol{\psi} \sim Multinomial(\boldsymbol{\psi}) \tag{6.18}$$

Taken together, the entire likelihood function of eq. 12 ensures that IP clusters coming from different runs can only be put together in a single top-level cluster if: **(1)** they come 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 sample, we establish the correspondences of peaks in matched IP clusters by grouping peaks with the same transformation type together (Figure 6.1C), forming aligned peaksets. The occurrences of aligned peaksets are counted and averaged across samples to get the estimate of matching confidence. More details on the inference procedure of Cluster-Cluster can be found in the Supplementary document.

6.4 Evaluation Study

6.4.1 Evaluation Datasets

We evaluated the performance of the proposed alignment methods on two metabolomic datasets. The first dataset is the Standard dataset, generated from a mixture of 104 standard metabolites used for the calibration of chromatographic columns. This dataset, alongside its associated alignment ground truth, has been used for performance evaluation of alignment methods in our previous paper [**?**]. The Standard dataset has 11 runs in total. While the runs are not true technical replicates, they are similar enough to be treated as replicates for the purpose of performance evaluation. Indeed the Standard runs represent a fairly challenging alignment scenario due to the variability in the RT values of peak features as the runs were produced from different LC-MS analyses separated by weeks and on different instruments.

Additionally, in this paper, we also introduce a new dataset obtained from sampling a bottle of 'Seven Giraffes Extraordinary Ale'. This Beer dataset is available in 3 runs. While the Beer runs have minimal RT deviations, it is fairly is representative of the typical biochemical diversity in a complex metabolomics study and can provide a useful portrayal on how the evaluated methods might perform in real-life usage.

Details on the construction of the alignment ground truth for the Standard and Beer datasets can be found in the Supplementary document.

6.4.2 Performance Measures

The different alignment methods being compared return as results the matchings of peak features coming from different runs. In the case of Cluster-Cluster, such aligned peaksets also have probability values, potentially corresponding to matching confidence, attached to them. To evaluate alignment results on multiple runs, we propose generalizing the pairwise performance measures outlined in our previous work of [**?**] to the alignment of multiple runs.

Following [**?**], performance of the evaluated methods is quantified in terms of precision and recall. 'Precision' here refers the fraction of aligned peaksets returned by a method that are relevant (in accordance to some alignment ground truth), while 'recall' refers the fraction of relevant aligned peaksets that are returned by an alignment method.

To provide a definition of 'precision' and 'recall' suitable for evaluating alignment performance, we begin by enumerating all the possible l-size combinations for every aligned peakset in both the method's output and the ground truth list. For example, an alignment method returns a list of two aligned peaksets $\{a, b, c, d, \}, \{e, f, g\}$ as output. When $l = 2$, this 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 the context of alignment performance, precision ($\frac{\boldsymbol{TP}}{\boldsymbol{TP}+\boldsymbol{FP}}$) is therefore the fraction of alignment items in M that are correct with respect to some alignment ground truth G, while

recall ($\frac{TP}{TP+FN}$) 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.

6.4.3 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. These two methods are described in details in our previous work of [**?**] and have been found in performance evaluations to be competitive against the direct matching methods used in MzMine2's Join Aligner [22] and SIMA [25].

To evaluate Cluster-Match, we followed the experimental procedure in [**?**]. 30 random pairs of runs were selected from the Standard runs and designated as the Standard training set, and another 30 pairs of runs selected as the Standard testing set. Each run is then processed through the first-stage ionization product clustering (PrecursorCluster) using the same set of parameters, as detailed in Section 6.4.4. Matching tolerance parameters (see Section 6.4.4 too) were varied within reasonable ranges on one pair in the training set and parameters that result in the best training performance (highest F_1-score) were then applied onto its associated pair in the testing set. Measuring performance on a different testing set provide us with a picture on how the methods being evaluated generalize to new data in actual usage. The number of available 3 Beer runs is too few to allow separation of the runs into training and testing sets, so the 3 beer runs were used at once for performance evaluation. For each method, we report the results obtained from all combinations of parameters being varied on the Beer dataset, i.e. the training results only.

To evaluate Cluster-Cluster, we selected 5 sets of 2, 3, and 4 Standard runs randomly and also the same set of 3 Beer runs. All runs have been processed through the PrecursorCluster model. For each data set, parameters for Cluster-Match were varied following see Section 6.4.4 to obtain the best attainable alignment performance for comparison. The best results from Cluster-Match are then compared against the output from Cluster-Cluster on the same set of data but using a fixed set of parameters for the second-stage clustering. We also ran Cluster-Cluster with and without the adduct fingerprint term to evaluate the importance of that term to the alignment results of Cluster-Cluster.

6.4.4 Parameter Optimization

Common to all the direct matching method used in the evaluation study are the m/z and RT window tolerance parameters, which 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.

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 (more details in [**?**]). 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 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 **??** 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 (intuitively, large values of α' will produce more top-level clusters, each having fewer member IP clusters inside), while 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.

6.5 Results and Discussions

6.5.1 Improved peak alignment performance by using clustering information in Cluster-Match

Figure 6.2 (top row) shows the density plots of all the training precision and recall values produced by the different methods from the entire 30 training sets of pairwise Standard runs.

Here, we set l (the size of peakset combinations to be considered during performance evaluation) to 2 to consider only pairwise features for performance evaluation. The results in Figure 6.2 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 (as measured by precision and recall) compared to matching by peak features only.

Similar results can also be observed from the results for the Beer data (Figure 6.2 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 here we see a general increase in precision of the results from Cluster-Match over the other two baseline methods. The MWG method, 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.

The next results of Figure 6.3 combine precision and recall into a single measure of F_1-score. In Figure 6.3, we report the best F_1-scores produced by each method on the 30 sets of training and testing Standard runs. Consistent with the improvements in precision from the training results, Figure 6.3 shows that Cluster-Match is able to produce higher training and testing F_1-scores compared to MW. Using a one-sided paired t-test, the means of the F_1-scores for MW are found to be statistically less than that of Cluster-Match in both the training (p-value=0.002) and the testing cases (p-value=0.026). On the training results, MWG produces even higher training F_1-scores compared to the other two methods 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 t_g and matching ratio α_g for MWG were also optimized for each training set during the training phase, whereas the same set of clustering parameters were used when performing the first-stage clustering for each run in the PrecursorCluster model. On the testing results, we found no statistically significant differences on the testing F_1-scores of MWG and Cluster-Match, suggesting that both methods generalize well to new and unseen data. Taken together, the results in Figures 6.2 and 6.3 demonstrate that in general, constructing alignment by working with groups of related peaks (Cluster-Match) results in a better alignment performance compared to alignment based on individual peak features

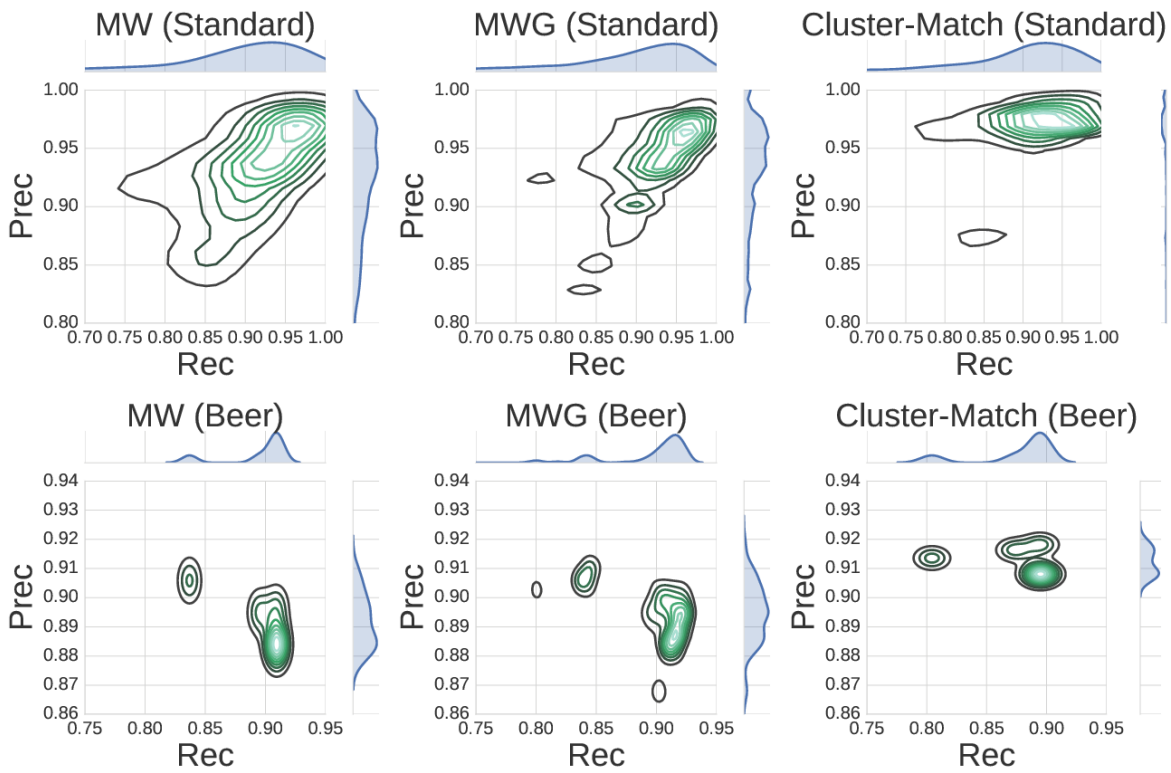

Figure 6.2: All the training results obtained by varying the m/z and RT window parameters from the alignment of the entire 30 sets of pairwise Standard runs (top row) and the 3 Beer runs (bottom row). For MWG, the grouping parameter t and score contribution α were also varied, while for Cluster-Match, the same set parameters of first-stage clustering was used for all input files.

(MW).

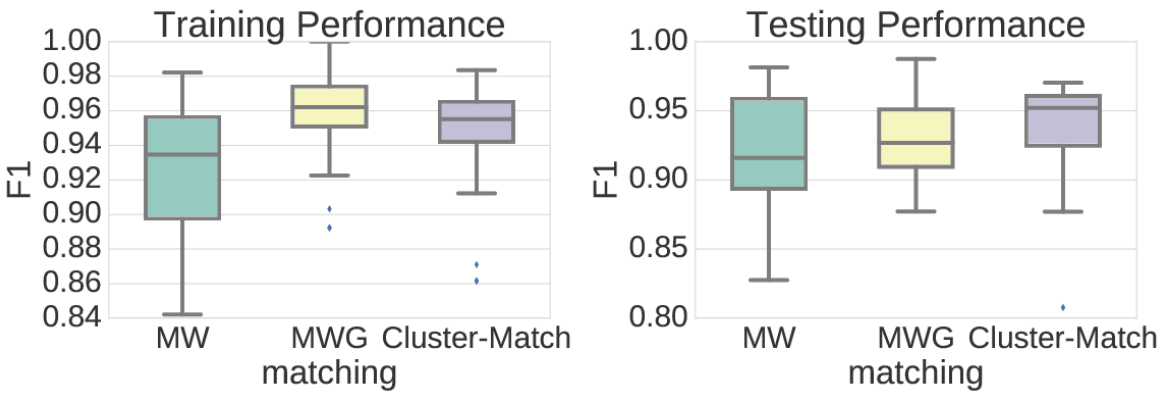

Figure 6.3: The best training and testing F_1-scores obtained from the alignment of 30 sets of pairwise Standard runs.

6.5.2 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. 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 6.4 shows a illustrative example of the results obtained by running Cluster-Cluster, using only one set of potentially sub-optimal parameters for the second-stage clustering, and Cluster-Match, using varying m/z and RT tolerance parameters, on one of the sets of 4 randomly selected Standard runs and the set of 3 Beer runs. A Precision-Recall (PR) curve, showing how precision and recall change together, can be computed from the output of Cluster-Cluster and is plotted alongside the results from Cluster-Match. We see from the examples in Figure 6.4 that generally, a decrease in the recall values is accompanied by an increase in the precision values along the PR curve on both the Standard and the Beer data. The results suggest 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).

Indeed, the results from our experiments on the randomly selected sets of 2, 3, and 4 Standard runs (Table **??**) and also on the 3 Beer runs (Table **??**) show that by setting some threshold values $\{0.30, 0.60, 0.90\}$ on the alignment results returned by Cluster-Cluster, we can vary the resulting precision and recall values. For instance, when looking at the the alignment of 5 sets of four Standard runs in Table **??** and considering only pairs of aligned peak features ($l = 2$) for performance evaluation, we see that the best average performance

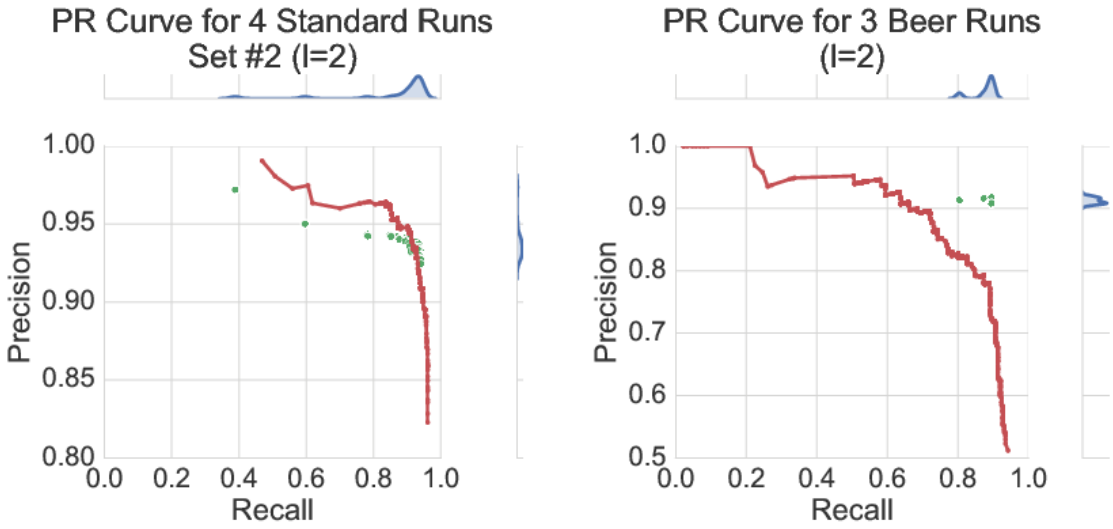

Figure 6.4: PR curves obtained from running Cluster-Cluster on one of the sets of 4 randomly selected Standard runs (left) and the 3 Beer runs (right). Green dots are performance points obtained from running Cluster-Match at varying m/z and RT tolerance parameters on the same datasets, with their distributions of the points plotted along the marginals. The same first-stage clustering results were used as input to both Cluster-Match and Cluster-Cluster.

from Cluster-Match is found at precision=0.93 and recall=0.94. At threshold=0.30, the result from Cluster-Cluster has a lower precision of 0.91 compared to Cluster-Match. By raising the threshold to 0.90 (consequently, decreasing recall as fewer aligned peaksets are now returned), we obtain from Cluster-Cluster an alignment result that has a higher precision value of 0.95. The results in Table **??** demonstrate that it is possible to extract subsets of the entire alignment results from Cluster-Cluster having a higher precision than can be attained by Cluster-Match with optimal m/z and RT tolerance values. Using Cluster-Cluster, the user can trade recall for precision; a potentially useful ability in the case where alignment ground truth is difficult to create or might not be available altogether. In this case, it might make sense to focus analysis effort on aligned peaksets with high confidence values in which we can be more certain that they have been correctly aligned.

Tables **??** and **??** also show the importance of taking into account the information on which adduct transformations have been assigned to an IP cluster (the adduct fingerprint) when grouping IP clusters coming from different runs together into the top-level clusters. At threshold 0.90, we observe a general decrease in the F_1-score performance from Cluster-Cluster without the adduct term on both the Standard and Beer datasets (Tables **??** and **??** respectively) in comparison to results from Cluster-Cluster with the adduct term. This can be explained by the fact that without using the adduct fingerprint term in the likelihood function, Cluster-Cluster allows IP clusters to be incorrectly put together as they have highly similar precursor mass and RT values but entirely different sets of member adduct ions, potentially

corresponding to different metabolites. This makes establishing the correspondence of peak features that should have be matched according to the alignment ground truth impossible, resulting in lower recall values upon evaluation. The inclusion of the adduct fingerprint term in the likelihood function of Cluster-Cluster is therefore necessary to ensure that we get a well-calibrated probabilities on the alignment results, especially on the higher-confidence regime that we are most interested in.

6.5.3 Running time

The first-stage clustering of PrecursorCluster is performed independently for each run. In practice, many peaks have no possible transformation except to its own candidate cluster and as such, not all peaks have to be resampled during inference via Gibbs sampling. This allows for efficient inference. On the Standard runs, where each input run may consist up to 5000 peaks, approximately fewer than a quarter of those peaks have to be reassigned. Taking 10000 posterior samples for each run, Gibbs sampling took approximately 20 minutes per Standard run on our desktop machine (Intel Core i5, 3.3GHz with 8GB of RAM). Furthermore, this step can easily be run in parallel as each run is processed independently of the others.

In the next stage of Cluster-Match, IP clusters are considered to be input features used for matching across runs. The approximate matching algorithm (MW) used in Cluster-Match runs in $O(m \log n)$ time, where n and m are respectively the number of vertices and edges in the graph G to be solved. In practice, this translates to a running time of less than a minute for each sets of Standard runs being processed. The alternative second-stage clustering via Cluster-Cluster requires longer time. By taking 1000 samples for the clustering of IP clusters in each top-level bins, the processing of 2 Standard runs takes approximately half an hour. The entire Cluster-Cluster step can also be easily parallelized as each top-level bin can be processed independently of the others.

6.6 Conclusion

In this paper, we have proposed a method that performs precursor clustering of related ionization product peaks and uses that clustering information to improve alignment (as measured by precision, recall and F_1-scores). The valuable information extracted from the first-stage clustering process of ionization product peaks, which lies at the heart of our proposed method, can potentially be used to improve other steps in the pipeline too. For instance, the identification of metabolites, which is currently one of the bigger bottlenecks in LC-MS data processing, can conceivably be improved by operating on IP cluster-level rather than

individual peak-level. This approach has been attempted in our previous work of MetAssign [36], which considers the relationship between adducts and isotopes in producing probabilistic model for the identification and annotation of peak features. In that work, peak features coming from different runs have to be aligned (matched) first to form a single consensus feature and a database of known compounds have to be provided. The proposed work in this paper addresses the problem differently by being able to work with peak features coming from multiple runs and not requiring such compound library to be provided for the grouping of ionization product peaks. As future work, a hybrid model can be developed that combines the best aspects of MetAssign and PrecursorCluster, such that the prior information of compounds expected to be present in the sample is used for clustering some peaks as in MetAssign, while other peaks are clustered entirely based on their possible transformations, as what PrecursorCluster has done.

Additionally, once the first-stage clustering of ionization product has been created, we have also demonstrated that establishing the correspondence of IP clusters across runs can be performed flexibly. We showed how the results from the direct matching of IP clusters (Cluster-Match) outperforms matching of peak clusters only. In addition, an alternative second-stage clustering of the IP clusters (Cluster-Cluster) is introduced that allows us to obtain probabilities that reflects confidence in the matching of the aligned peaksets. The subject of identifying and quantifying uncertainty has been extensively investigated in the problem of multiple sequence alignment (MSA) for genomics and transcriptomics [**?**, see e.g.]]Landan2009, Notredame2000, Penn2010, however 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. 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.

For practical use of our methods, we foresee the need to develop an interactive visualisation module, which lets user visualises the set of peaks that have been put together across different runs and their matching probabilities. Enhancements to the first-stage precursor clustering model can also be performed by taking into account the chromatographic peak shapes of IP peaks during clustering, and also the predicted RT time of metabolites. We also foresee as our long term goal a fully-integrated probabilistic LC-MS data processing pipeline that propagates information and uncertainties across the different steps of the pipeline. It is our hope that the ionization product clustering model, the Cluster-Match approach and the Cluster-Cluster model described in this paper may inspire others along the same research direction.

Chapter 7

Substructure Discovery in Tandem Mass Spectometry Data

Note:*[Around 20 pages]*

7.1 Introduction

7.2 Latent Dirichlet Allocation for Substructure Discovery

7.3 Evaluation Study

7.4 Results

7.5 Discussion and Conclusion

Chapter 8

Conclusion

Note:*[About 5 pages?]*

8.1 Summary of Contributions

8.2 Future Work

8.3 Summary and Conclusions

Appendix A

An Appendix

This is an appendix.

Bibliography

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

[2] J. H. Gross, *Mass spectrometry: a textbook*. Springer Science & Business Media, 2006.

[3] 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.

[4] 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. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1570963913001398

[5] 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. Suppl 7, p. S9, 2014. [Online]. Available: http://www.biomedcentral.com/1471-2105/15/S7/S9

[6] 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, aug 2011. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0169743911000608

[7] J. Xiao, B. Zhou, and H. Ressom, "Metabolite identification and quantitation in LC-MS/MS-based metabolomics," *TrAC Trends in Analytical Chemistry*, pp. 1–14, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0165993611003165

[8] 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.

[9] M. L. Metzker, "Sequencing technologiesthe next generation," *Nature reviews genetics*, vol. 11, no. 1, pp. 31–46, 2010.

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

[11] M. Katajamaa and M. Oresic, "Data processing for mass spectrometry-based metabolomics." *Journal of chromatography. A*, vol. 1158, no. 1-2, pp. 318–28, jul 2007. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/17466315

[12] T. M. Annesley, "Ion suppression in mass spectrometry." *Clinical chemistry*, vol. 49, no. 7, pp. 1041–4, jul 2003. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/12816898

[13] "A common open representation of mass spectrometry data and its application to proteomics research." *Nature biotechnology*, vol. 22, no. 11, pp. 1459–66, nov 2004. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/15529173

[14] J. Zhang, E. Gonzalez, T. Hestilow, W. Haskins, and Y. Huang, "Review of peak detection algorithms in liquid-chromatography-mass spectrometry." *Current genomics*, vol. 10, no. 6, pp. 388–401, sep 2009. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2766790{&}tool=pmcentrez{&}rendertype=abstract

[15] 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, oct 2008. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/18790129

[16] R. Smith, D. Ventura, and J. T. Prince, "{LC}-{MS} alignment in theory and practice: a comprehensive algorithmic review," *Briefings in Bioinformatics*, 2013.

[17] 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, feb 1978. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1163055

[18] 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. 1-2, pp. 17–35, may 1998. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0021967398000211

[19] P. H. C. Eilers, "Parametric time warping." *Analytical chemistry*, vol. 76, no. 2, pp. 404–11, jan 2004. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/14719890

[20] C. Christin, A. K. Smilde, H. C. J. 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. 7012–21, sep 2008. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/18715018

[21] 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.

[22] T. Pluskal, S. Castillo, A. Villar-Briones, and M. Oresic, "MZmine 2: modular framework for processing, visualizing, and analyzing mass spectrometry-based molecular profile data." *BMC bioinformatics*, vol. 11, no. 1, p. 395, jan 2010.

[23] 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. [Online]. Available: http://www.biomedcentral.com/1471-2105/13/214/

[24] R. Ballardini, M. Benevento, and G. Arrigoni, "MassUntangler: A novel alignment tool for label-free liquid chromatography-mass spectrometry proteomic data," ... *of Chromatography A*, vol. 1218, no. 49, pp. 8859–68, dec 2011. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/21783198http://www.sciencedirect.com/science/article/pii/S0021967311008776

[25] 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 (Oxford, England)*, vol. 27, no. 7, pp. 987–93, apr 2011.

[26] 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, nov 2003. [Online]. Available: http://bioinformatics.oxfordjournals.org/cgi/doi/10.1093/bioinformatics/btg315

[27] J. Wang and H. Lam, "Graph-based peak alignment algorithms for multiple liquid chromatography-mass spectrometry datasets." *Bioinformatics (Oxford, England)*, vol. 29, no. 19, pp. 2469–2476, aug 2013. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/23904508

[28] H. Lin, L. He, and B. Ma, "A Combinatorial Approach to the Peptide Feature Matching Problem for Label-Free Quantification." *Bioinformatics (Oxford, England)*, pp. 1–7, may 2013. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/23665772

[29] "Retention time alignment algorithms for LC/MS data must consider non-linear shifts." *Bioinformatics (Oxford, England)*, vol. 25, no. 6, pp. 758–64, mar 2009. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/19176558

[30] 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." *Nat. Protoc.*, vol. 2, no. 4, pp. 778–791, jan 2007.

[31] 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.

[32] 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. [Online]. Available: http://dx.doi.org/10.1021/pr500665j

[33] W. Dunn, A. Erban, R. Weber, and D. Creek, "Mass appeal: metabolite identification in mass spectrometry-focused untargeted metabolomics," *Metabolomics*, 2012. [Online]. Available: http://www.springerlink.com/index/0718581530254PG6.pdf

[34] 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, jan 2006. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1464138{&}tool=pmcentrez{&}rendertype=abstract

[35] 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, jan 2012. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/22111785

[36] 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 (Oxford, England)*, vol. 30, no. 19, pp. 2764–2771, jun 2014.

[37] 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. [Online]. Available: http://dx.doi.org/10.1016/j.trac.2013.09.008

[38] 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, dec 2005. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/16404815

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

[40] 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.

[41] Y. Wang, J. Xiao, T. O. Suzek, J. Zhang, J. Wang, and S. H. Bryant, "Pubchem: a public information system for analyzing bioactivities of small molecules," *Nucleic acids research*, p. gkp456, 2009.

[42] 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. [Online]. Available: http://pubs.acs.org/cgi-bin/doilookup/?10.1021/ci9501406

[43] 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.

[44] J. Xia and D. S. Wishart, "MetPA: a web-based metabolomics tool for pathway analysis and visualization." *Bioinformatics (Oxford, England)*, vol. 26, no. 18, pp. 2342–4, sep 2010. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/20628077

[45] J. Krumsiek, K. Suhre, and T. Illig, "Gaussian graphical modeling reconstructs pathway reactions from high-throughput metabolomics data," *BMC systems* . . . , 2011. [Online]. Available: http://www.biomedcentral.com/1752-0509/5/21/

[46] 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, oct 2012. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/23093944

[47] "Genetics meets metabolomics: a genome-wide association study of metabolite profiles in human serum." *PLoS genetics*, vol. 4, no. 11, p. e1000282, nov 2008. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2581785{&}tool=pmcentrez{&}rendertype=abstract

[48] 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.

[49] "Metabolomics in human nutrition: opportunities and challenges." *The American journal of clinical nutrition*, vol. 82, no. 3, pp. 497–503, sep 2005. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/16155259

[50] 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, dec 2006. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/17129827

[51] C. Rasmussen, "The infinite Gaussian mixture model," in *Advances in Neural Information Processing Systems 12*. MIT Press, 2000, pp. 554–560.