Prioritisation Algorithms for Data Acquisition in Liquid Chromatography Mass Spectrometry

Ross Daniel McBride

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 2025

Abstract

Liquid chomatography mass spectrometry (LC-MS/MS) is a powerful analytical platform frequently used to identify the composition of biological samples. For example, LC-MS/MS is one of the leading measurement technologies within metabolomics, which has applications in discovering disease biomarkers and novel drugs, in ecology and environmental science and in forensics and toxicology, among many others.

The goal of an untargeted LC-MS/MS experiment is to discover as many unique analytes in the sample as possible in order to generate hypotheses relevant to the experiment's goals. One of the most powerful tools in annotating analytes is the fragmentation spectra produced by tandem mass spectrometry, which are a kind of "molecular fingerprint" which can be matched against databases. However, collection of unambiguous fragmentation spectra requires individually targeting analytes for acquisition. As a consequence, resources (tandem mass spectrometry scans) must be efficiently allocated in order to collect as many fragmentation spectra as possible at the highest possible quality. The goal is to target as many possible "peaks" at the correct acquisition time to maximise their "intensity" (a proxy for acquisition quality).

To address this important resource allocation problem, this thesis presents several new "fragmentation strategies". Firstly we present TopNEXt, a framework for Data-Dependent Acquisition (DDA) strategies which utilises area and intensity comparisons between LC-MS/MS runs to develop advanced DDA strategies. We show that the strategy using all of these features, Intensity Non-Overlap is highly effective and is able to acquire fragmentation spectra for an additional 10% of our set of target peaks and with an additional 20% of acquisition intensity.

We then present a "pre-scheduled" method which uses a maximum bipartite matching algorithm to plan an acquisition in advance. We extend an existing technique to map the LC-MS/MS acquisition problem to an instance of the maximum bipartite matching problem. Our extensions include extending the technique to plan multiple runs and samples as

a set, solving a weighted version of the problem to optimise acquisition times and redundantly assigning unassigned scans to improve the robustness of the method. We show that this schedule can theoretically obtain completely comprehensive coverage of a sample in a low number of injections compared to other methods. However, we also investigate the trade-off between DDA and pre-scheduled methods by testing this pre-scheduled method in a situation significantly different than the one which it has planned for (which may happen frequently in reality). In this scenario we show that it still has performance comparable to the state-of-the-art, but only with the improvements we have made to the technique. Finally, we reflect on the common elements that make our techniques successful: namely, accounting for acquisition time and quality, and judicious use of redundancy to improve their robustness.

Acknowledgements

This thesis was only possible thanks to the support of various parties. Please note that this list is not given in any order of importance, and it is probably not exhaustive.

I would like to thank my first primary supervisor, Simon Rogers, for introducing me to the world of computational metabolomics. I am grateful to you both for taking me as your student and for having taught me the importance of playing around with your data. I would like to thank Bjørn Sand Jensen for his brief stint as my second supervisor, and the useful academic guidence he provided. I would also like to thank Kevin Bryson, my *third* primary supervisor, for having been the one to stick with the project until the end. I am also grateful for your willingness to indulge me in my progression as a researcher (for example, allowing me to co-supervise undergraduate projects) and your general open-mindedness towards the project.

Thank you to Vinny Davies and Rónán Daly for your part as my secondary supervisors. I am also grateful to Vinny for always being critically-minded and good to bounce ideas off, and to Rónán for the access to Polyomics facilities and criticism of my original methodology in the matching chapter. Thank you also to Joe Wandy for having been my colleague in developing ViMMS, and having done more for my career than I myself have done. Additionally, thank you to Stefan Weidt for having run experiments on the actual instruments where necessary.

Gavin Blackburn also kindly agreed to read part of this thesis despite having no formal obligation to me. Thank you for your technical corrections relating to mass spectrometry. I found your combination of knowledge, sunny disposition and ardor for your subject to be inspiring.

Thank you to Joe (again!), Grimur and Fran, for having previously written theses which I could use as a guideline when writing my own. Thanks also to Alex, Fran (again!) and Will for having been my peers, as fellow PhD students.

Thank you also to Ciaran McCreesh and Patrick Prosser. While you had no involvement in this project, I found your thoroughness and scientific rigour during my Master's year

inspiring. I hope the standards of my work even halfway approximate your dedication to your analyses. And, thank you to Jessica Enright, for having agreed to cover one of my Annual Progression Reviews at the last minute.

Thank you to my friends, Cam, Daniel, Jude, Ivan and Lena for having continued to endure my company. Thank you also to my family for ensuring I could focus on my PhD while charging me neither rent nor board.

Finally, to those of you who suffered through the earlier drafts of my work to help me improve it, I am grateful, and I hope you at least found some small part of it interesting.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Recent technological advances have made it possible to profile biological samples with more breadth than ever before. This has led to many emerging "-omics" areas of research — for example, genomics, proteomics and metabolomics. The range of applications for these technologies is staggering. For example, newspapers widely reported the case of Joy Milne, a woman who could smell otherwise-undetectable Parkinson's disease [1]. Researchers have since putatively isolated a potential biomarker explaining this odd phenomenon, developing a non-invasive swab test for Parkinson's disease [2]. Metabolomics, as the study of small molecules involved in metabolism, gave researchers a "snapshot" of many of the biological processes in their samples, allowing them to isolate the crucial biomarker.

Another interesting application of untargeted metabolomics is in detecting "food fraud" [3, 4, 5]. Essentially, an adversary attempts to disguise a food product as another[1]. A metabolomics analysis can resolve the food authentication problem by revealing biomarkers that are informative as to the composition of the sample. Metabolomics also has a myriad of other applications, including drug discovery through identification of natural products [6, 7, 8], ecology and environmental science [9, 10], nutritional research [11, 12] and forensics and toxicology [13, 14].

However, the quality of one's analysis can only be as good as the quality of one's underlying data. In many cases samples are profiled by mass spectrometry. Liquid chromatography coupled to tandem mass spectrometry (LC-MS/MS) is a sophisticated analytical platform which allows the fragmentation of analytes and the measurement of their fragments. The fragmentation spectra collected from this procedure serve as a sort of "molecular fingerprint" which allows high-accuracy annotation of metabolites. An untargeted metabolomics experiment profiles as broadly as possible to generate new research hypotheses, so aims to collect fragmentation spectra for a very large number of potential analytes.

[1] A well-known example of food fraud is horse meat being discovered in burgers, though that was by DNA analysis rather than metabolomics.

The strength of LC-MS/MS is its generality, in that it is capable of reporting very accurate information for a very large number of unknown analytes. As a result it is used in applications requiring this strength, but equally, this also means there is an ever-present need for even broader profiling. For example, in identifying biomarkers for Parkinson's, the researchers employed a popular technique called Mummichog analysis [15]. The motivation for Mummichog analysis, in the words of the creators of the technique, is that "metabolite identification forms the bottleneck of untargeted metabolomics" and that this is a consequence of tandem mass spectrometry being "inherently low throughput". This "low throughput" is essentially that fragmentation spectra can only be collected for a limited subset of analytes per each LC-MS/MS run or experiment, slowing down overall experimental goals. Many others have identified the throughput of identification (which often relies on tandem mass spectrometry) as one of the central challenges facing untargeted metabolomics [16, 17, 18].

The question this thesis asks is whether LC-MS/MS acquisition must necessarily be low-throughput, and whether an algorithmic solution can be posed to this problem. While computing scientists have a long history with chemical analysis (Dendral [19] is often given as the earliest example of an expert system) most improvements to the data acquisition process have been in more sophisticated instrumentation and experimental protocol. These improvements have also led to more data being produced, which produces even more insurmountable data-processing challenges. Addressing these challenges has required expertise from a variety of disciplines and generates constant demand for better software tooling [16, 20, 21, 22].

While most of this previous work in LC-MS/MS data processing generally focuses on post-acquisition processing of the data, there is relatively little research on how the machine's control software processes the acquisition itself. When the instrument chooses to collect fragmentation spectra and for which ions can have a significant impact on experiment throughput. The fragmentation strategies used in practice are often severely limited compared to the potential performance an optimal strategy could theoretically reach [23]. In practice, fragmentation strategies which target (and thus "fingerprint") individual species of analyte are prone to pathological levels of redundancy. DIA (Data-Independent Acquisition) is a popular option which circumvents this problem entirely by sampling multiple analytes at once, but this creates chimeric spectra which are often difficult to disentangle [24, 25].

In this thesis we will take the position that, as an alternative to DIA, we can improve the scheduling capabilities of DDA (Data-Dependent Acquisition) and pre-scheduled methods. By controlling redundancies in the targets they collect data for and selecting more optimal targeting times, we can obtain more and higher quality fragmentation spectra. We will first introduce the TopNEXt framework, a framework for DDA methods which uses computational geometry to offer scoring heuristics for targets in experiments of multiple LC-MS/MS runs. LC-MS/MS data is discriminated into rectangular RoIs (Regions of Interest), similarly to downstream data processing but in real-time, and the scoring heuristics provided by Top-

NEXt compare the area and signal intensity of an RoI as a potential fragmentation target to previous targets to decide whether to target.

We will also introduce a novel pre-scheduled (offline planning) method which builds on an existing maximum bipartite matching technique, which was used for theoretical performance benchmarking [23]. In addition to developing an implementation which can be used in an experimental context for the first time, we have made several improvements necessary to use the method in practice. These are extending the method to multiple experiments incorporating multiple sample types, introducing weighted matching so that acquisition intensities of targets can be optimised and introducing some redundancy using leftover "filler" scans in order to make the methods more robust. For both DDA and pre-scheduled methods we will show by experiment their efficacy at selecting a greater number of unique targets for fragmentation and doing so at more optimal times. By improving this fundamental technology we hope to see improvements in the throughput of various kinds of LC-MS/MS-based molecular study (metabolomics, proteomics, exposomics, etc.) though we focus primarily on demonstrating the improvements on metabolomics data.

1.1 Thesis Statement

Untargeted LC-MS/MS experiments are constrained by their capacity to collect identifying information for all analytes of interest at sufficient quality. When targeting each analyte individually, sophisticated scheduling must be employed to ensure that the broadest possible subset of the most interesting analytes can each be targeted at the optimal acquisition time. This scheduling problem can be addressed and sample characterisation by LC-MS/MS can become more comprehensive through novel use of scoring heuristics based on the geometric properties of Regions of Interest and through the use of maximum bipartite matching algorithms for planning.

1.2 Thesis Structure

This thesis is separated into seven chapters, including this introduction.

In **Chapter 2** we introduce the context to understand the research problem, including addressing in more depth its place within a larger scope, the structure of LC-MS/MS data, the challenges of processing LC-MS/MS data and details of fragmentation strategies currently used.

In **Chapter 3** we introduce the general methodology to evaluate our new fragmentation

strategies. This methodology includes evaluation measures, how "peak-picking" was performed to decide the evaluation set and details of how experiments were conducted.

In **Chapter 4** we introduce TopNEXt, a framework for DDA (Data-Dependent Acquisition) methods which respond to data in the acquisition as it is observed. TopNEXt extends commonly-used heuristics with area-based and signal intensity based scoring. Computing TopNEXt's heuristics uses computational geometry with rectilinear polygons. Several new methods using this framework are introduced, and we show that these methods obtain greater quality and quantity of fragmentation spectra compared to state-of-the-art.

In **Chapter 5** we extend the work in Chapter 4 by testing many of its assumptions by experiment. We show that results are consistent across different parameter settings, sample orders, and sample types. We also explore how different peak-picking assumptions affect the results.

In **Chapter 6** we introduce a novel technique using maximum bipartite matching algorithms for pre-planning an acquisition. This technique applies a previous benchmarking tool using maximum bipartite algorithms [23] to the actual acquisition process. Additionally, this method is augmented by several improvements, including its extension to multiple LC-MS/MS runs and multiple sample types, weighted matching to optimise acquisition times (and hence data quality), and redundant assignment of spare scans to improve method robustness. We show how this can be used to acquire a target set more efficiently than existing methods. We also explore the extent to which real-world variability affects the results, and show that the improvements introduced are necessary for this to be a viable method.

In **Chapter 7** we present our conclusions, reflecting on the contributions of this thesis and its place in both current and future research contexts.

The evaluation techniques described in Chapter 3 supported all the primary work described in Chapters 4 and 6 but also some secondary publications [26, 25]. Chapters 4 and 5 were published as a unit in *OUP Bioinformatics* [27]. Large parts of those chapters (including text, tables and figures) were directly adapted from that publication. Chapter 6 has been submitted for publication to *BMC Bioinformatics*. Some of the implementation for the work described in Chapter 4 (particularly where it concerns SmartRoI and WeightedDEW hybridisation) was performed by Joe Wandy. Lab data collection was performed by Stefan Weidt. Otherwise, all work described within is my own.

Chapter 2

Background

In the introduction we established some of the benefits of improving LC-MS/MS data acquisition and that we are going to do so by improving the algorithmic behaviour of the LC-MS/MS fragmentation strategy. To begin outlining how we are going to achieve this goal, though, some details on the structure of LC-MS/MS data and on where the research problem fits in a broader context will be required. -omics pipelines are complex, interdisciplinary and involve many steps. As an example, a very high-level overview of a metabolomics pipeline is given in Figure 2.1. Because of the complexity of the pipeline, each step depicted has many sub-steps that will not all be discussed here (for an alternative discussion see e.g. [28]).

In this workflow, untargeted data acquisition is only one step of many. Prior to collecting data, an analytical chemist must carefully prepare both the sample and the instrument — both of these factors will influence the subset of metabolites detected [17]. Next, the actual acquisition can be either targeted or untargeted. As the name suggests, a targeted analysis has much narrower scope and often takes on a confirmatory role or is used to more precisely quantify metabolites after an untargeted experiment generates an initial hypothesis. Following this, the data is subject to multiple post-processing steps, including "peak-picking" to extract relevant parts of the data and the subsequent putative annotation of metabolite identities and quantities. With metabolite annotations established, statistical testing is performed to find significant metabolites, pathway and enrichment analysis are performed to determine biological function, and finally these conclusions are integrated with other -omics data (if genomics and metabolomics both lead to the same biological conclusion, then that conclusion is more plausible).

Although our main focus will be the data acquisition step, we will also need to use certain post-acquisition processing steps to be able to interpret the output of fragmentation strategies and evaluate them. Ordinarily, the goal of the data post-processing step is to produce high-quality metabolite annotations (i.e. labels on the likely identities of metabolites in the sample). This is itself an iterative, difficult process where annotations are graded on a four-

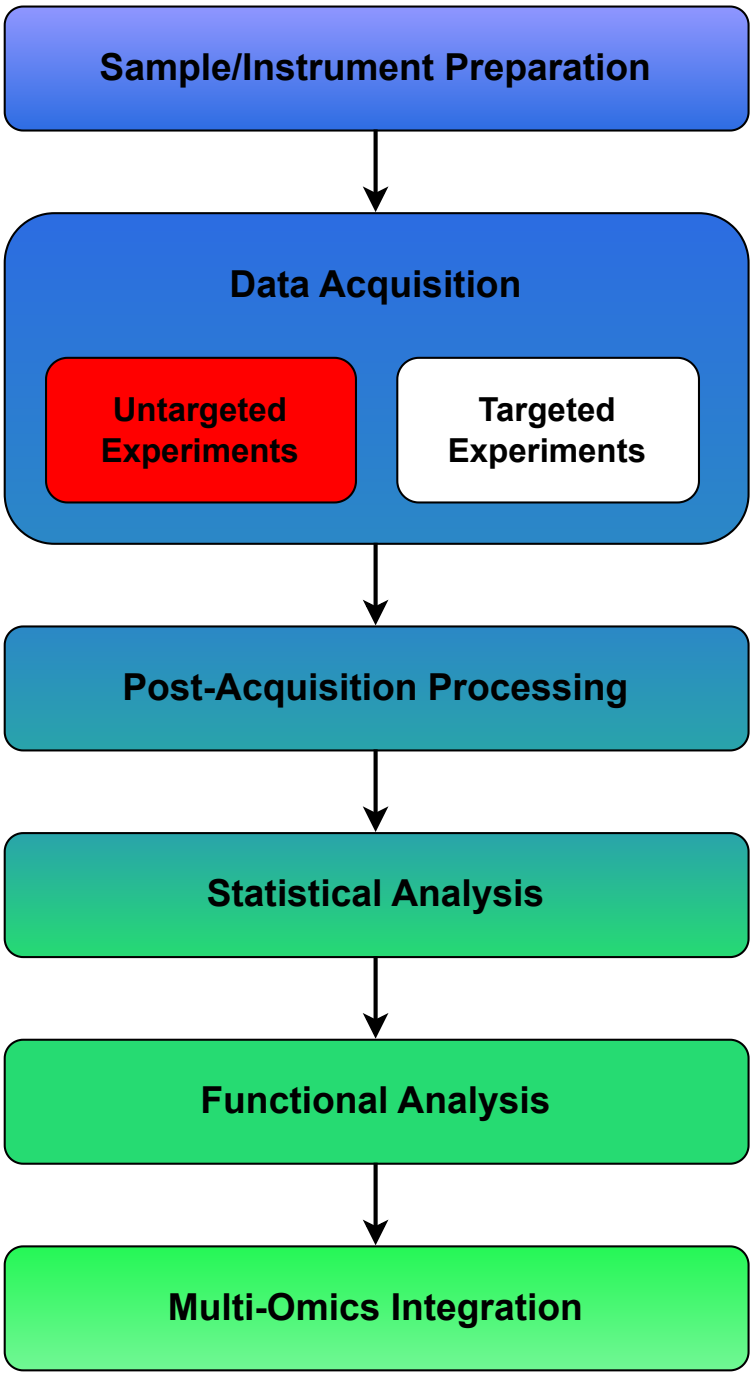

Figure 2.1: A high-level overview of a metabolomics pipeline. Acquisition strategies for untargeted experiments are the focus of this thesis, so the data acquisition step has been expanded to include them and untargeted experiments are marked in red.

point confidence scale, where 1 represents a manually verified metabolite, 2 and 3 are putative identifications and 4 is for unidentified/unclassified metabolites which can be distinguished from others [29, 30]. With a set of metabolite annotations we could then grade how many fragmentation spectra for known metabolites a given strategy obtained. Because of the various difficulties involved in metabolite annotation and our interest in the algorithmic behaviour of the strategy, we will instead be relating collected fragmentation spectra to a list of target "chromatographic peaks" chosen by a "peak-picker" (we call this the "target list"). We will explain the evaluation procedure and its motivation in more depth in Chapter 3, but peak-picking will therefore be explained in some depth in this chapter.

Finally note that our focus is on *untargeted* data acquisition. A targeted experiment restricts its scope to a fixed subset of analytes, whereas an untargeted experiment aims to profile as much of the sample as possible. Untargeted experiments are typically viewed as "hypothesis-generating", whereas targeted experiments are used to more accurately quantify analytes suspected to be important to a given hypothesis [17]. On paper the flow of this process begins with using an untargeted experiment to annotate a very large set of metabolites. Then, the results are analysed, a hypothesis generated, and a targeted experiment exclusively measures certain analytes to more accurately quantify them (i.e. obtain concentrations). The analyte concentrations are then analysed to confirm or deny the hypothesis. However, if a significant analyte is missed during the untargeted experiment, then it will not form part of the hypothesis and it may be necessary to return to the untargeted experiment. Therefore it is common in practice for this structure to be iterative, with data being laboriously passed back-and-forth and analysed between experts with different specialities and the multiple phases of data acquisition often needing to be coordinated on shared lab resources. More comprehensive data collection during untargeted experiments would streamline this cumbersome process.

In Section 2.1 we give an overview of the motivations of "-omics" technology for a non-expert. Section 2.2 describes the structure of LC-MS/MS data, starting with data generated from MS and building up to LC-MS/MS. It also describes the processing of LC-MS/MS data, including the challenges, peak-picking, and some light detail on the later processing steps from Figure 2.1. Section 2.3 describes the current landscape for fragmentation strategies. Finally, Section 2.4 describes the ViMMS simulator, a tool used to simulate LC-MS/MS experiments and to prototype fragmentation strategies, which will be used throughout the later sections of this thesis.

2.1 What is "-Omics"?

Even the most casual reader is likely passingly familiar with the study of genetics and with the DNA molecule. The classroom model of biology goes something like this: DNA encodes

proteins by the order in which four fundamental bases are arranged. The "code" in DNA is transcribed into messenger RNA, which carries it to the ribosome. The ribosome then synthesises proteins, the "workers" essential to virtually all functions of life. This is typically called the "central dogma" of molecular biology.

But what is the distinction between "genetics" and "genomics"? The "-ome" suffix has been adopted to mean something like "all members of a class considered as a whole". While genetics studies individual genes and heredity, high-throughput sequencing has allowed the collective study of the interactions of many genes, up to the entire collection of genes in an organism, called its "genome". This has been dubbed "genomics". Similarly, developments in other measurement technologies have enabled the study of other "-omes". "Proteomics" studies the "proteome", a large collection of proteins taken as a whole, "metabolomics" studies the "metabolome" i.e. the molecular participants in an organism's metabolism taken as a whole.

Then, why all these different -omics? The sequencing of the entire human genome in 2003 generated great excitement and popular belief that the secrets to the code of life would finally be cracked. It has proven more challenging in practice. Not all genes are expressed in obvious or independent ways, so it is very difficult to understand the overall biological system in which they operate. Measuring proteins or metabolites will give you information about what biological processes are actually occurring. For example, in natural products research (currently an area of active interest for clinical drug discovery) there is great interest in so-called "cryptic" genome sequences which are not expressed under laboratory conditions [31]. Many species of *Streptomyces* bacteria are estimated to have 25-50 regions which could produce metabolites, 90% are cryptic under normal laboratory conditions [32].

If you were to study only genomes, a DNA sequence will tell you the amino acid sequences of the proteins it codes, but not their structures, concentrations, or the biological context ("pathways") which among other things lead to the production of metabolites [33]. Although a protein will fold in a deterministic way given its amino acid sequence (barring misfolds) not all structures resulting from a given amino acid sequence are known and structure determines the protein's biological function. Consequently, protein structure prediction remains one of the most important problems in computational biology [34, 35]. The concentration of a protein also determines how significant its activity is (i.e. a higher quantity of a given protein means more activity). For example, with more of a given enzyme catalysing a given metabolic reaction then the rate of that reaction will increase [36]. Some proteins, too, are subject to "post-translational modification", meaning they are later changed from the sequence encoded into them by DNA, affecting their structure and function [37].

Metabolites and metabolic reactions are even less tractable by genome-only analysis (although it is for example possible to mine genomes for helper proteins with known metabolic

functions in natural products research [38, 39]). However, since metabolites are direct participants in and by-products of the metabolic reactions an organism is undergoing in its current environment, they are often considered the link between an organism's genotype and its phenotype (i.e. the actual characteristics an organism displays) [40]. Metabolite expression can change as part of the pathology of a disease [2, 41], including anxiety disorders [42, 43], or in response to nutritional deficiency [44], stress [45] or exercise [46].

Despite its utility, metabolomics has been relatively neglected compared to its genomic and proteomic cousins [47]. This can be attributed to both the DNA-centric view of cell biology [33] and the continued improvements in measurement technology which have allowed its study (which we will discuss in the following section). The emergence of metabolomics as a growing field, thanks to these improvements in measurement technology, is one of the primary factors motivating this thesis[1]. Several other -omics technologies are also approaching maturity. Lipidomics is on paper a subset of metabolomics, but chemical structures common to lipids motivates their study as their own subfield [48, 49]. Exposomics studies the effect of environmental exposure on an organism at a molecular level [50]. And many other "-omics" have been coined, too.

One further thing to note about these different classes of biomolecules is that they have different structural properties. Proteins are particularly large molecules formed of long chains of peptides (themselves composed of amino acids) which deterministically fold into a structure which gives them their function. Metabolites, conversely, are largely small molecules which will often hold only a single charge when ionised. This typically translates into different challenges and differences in analytical procedure. For example, proteins are often first digested into individual peptides, those peptides are analysed by LC-MS/MS, and once they have been (putatively) identified they are then sequenced to reconstruct the whole protein. Because metabolites are small molecules usually not composed of peptides (although in some cases they are [51]) they rarely follow this procedure and are often whole molecules when injected into LC-MS/MS. However, across known life, proteins are constructed from a set of approximately 20 amino acids, meaning these long chains essentially have a known "alphabet". The same is not generally true for metabolites meaning that metabolomics studies must grapple with this unknown.

However, while it is important context that it is not straightforward to directly translate from one "-omics" field to another, the application of LC-MS/MS is common to several fields (e.g. proteomics, metabolomics, lipidomics, exposomics) and largely operates under the same principles. Thus, although we will focus primarily on metabolomics (with occasional forays into proteomics during this background chapter) the content of this thesis likely has broad applicability to other "-omics" fields. Generalising further, this thesis focuses on algorithmic

[1] Another consequence of this emergence is that it is quite an exciting time to be involved in computational metabolomics!

resource allocation problems which would perhaps apply to other problems with a similar structure.

2.2 LC-MS/MS

One of the most important measurement technologies in analytical chemistry and hence the study of molecular "-omes" is mass spectrometry (MS). We will explain the fundamentals here, but for an alternative overview see [52]. The basic principle of mass spectrometry is to separate ions of different m/z (mass-to-charge) ratios. The first such machine was created by J.J. Thomson in 1910. Mass spectrometry was later famously applied in the Manhattan Project where machines named "calutrons" were used to separate the isotopes of uranium, and thus create the enriched uranium necessary for a fissile reaction [53] [2].

An analytical chemist aims to identify and quantify molecules in a given, unknown sample, and MS is an extremely powerful tool for doing so. Modern MS machines are considered to have three parts: molecules are given charge and become ions at an *ion source*, then are separated by mass-to-charge ratio by a *mass analyser* and finally are assigned an *intensity* value at a *detector* which is correlated to their relative abundances. In other words, ions are separated into various bins by m/z and then an indicator value correlated to their quantities is measured. Theoretically, this means if you know the charges held by each ion, you can work out quantities (and hence concentrations) of different molecules of different masses originally injected into the MS instrument — although in reality this is a technical process requiring calibration curves. A crude layperson's metaphor is that (with current levels of technology) an MS instrument is a highly precise weighing machine capable of separating masses on the order of 10^{-3} Da. An illustration of a "mass spectrum" that would be produced by such an instrument can be seen in Figure 2.2.

Achieving this separation by m/z can be achieved by several different techniques. For example, time-of-flight mass spectrometry measures the time for an ion to travel a known distance. An electric field of known, fixed strength is used to accelerate each ion, so that all ions of a given charge are accelerated with equal force. This means that ions with more mass (which require more energy to reach the same speeds) travel to the detector more slowly i.e. ions will be sorted in ascending order of m/z. Quadrupole mass analysers, by contrast, use two pairs of parallel electrically charged rods — the voltage applied to the rods will alter the trajectory of ions passing through. These alterations in trajectory can be controlled by changing the amount of voltage applied, allowing the entire m/z range to be scanned. The importance of this from a data analysis perspective is that different instrument types have different prop-

[2] Frank Oppenheimer, the younger and lesser-known brother of Robert Oppenheimer, worked on these calutrons, which he called "racetracks".

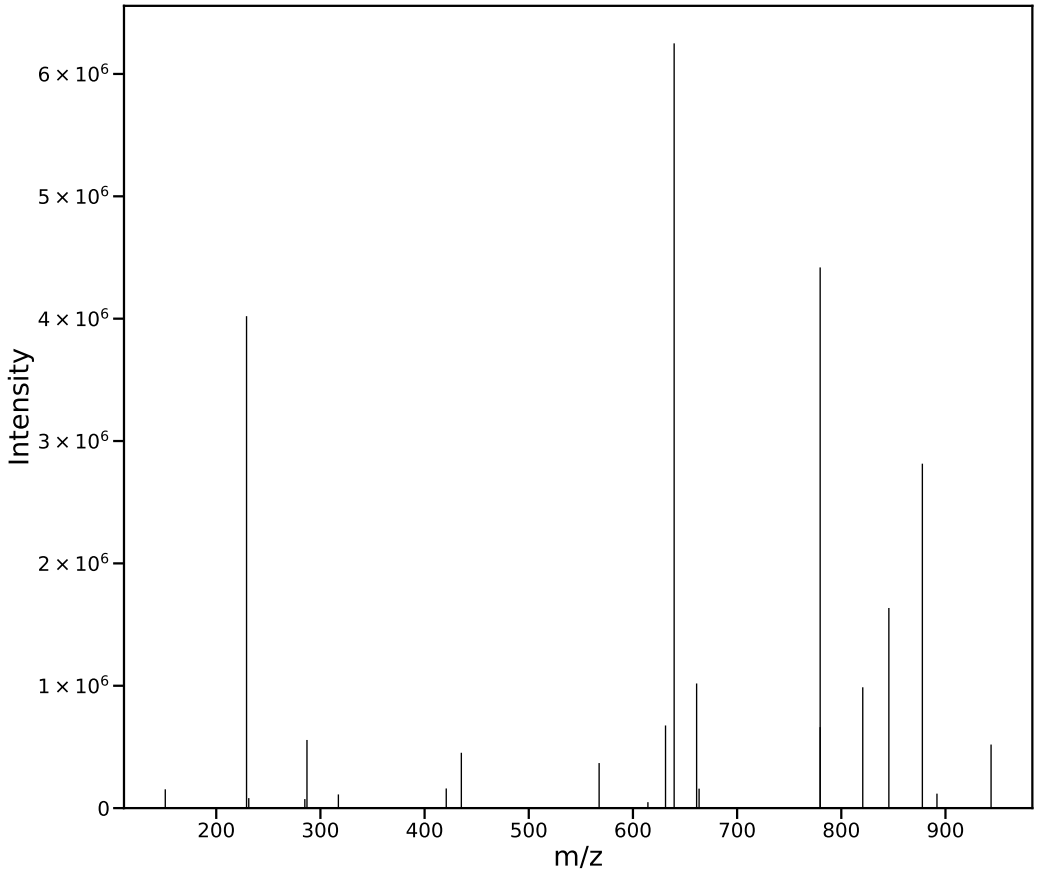

Figure 2.2: An example of a mass spectrum. Ions of different m/z (mass-to-charge ratio) produce intensity readings when deflected into a detector. The height of the peak (i.e. the intensity) is linked to the abundance of ions of that m/z.

erties in terms of resolution (ability to separate ions of close mass) accuracy (proximity of the reported m/z to the actual value) and so on [52].

There are also alternatives to mass spectrometry for compound identification. The most common is NMR (nuclear magnetic resonance) spectroscopy [3]. NMR is typically viewed as complementary to MS, as both have different strengths and weaknesses. While MS is highly sensitive (it can detect analytes at lower quantities), NMR spectroscopy is more reproducible, provides easily interpretable quantitative information and is useful for structural elucidation [7]. It will have no significant presence in this thesis, however.

[3]NMR is also used in medical MRI (magnetic resonance imaging) scans.

2.2.1 LC-MS

MS technology alone is not sufficient for the analysis of -omics samples. An organism's entire metabolome or proteome can be incredibly complex and contain thousands or tens of thousands of molecules. It is plausible that two would share the same m/z: for example, structural isomers, which are made of all the same atoms but are arranged in a different configuration. But in measuring these with direct injection into the MS, the peak on the mass spectrum would only tell us the sum of their abundances. Additionally, ion suppression effects occur when an analyte reduces the ionisation efficiency of others, perhaps by competing for the ionisation capacity of the ion source. If they cannot be ionised then they naturally cannot be analysed by MS, so relevant signals may not appear due to these ion suppression effects. Returning to the previous metaphor, imagine you are conducting a study measuring distributions of human weight by some orthogonal property, e.g. height. If two people of different heights stepped on the same scale, you would only know the sum of their heights — you would hope they would queue in an orderly way in order to be measured separately. You would also hope that they would not all pack into the same weighing room and prevent others from entering.

We therefore need to pair MS with another separation technology, which will separate analytes by chemical properties orthogonal to mass. We will focus on applications using liquid chromatography (LC). Gas chromatography (GC) and gel electrophoresis are often considered to be complementary techniques to LC [54, 55], but we will not be considering them any further.

It is not trivial to interface LC and MS, however. Essentially, modern chromatography uses high pressure pumps to separate analytes in solution, but a mass spectrometer favours conditions of high vacuum to prevent collisions between analytes and ambient molecules. As a result, various kinds of eclectic interface were built between LC and MS setups [56, 57]. But this ad-hoc competition effectively ended with the invention of atmospheric pressure ionisation methods, the most notable of which is electrospray ionisation (ESI), for which John Fenn was awarded a portion of the 2002 Nobel Prize in Chemistry [58]. In ESI a high voltage is applied to the sample solvent as it is introduced by needle and made to evaporate, causing droplets to disperse into a fine spray as they reach the limit of charge they can bear [52]. This spray is then introduced into the mass spectrometer. ESI also has the benefit of being useful for the analysis of large biological molecules, e.g. proteins, as it often produces multiply-charged ions. This compresses m/z values (as charge is the denominator here) and so increases the effective mass range of the analysis [59].

With LC-MS, analytes are generated into the MS as a function of RT (retention time) and the MS repeatedly scans to survey this population, producing one mass spectrum similar to Figure 2.2 per scan. The data is therefore three-dimensional with each value being located

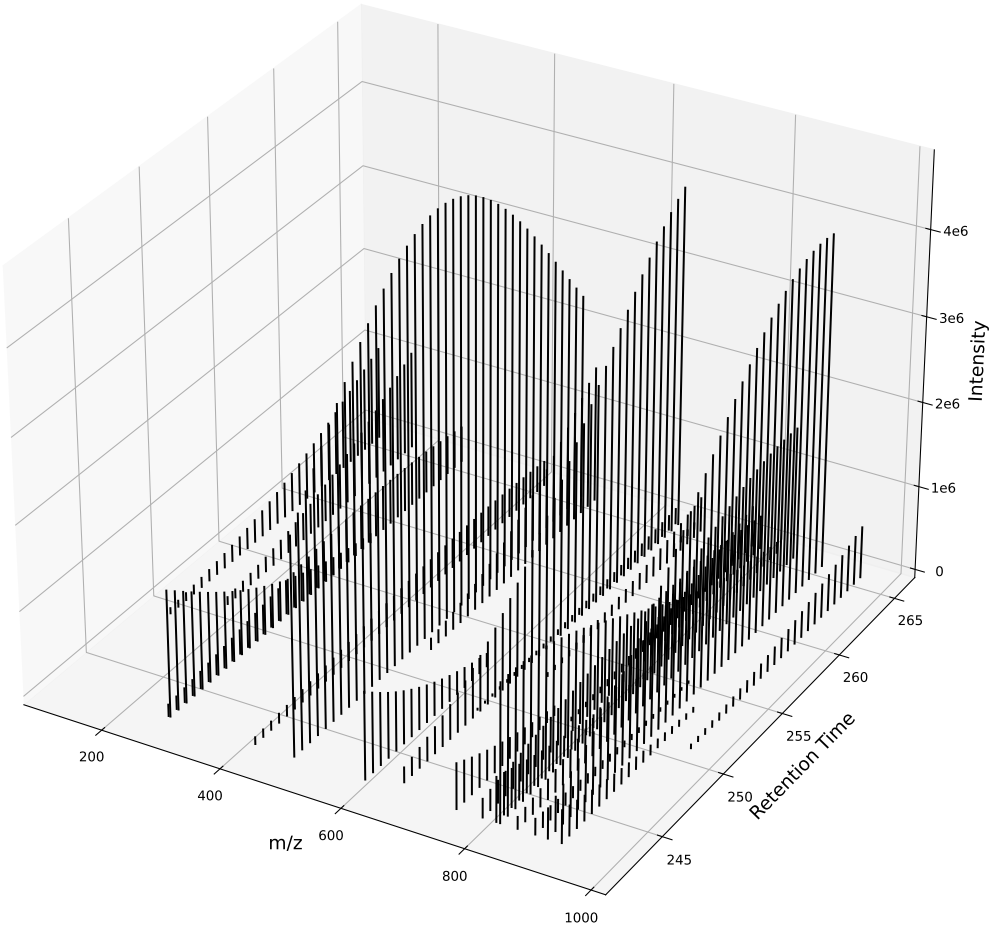

Figure 2.3: An example of individual mass spectra being collected over time via LC-MS. Each scan (one mass spectrum) is recorded at a single instant in RT. Over enough scans analytes produce approximately Gaussian-like shapes on the intensity axis i.e. chromatographic peaks.

in $(rt, m/z, intensity)$ space. The mechanics of the chromatographic column cause ions to spread out across RT in a Gaussian-like distribution: consequently the chromatographic trace across RT will frequently have a Gaussian-like trace in the intensity dimension (which is related to abundance). These Gaussian-like traces are known as *chromatographic peaks*. An example can be seen in Figure 2.3.

2.2.2 Tandem Mass Spectrometry (MS/MS)

So far we have addressed how we can measure abundances of different masses of ion, but not how we can assign an identity to those analytes. Of course, the mass and RT of an ion are of some use in identifying the chemical species, and this may be sufficient given context-

specific information. A specialised assay testing for a specific reactant may also be used to confirm a suspected identity. But this is easier when one already has a clear hypothesis in mind. In the early stages of a metabolomics analysis, one may need to generate hypotheses by filtering through the hundreds or thousands of metabolites present, many of which may not be so easy to test for. And it is not always possible to discern a chemical formula, or structure, from mass and RT alone.

Another technology which helps address this is *tandem mass spectrometry*, or MS/MS. This technique dates from the 1960s, is considered to have first been formally described in 1966 [53, 60], and allows structural information to be obtained for analytes. In MS/MS we first capture an m/z range of ions from a given scan and feed them into further instrumentation — the m/z range is known as the *isolation window*. The ions are filtered first into a collision cell, which breaks them into fragments (for example by accelerating them into a neutral gas) and then into a second mass analyser which (as before) deflects the fragments by m/z and registers them with the detector. This scan gives us a mass spectrum of the fragments, known as a *fragmentation spectrum*.

Fragmentation information is very important for elucidating chemical structure (in the introduction we referred to fragmentation spectra as "molecular fingerprints"). For example, the initial interest in LC-MS was limited because the atmospheric pressure ionisation methods used in LC-MS are "soft" ionisation methods, where a "hard" ionisation method causes analytes to fragment as they are ionised (in-source fragmentation). Coupling LC-MS to MS/MS was therefore a necessity for analytical chemistry [61].

The reason fragmentation spectra are so rich in structural information is that different bonds within a molecule have different energy requirements to break. Thus, the way in which an ion species breaks in the collision cell partially depends on its structure. Fragmenting a large number of individual ions produces a distribution of fragments of different mass characteristic of the internal structure. For example, imagine an arbitrary chemical structure in your head, then imagine it being broken into two pieces in the middle as compared to being broken at one of the sides (and then some ions will fragment further) giving pieces of different masses [4]. Note that only the pieces remaining charged will be measurable by the second mass analyser, however.

With both liquid chromatography and tandem mass spectrometry (LC-MS/MS) we now have two types of scan distributed along RT. A survey scan, where the first mass analyser produces a mass spectrum of all intact ions currently eluting from the LC, and a fragmentation scan, where the second mass analyser produces a fragmentation spectrum giving the $(m/z, intensity)$ pairs of the fragments. These types of scan are also known as MS1 and

[4] The previous weighing room metaphor somewhat breaks down here because you are unlikely to be breaking your study participants into pieces in order to study their configuration.

MS2 scans respectively. It is possible to run several MS1s in a row, and each MS1 may be followed by several MS2s. The MS1 scan preceding an MS2 is known as the *precursor scan*, and any intact ions covered by the isolation window are known as *precursor ions*. Both types of scan are useful: MS1 scans provide (alongside other identifying information like elution RT) relative quantitative information for the whole mass range scanned, and each MS2 provides a fragmentation spectrum which can be used for compound annotation. Note that these different levels of scan take different average amounts of time to process (based on how long the machine takes to collect sufficient ions). Average times may also differ when switching level between MS1 and MS2 and vice versa due to parallelisation in the instrumentation.

Throughout this thesis the term **scan** will refer to either an MS1 or MS2 scan. A(n LC-MS/MS) **run** will be one complete analysis by LC-MS/MS, which produces one .mzML file. This corresponds to one injection of the sample on the instrument, but we call this a "run" to avoid confusion with simulated experiments, where nothing is physically injected. **Sample type** will refer to different biological origins between various samples (e.g. different brands of beer). A reference to a "sample" will either be to sample type (e.g. in the term "multi-sample") or the physical sample collected itself. A **fullscan** run is a run where only MS1 scans have been used [5].

The balance of MS1 and MS2 scans, and the assignment of isolation windows for the MS2 scans, is the responsibility of a **fragmentation strategy**. Given a limited amount of time in which to assign scans (and with some variability in their times) we must give them an assignment that will result in the richest data. This is an information processing challenge, so it is amenable to analysis by Computing Science. The topic of this thesis is improving fragmentation strategies in the context of an untargeted experiment (an environment with many unknowns) so detailed background and review will be given later in Section 2.3.

2.2.3 Challenges of LC-MS/MS Data

Although the high-level principles explained so far are relatively simple, mass spectrometry data comes with many notorious technical challenges — many artifacts can be found in the data. There are many sources of potential contaminants [62, 63] and many analytes which may not be of interest. In addition to chromatographic peaks, there are mass peaks along the m/z dimension which are typically collapsed to a single aggregate value in a process known as "centroiding" [16]. Intensity values for the same analyte differ between LC-MS/MS setups [16], and intensity values may also contain "spike noise" with a sudden massive, unrepresentative increase [64]. The data itself is very densely packed with readings (millions of individual points per run) and chromatographic peaks must be putatively identified from

[5] The language used has evolved, but MS1s were once simply called "scans" so to run in fullscan mode is to run only MS1s.

it before analysis of potential metabolites can be done. Peaks may drift in RT between each LC-MS/MS run as a function of the chemical properties of the LC column [65], and may entirely drop out of some runs [66].

Each analyte may also produce multiple chromatographic peaks, for example from multiply-charged ions, isotopes and adducts. The most obvious example is multiply-charged ions — if two types of ion have the same mass but one is singly-charged and the other doubly-charged, then the latter will have half the m/z value of the former. It is common in metabolomics to assume that all molecules are singly-charged [67] but in proteomics charge-state filtering is an important part of instrument processing [68].

Isotopes contain differing numbers of neutrons in the atomic nucleus. Each neutron has a mass of approximately 1 Da — so for a singly-charged ion, additional traces appear separated by 1 unit in m/z space. Natural abundances of carbon in nature are approximately 98.89 % for $^{12}\mathrm{C}$, 1.108% for $^{13}\mathrm{C}$ and 10^{-10}% for $^{14}\mathrm{C}$. For sample ions containing carbon you would expect to see abundances (and hence peaks) in this ratio. In proteomics in particular it is frequently assumed that isotopic peaks will appear and be distributed in an "averagine distribution" [69]. Because an unknown protein will have unknown numbers of each amino acid (and hence potentially isotopic atoms) "averagine" is a hypothetical amino acid where atom counts were averaged according to the statistical distribution of amino acids in a protein database. This then allows calculation of an average isotopic pattern from the atomic isotoping distributions. As an example of this being used in practice, MaxQuantLive (an example of real-time MS control software implementing a fragmentation strategy) will only collect fragmentation spectra for peaks observed to have at least two isotopes which fit an averagine distribution [68].

Similarly, sample molecules may bond with others when being ionised in the source, forming "adducts" [70]. These product ions may differ in mass by, for example, 22 Da when comparing a hydrogen adduct to a sodium adduct, which will consequently produce several chromatographic peaks. Of course, because isotoping and adduct patterns are characteristic, they can be useful in annotation [71, 72], but it is not likely to be useful to collect fragmentation spectra for each in every instance.

Finally, fragmentation spectra also come with their own challenges. As we mentioned in Section 2.2.2, it is a challenge to obtain fragmentation spectra for as many unique analytes as possible, and this challenge increases with sample complexity. Also, fragmentation patterns are generally only reproducible given sufficiently abundant enough precursor ions for fragmentation, i.e. low-intensity ions will have a low-quality fragmentation pattern. In some cases two different molecules may also have highly similar fragmentation patterns (e.g. some isomers) and must therefore be distinguished by some other means (e.g. RT). The exact fragmentation spectrum produced by a given molecule also depends on the collision energy used

to break its chemical bonds — higher energies will cause further fragmentation — so comparisons must use similar collision energies. In this thesis targeting as many unique analytes as possible, and targeting them at high precursor intensities to obtain high-quality fragmentation spectra will be of particular interest.

2.2.4 Peak-Picking

The process of identifying plausible chromatographic peaks in the data has several names, including "feature finding", "peak detection" and "peak-picking". Well-formed chromatographic peaks have a Gaussian-like shape, so initially core peak-picking algorithms used a simple Gaussian curve fitter [73]. Currently the most well-established mass spectrometry software packages XCMS [74] and MZMine [75] use continuous wavelet transforms for peak-picking. The continuous wavelet transform is a technique similar to the Fourier transform in that it can be used to decompose multiple overlapping signals into their constituents [76]. In the context of peak-picking, this allows separating the combined signals of multiple overlapping peaks and noise signals [77].

The first example of this concept being deployed for peak-picking is $centwave$ [77], which is still the primary algorithm for XCMS. In $centwave$, we must first build RoIs (Regions of Interest). Each RoI is a bounding-box for points in $(rt, m/z)$ space — the RT bound is determined by the points belonging to that RoI, and the m/z bound is a fixed tolerance from the mean of the points. Each MS1 scan is processed in order of RT, and if a point would fall within a "live" RoI's m/z tolerance then it is appended to the first such RoI. If no appropriate RoI exists for that point, then a new RoI is created. A "live" RoI is one which had a point appended in the previous scan. After each scan is fully processed, any RoI which did not have a point appended becomes "dead" and its m/z range is now open for new RoIs to form. Each RoI is also adjusted to be re-centred on the mean m/z of its constituent points. Once RoIs are constructed, the continuous wavelet transform algorithm is used to decompose the RoIs into smaller bounding boxes containing peaks. An example of how RoIs might segment the data is given in Figure 2.4.

A more recent algorithm is ADAP [78, 79], which has supplanted $centwave$ as the primary peak-picking algorithm of MZMine. It operates on the same basic principles but with some tweaks. For example, rather than building RoIs from the MS1 scans in order of RT, points are sorted in descending order of intensity and RoIs are built starting from the highest intensity points first. The highest intensity points are often most relevant so this should lead to more coherent RoIs.

In addition to the actual detection of peaks, it is also necessary to align peaks between runs

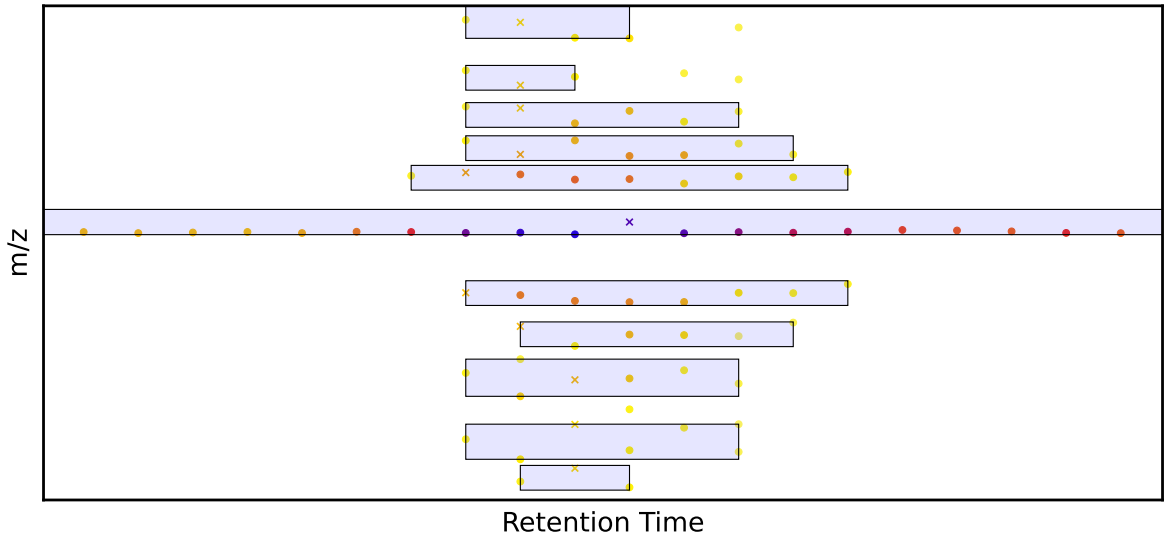

Figure 2.4: An example of how RoIs might be drawn around a small slice of LC-MS/MS data, using the *centwave* algorithm. All points are from MS1 scans - crosses are the precursors of an MS2. The colour of the points represents intensity, going from yellow to red to blue with higher intensity. Light-blue rectangles are RoIs.

and sample types and determine their correspondence [6]. Over long-running experiments in particular, the chemistry of the chromatographic column changes, causing drifts in retention time. To identify common peaks between runs this drift first needs to be corrected (which is itself an open research problem) [80, 81, 82, 83, 74]. After this, different instances of peaks across runs must be identified. Because the corresponding peaks should (hopefully) now have similar retention times, this is often done by simple heuristics like comparing their RTs, RoIs and intensities [84] — for example, the MZMine "join aligner" greedily assigns the closest peak in $(rt, m/z)$ space from a given run to an aligned peakset [75]. The user specifies maximum tolerances for both RT and m/z and the weighting between the two values when considering the distance (as they lie on very different scales). Another common (but not universal) processing step is "gap-filling", where potentially missing peaks are imputed [85]. It is also common to group isotopes and adducts. Because of drift, peak dropout and other sources of variation, it is common to see peaks picked vary per different observations of the same sample type, and to increase in number as more observations are aligned (we will see a demonstration of this in Chapter 6).

MS-DIAL [86] is another established software package for peak identification (originally developed for analysis of Data Independent Acquisition data, which will be covered in Section 2.3.2) but we will primarily focus on XCMS and MZMine. More recent methods for peak-picking have also been published — for example, several methods which follow the

[6] The XCMS community seems to use the term "alignment" strictly for correcting RT drift and "peak correspondence" for identifying which peaks correspond run-to-run. The MZMine community typically identifies both as a single step called "alignment".

same pattern but use deep learning image recognition to classify peaks [87, 88, 89]. The $asari$ package [90] takes a significantly different approach, by reordering some of the steps. Rather than trying to find precise $(rt, m/z)$ bounding boxes for peaks prior to alignment, $asari$ instead tries to separate a collection of runs into "mass tracks" which are essentially 1D intervals on the m/z dimension. Only once they have been aligned across the runs are those mass tracks "sliced" into RT bounds for individual peaks. This greatly simplifies the alignment process, which is quite temperamental in XCMS and MZMine. However, all of these methods (other than MS-DIAL) are too new to have seen widespread adoption or community consensus. We will primarily be using XCMS and MZMine for processing LC-MS/MS data in this thesis.

2.2.5 Compound Annotation

Once we have performed our untargeted LC-MS/MS experiment the next challenge is to derive molecular annotations from the fragmentation spectra. Although these later processing steps are not the focus of this thesis, some brief context may be useful later when considering our methodology for evaluating fragmentation strategy performance. This problem has a long history within Computing Science, with the Dendral [19] software often being cited as the first example of the "expert system" programs which later drove a second AI Spring [91]. Expert systems are programs that used rules to emulate the knowledge of experts — in Dendral's case, some of the knowledge of professional mass spectrometrists in identifying chemical structures from mass spectra. Given a mass spectrum and some task-specific constraints, Dendral generates chemical structures that could have produced the spectrum, pruning the search space according to the constraints it was given. It then used a knowledge-base of mass spectrometry (a machine learning component generates some of these rules from known examples) to identify the most plausible of these structures.

This is a form of *de novo* structural elucidation, and is a useful strategy when dealing with biomolecules for which reference data has not been collected. But it is more typical to search reference databases for known analytes. The standard metrics for comparing spectra pairwise are the cosine score and modified cosine score. This technique (well-known among data scientists) treats each mass spectrum as a vector and computes the angle between them. This metric is convenient because it only considers the direction of the vectors and not their magnitude: essentially it discards the scaling on the intensity (which varies per LC-MS/MS setup) but maintains the distribution of the fragments.

However, recent work has also attempted to improve mass spectral similarity comparisons by using entropy-based scores [92] and with methods like Spec2Vec, which leverages ideas from text-processing to create lower-dimensional representations of spectra [93]. In metabolomics, such similarity measures are used to make direct comparisons to databases such as the Hu-

man Metabolome Database [94], MassBank [95], the NIST mass spectral library [96], the GNPS mass spectral libraries [97] or METLIN [98]. In proteomics, the work of peptide sequencing may be performed efficiently by software such as MASCOT [99] and Comet [100]. Additionally, the MetAssign software [101] takes a Bayesian approach to metabolomics annotations — a prior probability is assigned to each potential match given the accuracy of a mass match, then the posterior probability is updated using the presence of characteristic co-eluting peaks.

However, spectral databases are incomplete [102, 103, 48, 104] and queries are prone to returning false positives [17, 92, 104, 105, 106, 71]. Because this sort of database searching is so useful in putative identification, it would be preferable if the process of constructing a high-quality database could be eased. An improved data acquisition process (like the ones we will introduce in this thesis) would also help to build these databases by collecting spectra for more metabolites and by collecting them at higher quality (to avoid false positives).

Partially because of the difficulties we have mentioned, metabolite annotation is regarded as highly laborious and given the vastness of chemical space it is unlikely we will ever fully manually explore it and create a truly comprehensive database [102]. A useful tool in organising the known chemical space is "molecular networking". Essentially, a given reference library of chemicals is organised as a graph in mass spectral space with a defined distance metric. This allows discovery of families of related chemicals and the propagation of identifying information based on spectral data alone. The original molecular networking approach used only MS2 information [107], but more recently "Feature-based Molecular Networking" also incorporates MS1 information [108].

A related approach is MS2LDA [109], which uses topic modelling (taking inspiration from the use of machine learning in text processing) to identify substructural features in mass spectra. Another popular use of machine learning is to essentially try to solve the same problem as Dendral: that is, map mass spectra to chemical structures. For example, CSI:FingerID [103] uses an ensemble of support vector machines trained on individual molecular properties and MassGenie [110] uses transformer-based neural networks for this mapping. There is also some interest in solving the inverse problem (mapping structures to predicted spectra) which is typically considered to be more tractable. For example, CFM-ID [48] first combinatorially generates a fragmentation graph containing all possible fragmentations for the given structure, then uses its machine learning component to assign probabilities to each possible transition. MetFrag [111] is another example with a similar approach. In both cases, this can be used for comparison with databases of known structural information [104, 48, 112] as a complement to databases of known spectral information. A good overview of methods for metabolite annotation can be found in [102]. However, the important thing to draw from this section is that compound annotation in complex biological mixtures, particularly metabolite annotation, remains a challenging and unsolved problem.

2.2.6 Further Steps

The remaining steps are of limited relevance to this thesis, so we will only cover them briefly. Once annotation is performed, quantities must be assigned to metabolites. Because LC-MS/MS is not absolutely quantitative like NMR, this requires the use of calibration curves [71]. Analysis by targeted experiment is also often useful in interpreting quantities [17, 83]. Once metabolites (or proteins) have been reduced to a list of identities and concentrations, their significance can be analysed with standard statistical techniques [28]. This may provide potential biomarkers.

Beyond biomarkers, an analyst may attempt to understand biological function through pathway analysis. A biological pathway is a directed graph showing the flow of a biological process. In practice, this can mean searching metabolites against lists of known biological processes to see if there is an unusually high concentration corresponding to a specific pathway i.e. enrichment analysis. The Mummichog technique we described in the introduction is a form of pathway analysis, and takes the position that because metabolite annotation is difficult, we can instead try to link correlated features in the MS1 data directly to known pathways [15]. The products of biological pathways with a known starting point can also be predicted using Biotransformer [113]. A good review of pathway analysis methods can be found in [114].

The final step as presented in Figure 2.1 is integration with other -omics data. Because each -omics provides a slightly different picture of the underlying biological processes, there is great enthusiasm for more holistic multi-omics methodology [28, 8, 115, 116, 117, 118].

2.3 Fragmentation Strategies

As we mentioned when we introduced fragmentation strategies in Section 2.2.2, an LC-MS/MS fragmentation strategy is the algorithm controlling the choice of MS1 and MS2 scans for each RT, and the m/z range of isolation windows for MS2 scans. The choice of fragmentation strategy is critical to maximising acquisition quality in an untargeted experiment, so this thesis aims to improve on existing fragmentation strategies.

Computers have been used for control of mass spectrometry instruments coupled to a separation technique since at least 1967 [119]. Then the use of the computer was limited to the processing and recording of scan data and its offline interpretation [120]. The real history of untargeted fragmentation strategies begins in 1994, when a simple program was employed to heuristically select peaks for fragmentation [121]. In the lead up to 1994, Finnigan Corporation (later acquired and renamed by Thermo Fisher Scientific) had released a series of mass spectrometers controlled by integrated computers. Although the premise of computer-control

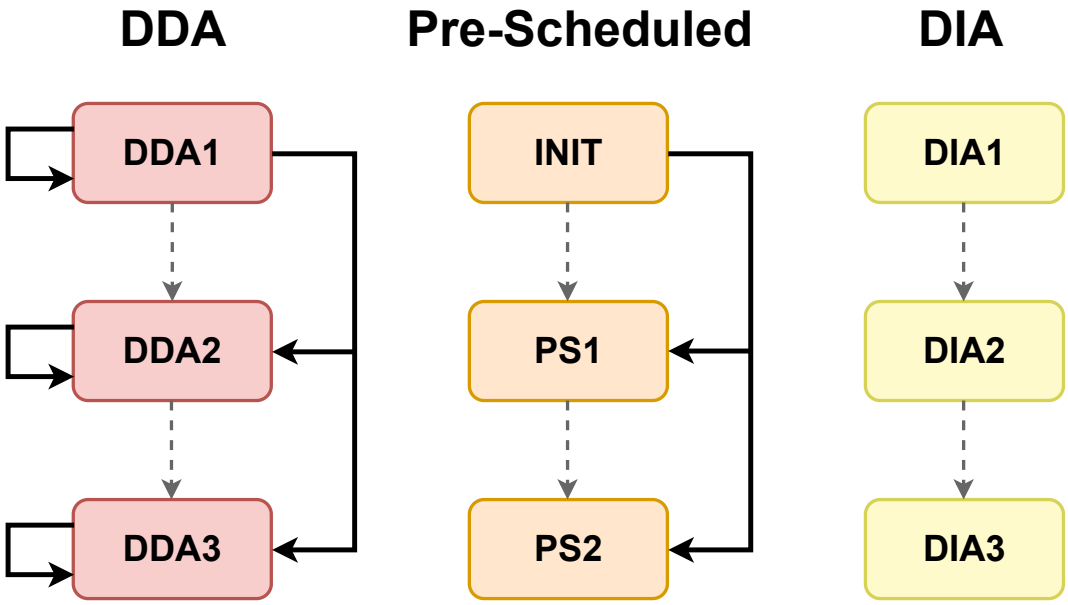

Figure 2.5: Diagram showing the flow of information in different families of acquisition strategy. The dotted grey arrows show the order of runs, and the bold black arrows show how information is used in subsequent scans. DDA has a self-loop representing the use of scan-level information. Both DDA and pre-scheduling have information flowing between samples. DIA has neither because the schedule is set completely in advance.

was not new, these instruments came bundled with an "Instrument Control Language" (ICL) which allowed programmable control of the instrument.

This development quickly led to the creation of a strategy to follow each MS1 scan with four MS2 scans, with each MS2 scan targeting one of the four highest intensities in the precursor MS1 scan. Despite its age, this strategy remains in widespread use today — it is now known as "TopN". The N refers to the maximum number of MS2 scans to follow each MS1 scan with (so the original example would be called Top4).

TopN is the archetypical example of a **DDA** (Dataset-Dependent Acquisition) strategy [7]. DDA strategies respond in real-time to the data they observe and attempt to target individual chromatographic peaks[8] with an MS2 scan using a narrow isolation window. However, because of their reliance on simple heuristics like TopN, DDA methods often engage in inefficient and redundant behaviour.

In response to this known issue, a recent trend is towards **DIA** (Dataset-Independent Acquisition) methods, which have a completely different philosophy. In DIA a schedule of very wide isolation windows is set in advance and the instrument follows this schedule exactly.

[7] Many authors refer to TopN as "DDA".

[8] In the context of fragmentation strategy targeting, we will often refer to chromatographic peaks as just "peaks". "Peaks" in individual MS1 scans are referred to as "points" to avoid ambiguity.

This produces hybrid spectra which regularly sample fragments from eluting analytes. It is therefore possible to have each measurable analyte be observed in at least one DIA fragmentation spectrum. However, because these signals are mixed together, they are not so easily analysed. DIA schemes forward the difficulties of data acquisition to the analysis step, where signals must be deconvolved.

A third approach is to target individual peaks with individual MS2 scans with narrow isolation windows (like DDA) but to do so following a pre-determined schedule (like DIA). Locations to target are decided by the analysis of some representative data, rather than being done in real-time as data appears. As a consequence, these "pre-scheduled" methods can perform more compute-intensive processing and incorporate knowledge of peaks that would occur beyond the current RT. However, they also risk data collection going awry due to significant differences between their representative data and the actual acquisition — DDA does not have this issue due to the real-time feedback loop. An illustration of the differences in flows of information in these three fragmentation strategies is shown in Figure 2.5.

These approaches are quite different, so we need some basis on which to compare and analyse them. A well-designed fragmentation strategy should satisfy (at least) three primary criteria:

- **Comprehensiveness**: fragmentation spectra should be obtained for as many unique analytes as possible.
- **Interpretability**: fragmentation spectra should be of sufficient quality that any relevant chemical information can be inferred from them.
- **Robustness**: the strategy should still function under conditions of noisy data.

Each of these families of approach has a weakness in one of these criteria. In DDA methods, analytes compete for limited MS2 scans and the methods lack the ability to plan ahead, thus DDA data collection is not particularly comprehensive. DIA methods create chimeric spectra and thus a complex deconvolution problem which limits interpretability. Pre-scheduled methods struggle when their plan differs significantly from reality and thus lack robustness. However, pre-scheduled methods are also far from maturity in terms of their planning capabilities, so their comprehensiveness can also be improved.

DIA has largely forwarded the difficulty of the problem onto deconvolution, an orthogonal problem to scheduling MS1/MS2 scans for DDA and pre-scheduled strategies, so it will not have much presence in the later parts of this thesis. We will instead focus on improving the comprehensiveness of DDA and pre-scheduled methods and comparing the trade-off in robustness.

2.3.1 DDA

TopN implementations (which often come bundled with vendor control software) are typically more complicated than the simple description from Section 2.3. They (including the original 1994 program) also contain an intensity threshold parameter, and precursors under such a threshold cannot be selected for fragmentation. If a duty cycle (the pattern of an MS1 followed by N MS2s) contains insufficient targets above the threshold then it will only schedule as many MS2s as there are reachable targets and terminate early.

Because it chooses to greedily fragment high-intensity signals without remembering its behaviour in previous scans, TopN tends to refragment the same high-intensity peaks repeatedly and so struggles to obtain the broad sample coverage necessary for an untargeted -omics experiment. To address this, vendor control software also allows TopN to be augmented by exclusion/inclusion windows. An exclusion window tells the method to *avoid* a region (usually entirely) and an inclusion window tells it to *prioritise* the region (often overriding whatever other task it could perform). These are specified as 2D intervals in $(rt, m/z)$ space. Standard implementations also commonly use Dynamic Exclusion Windows **(DEWs)** which are created dynamically every time an MS2 scan is scheduled. A DEW is created at the RT of the MS2 scan and shares the m/z centre of its isolation window. Its dimensions in RT and m/z are set by fixed global parameters. It forbids another isolation window from being centred within its bounds, so essentially it prevents refragmentation of the same m/z value for some duration (equal for all DEWs). Of course, each peak will be fragmented again as soon as the DEW expires, so TopN will still be prone (to a lesser extent) to refragmenting dominant peaks.

Even if DEW RT bounds were set individually it is not simple to set them in such a way that long, high-intensity peaks are not wastefully refragmented, and that short, low-intensity peaks have an opportunity to be fragmented close to their apex intensity.**WeightedDEW** and **SmartRoI** extend the DEW concept in an attempt to address this weakness [23].

WeightedDEW, rather than having a DEW with a single binary exclusion function, splits the DEW into two halves, which allow their lengths to be specified independently by the user. If a point would fall into the first half, then the same binary exclusion as for the regular DEW is applied. The second applies a linear weighting to the log intensity of the point scaling with the distance travelled along the second half. The weight scales from 0 at the end of the first half and beginning of the second, to 1 at the end of the second half. In other words, this DEW a piecewise function where RT values below a certain threshold are completely excluded, and RT values below a second threshold are linearly weighted by the proportion of the distance between the two thresholds. This gives smoother criteria than the binary exclusion of the DEW and thus allows more aggressive parameter settings.

SmartRoI, on the other hand, uses real-time RoI tracking (using a custom implementation of

the $centwave$ RoI-builder described in Section 2.2.4) to track peak-like objects in real-time. Refragmentation is forbidden within the same RoI unless:

- Peak intensity has risen from the last fragmentation intensity by more than a parameter value α.
- *Or* peak intensity has fallen from the last fragmentation intensity by more than a parameter value β.

These criteria are intended to distinguish when a peak has ended and another has begun at nearby m/z values. The allowance for peak intensity rising also allows peaks to be refragmented at more optimal intensity values. Thus these criteria guide methods towards fragmenting each RoI (as a stand-in for peaks) once each, near the apex intensity. The work introducing these methods found that both of these methods could be used to improve the number of peaks independently chosen by a peak-picking tool which also had a fragmentation spectrum associated with them (i.e. a measure of coverage).

Another weakness of this TopN approach is that the DEW will only exclude refragmentations during the same LC-MS/MS run. If given the same LC-MS/MS data for multiple runs in a row, TopN will follow the same course of behaviours each time. In the real world, variations between runs are responsible for making TopN take different courses of action and thus acquiring fragmentation spectra for more analytes (though this also means its behaviour is unpredictable and difficult to reproduce). Often sample stochasticity is relied on to ensure TopN can obtain sufficient coverage. A simple approach to extend TopN to avoid regions fragmented in previous runs is for it to "remember" the DEWs and include them as exclusion windows on future iterations. We will call this method **TopN Exclusion**[9]. An early example of this technique was used to more deeply profile the proteome of human embryonic stem cells compared to a conventional TopN analysis [122]. The "IE-omics" software provides a semi-automatic implementation of this procedure, and it was demonstrated to give significant increases in lipids annotated in Red Cross plasma and substantia nigra tissue compared to conventional TopN analysis [123]. Another example obtained 29% additional peptide annotations in HeLa cells with this technique, and a further 10% by only excluding those targets which could be identified between runs [124].

2.3.2 DIA

Fragmentation strategies for DIA are much simpler to describe than DDA. All Ions Fragmented (AIF) mode simply fragments every precursor on every MS2 [125]. Such spectra are

[9]TopN Exclusion is often named using some variation of the term "Iterative Exclusion" e.g. IE-DDA, IE-MS, IE-omics, etc. But iterative exclusion approaches aren't just limited to TopN-like approaches.

of course extremely difficult to interpret, because the parent of a given fragment could be any currently eluting precursor ion. The most notable alternative strategy is SWATH, where smaller (though still one or two orders of magnitude larger than DDA) partially-overlapping isolation windows are run sequentially to cover a larger mass range a part at a time [126].

Obviously, signal deconvolution is quite difficult to perform *de novo*, and gets more challenging as more analyte species are captured in the isolation window. But one of the primary differences between DIA and DDA data is that while DDA samples MS2 data at an isolated, variable timepoint, DIA is continuously and regularly sampling MS2 spectra for a potentially large group of ions. Ions analysed by DIA therefore form chromatographic traces on the MS2 dimension as well as the MS1 dimension. This then allows the MS1 and MS2 peaks to be correlated by RT — if a given fragment appears only for the time a given precursor ion is observed in the MS1 data, then it is very likely that it comes from that precursor ion and forms part of its fragmentation spectrum [10]. Statistical modelling then allows you to associate the most plausible fragments with the MS1 data [127, 128, 129].

Studies into the effectiveness of *de novo* DIA deconvolution have generally found poor performance and lower spectral quality compared to DDA [24, 25]. Poor deconvolution risks both false negatives and false positives. Still, DIA remains popular within proteomics [130]. One of the most commonly-cited reasons is DIA's quantitative performance — regular sampling of MS1 and MS2 data across wide mass ranges theoretically gives more information on peak shape and hence quantification. The regular sampling also means DIA data is more reproducible.

Another advantage of DIA is that not all deconvolution is *de novo*. If the identity of one of the analytes is known prior to attempting to deconvolve the others then it is possible to essentially factor it out and make the remaining problem easier. One way of getting these known factors is to combine the analysis of DDA and DIA methods [131, 132]. Another popular approach is database matching — if a given analyte is known to appear at a particular RT, and it looks like its fragmentation spectrum is plausibly at that RT, then its annotation can be performed before deconvolving the rest of the signals [127].

Furthermore, targeted acquisition can be used to confirm a database match [126] — Hybrid-DIA attempts to insert these targeted scans within the same acquisition [133]. But this has the disadvantage of creating bias towards compounds with known spectral matches, but as we covered in Section 2.2.5, database coverage is very limited. While the confirmation by targeted scan helps, false positives will result in cumbersome wastes of scan time, and may result in erroneous downstream analysis. It is likely the case that peptide space is better characterised than metabolite space (e.g. due to the increased popularity of proteomics as a

[10] Note that some additional data-processing is required to get clean MS2 peaks out of fragments with similar mass e.g. separating overlapping peaks.

field and the fact that peptides are basic building-blocks of proteins) so this may be a reason that DIA is more popular in proteomics.

It has also been argued that because it is very comprehensive, DIA data can be repeatedly re-analysed later with better software tools and chemical knowledge [126, 134, 135]. Items missed in DDA or pre-scheduled data would require the acquisition process to be carried out again. However, this is largely speculative — it is not necessarily clear, for example, whether a given piece of historical data even contains the information necessary to deconvolve the analytes (e.g. too large an isolation window size was selected). Additional targeted experiments can be used, but this negates the advantage of not needing additional acquisition, and potentially risks going in circles looking for patterns in data that aren't there.

But of the motivations given for DIA, perhaps the most interesting for our purposes is that it allows the analysis of low-abundance analytes [130, 124, 131]. This is not an inherent property of DIA — rather, it stems from the fact that *DDA* methods are overly biased towards high-intensity analytes and thus tend to miss low-intensity analytes. The subpar comprehensiveness of alternatives (DDA and pre-scheduling) is a significant motivator for DIA's use. Additionally, if proteomics methods rely on peptide space being better characterised than metabolite space, this implies many of the methods used for DIA in proteomics will not easily translate to the metabolomic space. This directly motivates our approach to attempt to improve DDA and pre-scheduled methods as an alternative to improved DIA analysis.

2.3.3 Pre-Scheduling

Having covered both of the most mainstream approaches, we will now cover pre-scheduled methods. These are often considered to be a subset of DDA, but the fact that their processing is completely offline like DIA (and therefore robustness is more of a challenge) makes them worth delineating as their own category. A recently published pre-scheduled method is **DsDA** (Dataset-Dependent Acquisition) [136]. This method was designed to focus on the commonalities between a set of LC-MS/MS runs, hence being "Dataset-Dependent". It is an iterative scheme where the last n runs are used to plan for the $(n+1)$th run. As the result of a run is collected, it is peak-picked and aligned with all other runs so far — the provided implementation uses a simple protocol in XCMS to do this. Each aligned peak is scored so that increased numbers of runs appeared in and increased MS1 intensity increase the score, and increased MS2 intensity at a previous fragmentation decreases the score. These scores are then used to prioritise filling in targets for MS2 scans in a preset schedule of scans in a TopN-like pattern. To start the sequence of runs, a TopN run is used.

DsDA was shown to increase sample coverage compared to TopN, and to distribute excess MS2 scans to peaks with lower precursor intensity, for which additional fragmentation spectra may aid annotation most. This was demonstrated in long-run experiments of 20, 40 and

50 LC-MS/MS runs. A later work compared DsDA against TopN by adjusting the level of sample variation for each LC-MS/MS run (in simulation). It found that DsDA outperformed TopN at low levels of sample variation, but as samples became more different from one another TopN became more performant relative to DsDA [137].

Another recent pre-scheduled method is MS2Planner [138]. MS2Planner is given some existing representative data which mimics the structure of the acquisition it is to perform (e.g. a prior fullscan) then it extracts relevant targets, and finally converts those targets into a longest-path graph problem. By finding the longest path on a directed acyclic graph where transitions represent moving from one target to another, MS2Planner tries to optimise the total number of targets acquired.

Unlike DsDA and the pre-scheduled methods we will be considering later, MS2Planner tries to optimise scan *lengths*. The user is allowed to set a global minimum value for the total ion current (TIC) of each MS2 scan (i.e. how many precursor ions will be collected, affecting the quality of the scan) and only targets above this TIC will be considered. The TIC is predicted from the MS1 intensities within the scan window on the representative data: a linear regression is used to do this. A minimum time is given to each MS2 scan to ensure it can be above this TIC threshold.

Because of this, the lengths of scans MS2Planner considers are variable: we will generally be assuming that MS1 and MS2 scans each have a single mean length and are distributed closely around that mean. The existing implementation of MS2Planner will also only consider one sample type at a time, and iteratively exhaust targets from that sample type, where it may be possible to combine the analysis of multiple sample types. Because of these differences in our assumptions, we will not be benchmarking our new methods against MS2Planner in later chapters (but we will compare against DsDA in Chapter 6).

There is one significant difference between DsDA and MS2Planner that will be important in Chapter 6, though it will be in the context of comparing DsDA to our new methods. MS2Planner decides its schedule based on fullscan (MS1-only) data collected beforehand, whereas DsDA uses the iterative set of previous runs. This design choice has many consequences. Most obviously, it requires the collection of an extra fullscan file outside of the fragmentation runs, but this data is itself valuable and is often collected as standard protocol as a result. It also results in DsDA being more sensitive to the choice of N in its scan schedule: more MS2 scans means more chances to acquire fragmentation spectra, but it also means the peak-picking of previous runs will be less accurate. On the other hand, peak-picking is sensitive to the number of run outputs given to it, and the number of picked peaks will nearly always increase with the number of those inputs, even if they are of the same sample type. By having access to more runs in large experiments, DsDA may eventually form a clearer view of the peaks. Finally, note that as a consequence of this choice DsDA

may only greedily operate on the next upcoming run whereas MS2Planner is able to plan for several at once (although in actuality it also uses a greedy algorithm, where it simply removes anything targeted in the previous run).

Finally, the Thermo vendor method AcquireX also appears to contain a pre-scheduled component (which is likely used to augment TopN via inclusion windows). Vendor methods are bundled with the machine and consequently are often what is used in practice (which is one of the reasons vendor-provided implementations of TopN continue to endure). But unfortunately vendor methods (including AcquireX) are generally proprietary, with limited detail released to the public (and are therefore not cross-platform either). Thus, like MS2Planner, we will not use AcquireX in our benchmarking in later chapters.

2.4 ViMMS

One final technology we must address is **ViMMS** (the Virtual Metabolomics Mass Spectrometer) [137, 26]. ViMMS is a Python-based simulation software for LC-MS/MS metabolomics designed to allow the testing of LC-MS/MS fragmentation strategies. Specifically, it can produce realistic MS1-level information from either a list of chemicals provided by a user, or by mimicking the structure of existing MS1 data. MS2 data can be provided by a variety of means, including by being manually specified for each chemical, or by sampling from some distribution. The primary purpose of ViMMS is to provide a realistic mock-up of LC-MS/MS data for the purpose of testing fragmentation strategies, thus ViMMS does not directly simulate any biology (nor any of the underlying physics of a mass spectrometry instrument). To produce biologically plausible data the input to ViMMS must itself be biologically plausible. ViMMS allows three primary kinds of experiment to be run:

- Simulated experiments can be run using an input list of chemical objects. ViMMS provides some facilities for sampling purely synthetic datasets of chemical objects from statistical distributions or databases. Alternatively, if the user has a list of chemical objects or has a program that can produce such a list (including m/z, RT and max intensity, and so on) then this can be given as input to ViMMS. ViMMS may also generate plausible values for any piece of information missing from the list of chemicals in this case.

- Simulated experiments can also be run using **re-simulated** datasets created from existing data. This procedure mimics the structure of LC-MS/MS data directly without reference to underlying chemicals. For this, the input is an .mzML file containing data which we wish to replay. If scan times need to be adjusted then new scans are created by linearly interpolating values. This approach can be used to repeat data exactly, or a

larger collection of data can be sampled to produce a synthetic dataset with the same characteristics as the dataset sampled.

- Lab experiments can be run through ViMMS controlling a real LC-MS/MS instrument through its API, given some bridging code between the two.

Synthetic datasets can be used for precise control and knowledge of the parameters of the experiment in ways that would not be straightforward in a more realistic scenario. For example, the experiment mentioned in Section 2.3.3 used synthetic datasets to compare DsDA against TopN at varying levels of sample similarity. Re-simulations, on the other hand, replay existing data. This can be used to test experimental hypotheses in a realistic scenario or to tune parameters in preparation for lab experiments. Instrument control allows for final validation of and actual deployment of novel fragmentation strategies for future experiments.

Importantly, the same Python controller code implementing a fragmentation strategy can be used for all of these methods. Only datasets are swapped when changing from one kind of simulated dataset to the other, and a simulated MS object is replaced with an instrument connection when connecting to a real machine. This approach is also theoretically cross-platform, requiring only some bridging code between the Python-based high-level code in the ViMMS codebase and an instrument control API. Practically speaking, this is limited by the completeness of ViMMS' features relative to the target instrument model[11] and the fact that at the time of writing only one set of bridging code has been written, for the Thermo Fisher "IAPI" (Instrument Application Programming Interface) and thus any instruments compatible with it.

Nonetheless, the ability to write code which can run both in a simulated context and on the connection to a real instrument is a powerful approach. In later sections of this thesis we will extensively use re-simulation features and supplement them with lab experiments using dynamic instrument control. The specifics of our use will be later explained in Section 3.3.

2.5 Conclusion

In this chapter we have introduced the current status of fragmentation strategies in LC-MS/MS and much of the background context required to understand them. Here are some of the most salient points to remember going forward:

- Mass spectrometry (MS) produces "spectra" which are composed of $(m/z, intensity)$ pairs. m/z is the ratio of a molecule's mass to its number of charges, and intensity is

[11] For example, you cannot model ion mobility mass spectrometry using ViMMS at the current time.

how much signal they produced on a detector (which is related to relative abundance of ions).

- Liquid Chromatography (LC) separates ions in a mass-orthogonal way, which is used to generate them into the MS over time. The time a molecule is retained by the LC column is known as the retention time (RT). This causes LC-MS data to be triples of $(rt, m/z, intensity)$. We call a single injection of sample into LC-MS a "run".

- When LC-MS scans are ordered by RT, analytes (and contaminants etc) of sufficient abundance will produce traces along RT that have a Gaussian-like peak shape. We typically start from these "chromatographic peaks" when analysing the data.

- In tandem mass spectrometry (MS/MS), we isolate ions of a particular m/z range, fragment them, and perform an MS scan of the fragments, giving us $(m/z, intensity)$ pairs. The distribution of the fragment masses is characteristic of certain molecules or molecular structures, so we can use this as a "molecular fingerprint" in, say, a database lookup.

- In LC-MS/MS we have a choice of MS1 scan (measure unfragmented precursor ions currently eluting from the LC) or MS2 scan (isolate precursor ions from the previous MS1 scan in a given m/z range, fragment them, measure the fragments). The range of m/z used by an MS2 scan to collect precursor ions for fragmentation is the "isolation window". How we assign MS1 and MS2 scans in this procedure is the work of a "fragmentation strategy".

- Each MS1 scan is a list of $(rt, m/z, intensity)$ triples at a given RT. An MS2 scan is a list of $(m/z, intensity)$ pairs (a "fragmentation spectrum") for a given isolation window. Chromatographic peaks will again appear on the MS1 data, but will be broken up by MS2 scans, reducing the number of data observations. A "fullscan" LC-MS/MS run contains only MS1 scans.

- After data acquisition an early processing step is "peak-picking", where chromatographic peaks are identified in the data. These are often represented by a bounding rectangle drawn in $(rt, m/z)$ space around a given peak. In order to pick peaks software tools like XCMS [74] and MZMine [75] typically begin by roughly tracing bounding boxes as RoIs (Regions of Interest) which are later filtered down into more precise peak bounds.

- Assigning putative identities to compounds analysed by LC-MS/MS is very difficult due to the sparsity of database entries, the risk of false positives and the complexities of *de novo* structural elucidation. Many methods have been proposed to improve spectral comparisons, transform from spectra to structures and vice versa, and to compare to

structural databases, but compound annotation remains an active area of research and a challenging research problem.

- A fragmentation strategy can be either "targeted" or "untargeted". Targeted experiments restrict themselves to a subset of interesting analytes, whereas untargeted experiments attempt to collect data for all analytes. Truly global untargeted analysis is both more enticing and more difficult.

- DDA (Data-Dependent Acquisition) and pre-scheduled strategies attempt to isolate single ion species for fragmentation, to produce unambiguous spectra. Consequently, their annotation performance is limited by their ability to MS2-target as many unique chromatography peaks as possible near the apex of the peak.

- DIA (Data-Independent Acquisition) is an alternative which isolates many ions at once, but this creates a complex statistical deconvolution problem which is fundamentally orthogonal to the challenge faced by DDA and pre-scheduled methods.

- DDA and pre-scheduled methods are often considered one category, but DDA responds to data in real-time and pre-scheduled methods compute a plan offline. The latter approach allows more sophisticated processing but is less robust.

- Inclusion and exclusion windows are commonly used to augment DDA strategies: these take the form of $(rt, m/z)$ rectangles, the same format used by both RoIs and picked chromatographic peaks. As the names suggest, an inclusion window tells the DDA strategy to prioritise an area, and an exclusion window tells it to avoid that area.

- Pre-scheduled methods can either iteratively process data as it comes in to plan the next run, or they can pre-process some representative data which mimics the data they actually want to acquire.

- The most common DDA method is TopN, which follows each MS1 scan with up to N MS2 scans that target the most abundant precursors from the MS1 scan.

- Standard implementations of TopN include "Dynamic Exclusion Windows", which forbid targeting an m/z value again for a limited time.

- TopN Exclusion [122] is a method which extends this to multiple LC-MS/MS runs by remembering the DEWs between runs.

- SmartRoI and WeightedDEW [23] are recently published DDA methods which focus on extending DEWs with advanced rules allowing more complex decision-making.

- DsDA (Dataset-Dependent Acquisition) [136] is a recently published pre-scheduled method which iteratively processes runs. Potential targets are scored highly if they appear in many of the processed runs or if they have high MS1 intensity, and lowly if they have previously been fragmented at high MS2 precursor intensity.
- ViMMS (Virtual Metabolomics Mass Spectrometer) [137, 26] is a simulator for LC-MS/MS data, designed to allow the rapid prototyping of fragmentation strategies. It can replay existing data for a simulated experiment, or control a real instrument through its API.

Having covered these fundamentals, we are now able to move on to developing new untargeted fragmentation strategies for LC-MS/MS, which will increase the number of fragmentation spectra collected and their quality. First, however, now we have covered the terminology we will describe the remainder of the thesis in greater specificity compared to Section 1.2.

Chapter 3 will explain the broad methodology used for evaluation — this will use a strategy based directly on peak-picking rather than compound annotation given the difficulties with the latter. We will also explain the specifics of how we use ViMMS to run experiments, and the datasets we will use.

In Chapter 4, we will investigate the use of DDA strategies. Specifically, we will extend TopN-like strategies by comparing current intensity values to fragmentations from previous runs. We will also compare real-time RoI traces in the current run to exclusion windows from previous runs, which are generated by remembering previously fragmented RoIs. This will use comparisons of the geometric area of the RoIs and exclusion windows. These two concepts will also be combined to allow a method that uses both intensity and area-based information. The primary experiments demonstrating that this allows obtaining more fragmentation spectra at higher intensity values will be presented in Chapter 4, then Chapter 5 contains a number of other experiments testing different method parameter settings, peak-picking settings, sample orders and sample types to show the generalisability of the methods.

Chapter 6 takes a complementary approach by instead investigating a new pre-scheduled method for data acquisition. It extends a previous maximum bipartite matching technique [23] by allowing it to plan for multiple sample types, optimising acquisition time by targeting the maximum intensities and by reassigning leftover scans to improve robustness. Our experiments will show that in principle this allows completely comprehensive data acquisition within only a few runs thanks to the global planning of this method. However, because pre-scheduled methods are offline in contrast to DDA, we will also focus on the trade-off in robustness between DDA and pre-scheduled methods. This will demonstrate that the improvements made to the previous technique are necessary for practical use, but also that further progress may require hybridisation with DDA.

Chapter 3

Methodology

The overarching goal of this thesis is to present improved fragmentation strategies for untargeted LC-MS/MS data acquisition, but to do so, we must quantify "improvement". The most direct way to evaluate a fragmentation strategy would be to count how many metabolites can be annotated by using it in a given experiment. This experiment would collect fragmentation spectra from a set of LC-MS/MS runs, and then it would be possible to compare them against a known database of ground truth metabolites. Every match above a certain accuracy threshold would then be counted as a point in the fragmentation strategy's favour. This is a standard approach, and useful because it is a direct representation of "real-world" performance — an untargeted metabolomics experiment is aiming to annotate as many metabolites as possible and comparison to databases is a typical initial step in doing so.

However, as we explained in Section 2.2.5 metabolite annotation is difficult, and an active research problem. One of the reasons for this that we discussed is that databases are often incomplete or prone to false positives. As a consequence, evaluation by metabolite annotations comes with several difficulties. Firstly, it is reliant on the accuracy of the ground-truth database. While it is true that if an improved fragmentation strategy were to be deployed today, analysis would be largely reliant on the completeness of existing databases, new data continues to be published, and so how a fragmentation strategy will be evaluated tomorrow may change. Furthermore, since we are trying to design new untargeted fragmentation strategies that may let us detect the presence of unknown metabolites, an evaluation strategy that only considers known metabolites has a certain bias. For relatively well-understood samples this may be fine but there may be low-intensity "borderline" metabolites which only an improved fragmentation strategy can annotate. We also have to be cautious of false positives when database matching [92]. It is therefore useful to have a different method of evaluation to complement metabolite annotation.

A second motivation for a different evaluation protocol is that as computing scientists we also care about *algorithmic* behaviour (in order to obtain insight that will let us develop

better algorithms). Decomposing the evaluation to a more "intrinsic" measure detached from biochemistry and the current status of the field provides a complementary perspective based more on the structure of the LC-MS/MS data itself. Thirdly, simulated experiments are much less expensive to run compared to lab experiments, and we will be using them extensively in this thesis. However, to use metabolite annotations with *simulated* data it is necessary for the simulator to give a realistic representation of a biological mixture. This is a difficult research problem in its own right, and requires building in assumptions to the simulator which may lead to erroneous conclusions.

To address these difficulties we use an intermediate evaluation step where the strategy is evaluated on *peak coverage* and *intensity coverage*. Peak coverage (which we will usually refer to simply as coverage) is a measure of how many chromatographic peaks we have collected a fragmentation spectrum for. Intensity coverage measures the average maximum precursor intensity each peak was fragmented at. A higher precursor intensity for a given MS2 scan means more precursor ions to fragment, and thus typically a higher-quality fragmentation spectrum. Full definitions are given in Section 3.1. These metrics are calculated from an output set of .mzML files [139, 140, 141] produced by a fragmentation strategy and a set of target peaks. Computing these metrics requires linking various pieces of data between the output .mzMLs and the target peaks (an example of what we need to link is the intensity each peak was fragmented at) and doing this efficiently can be done with some simple computational geometry. To produce the target peaks for the evaluation we used existing peak-picking tools on fullscan .mzMLs of the same sample types given to the fragmentation strategy — details of the setup are in Section 3.2. Section 3.3 describes our use of ViMMS to run experiments and Section 3.4 describes the datasets we will use in later chapters (we cover them together as they use the same underlying data).

The evaluation techniques described in this section supported all the primary work described in Chapters 4 [27] and 6 but also some secondary publications [26, 25]. Their implementations remain a core part of the infrastructure of ViMMS.

3.1 Evaluation Metric Definitions

The two primary evaluation measures we will be using in this thesis are peak coverage [1] and intensity coverage. As mentioned in the introduction to this chapter, these metrics can be computed given an ordered list of .mzML fragmentation runs to evaluate, and a list of target chromatographic peaks to evaluate them against. A minimum intensity threshold parameter should also typically be provided.

[1] There are other ways to measure sample coverage (like metabolite annotations) but since we will only be using peak coverage, we will often refer to it just as "coverage" in non-background material.

Peak coverage and intensity coverage are defined using some simple geometry. For the purposes of determining coverage, each MS1 scan in the input .mzMLs is treated as a collection of $(rt, m/z, intensity)$ points with identical RT values but differing m/z and intensity values. Similarly, an MS2 scan is treated as either a single point in $(rt, m/z)$ space or a 1D interval with finite width in the m/z dimension. When treated as a point we use the midpoint of the isolation window, and when treated as an interval the entire isolation window is used as the interval. A precursor intensity from the MS1 points is also associated with each MS2 scan depending on which one of these two representations is being used. Finally, target peaks are specified as 2D $(rt, m/z)$ intervals i.e. rectangles.

We say a peak is *targeted* [2] for fragmentation if there is an MS2 scan associated with it and that MS2 scan has a valid precursor above the intensity threshold parameter (as there is a minimum intensity at which a fragmentation spectrum is useful). **Peak coverage** gives the number of targets which we have successfully targeted for fragmentation across some set of input .mzMLs, so is simply the number of peaks for which these criteria are met.

Let T be the set of target peaks, τ be a member of T and $|T|$ be the cardinality of T. Additionally, let λ_{min} be the minimum intensity threshold and let λ_f be a given target intensity for a peak τ where f is used to denote an instance of targeting for MS2 (the "f" is for "fragmentation"). $\underset{f}{max}(\tau, \lambda_f)$ denotes taking the maximum targeting intensity of a given target peak τ over all fragmentation events f. Inequalities should be interpreted as outputting 1 if true and 0 if false. Now Equation 3.1 gives the definition of coverage.

$$coverage = \frac{\sum_{\tau \in T} \left(\underset{f}{max}(\tau, \lambda_f) \geq \lambda_{min} \right)}{|T|} \tag{3.1}$$

But how do we decide when a given MS2 successfully targets a given peak? If the evaluation is defined to be in "point mode" (i.e. MS2s are treated as points rather than intervals) then an MS2 is associated with a peak if its point falls within the peak's bounds. A precursor of that peak is any point on the previous MS1 scan which falls both within the peak bounding box and within a small m/z tolerance of the MS2 point.

Conversely, if operating in "interval mode" then a peak is considered to be associated with an MS2 if its interval completely envelops the peak on the m/z dimension at any point within the peak's RT bounds [3]. To be a successful targeting, that MS2 must then also have a precursor in the previous MS1 scan which falls within the m/z bounds of the MS2 interval and the bounds of the peak box on both dimensions. We will generally use interval mode in

[2] Note that the term "targeted" is overloaded. "Targeting" in the sense of scheduling an MS2 scan is separate from whether an experiment is targeted or untargeted, which relates to the goals of the experiment.

[3] Isolation windows on our setup are generally two or three orders of magnitude larger than the m/z width of peaks we produce, so anything directly targeted by a DDA or pre-scheduled strategy should meet this criterion.

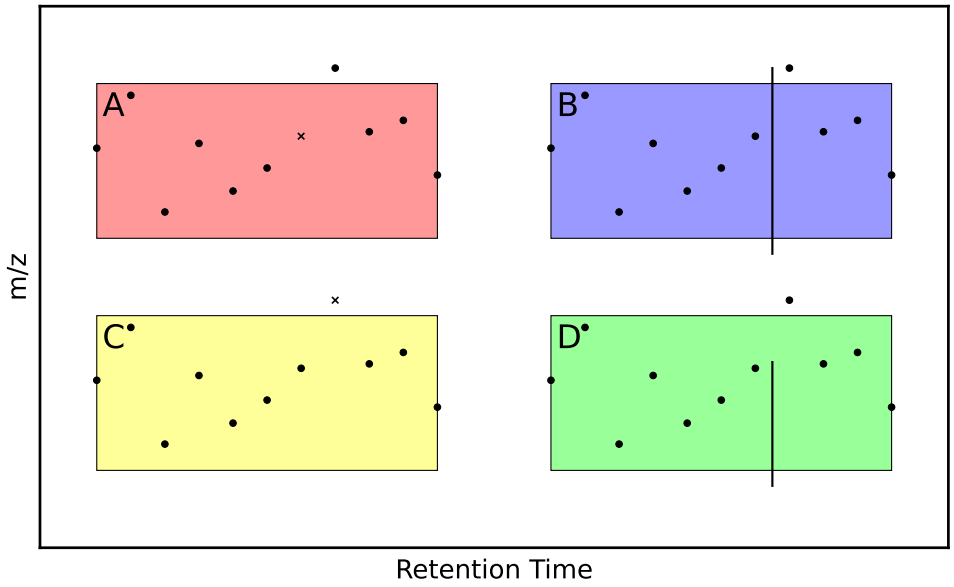

Figure 3.1: A toy example of how the two different modes of the evaluation work. Points represent individual points from MS1 scans. A and C show evaluations in point mode: a cross represents a targeted precursor. B and D show interval mode, where the vertical line shows the interval (not to scale: isolation windows are several orders of magnitude larger). A and B are successful targetings: C fails because the targeted precursor (the cross) is outside of the box, and D fails because the interval does not fully envelop the peak box.

this thesis (it is slightly more realistic as in reality all ions in the window will be collected) but previous work [23] used point mode. An example can be seen in Figure 3.1.

Intensity coverage is a generalisation of peak coverage. Each peak successfully targeted is given a score in the range $[0, 1]$ equal to the proportion of the precursor intensity of the fragmentation against the highest MS1 intensity observed in any MS1 point contained inside the peak box. Given multiple instances of successful targeting, the one with the highest precursor intensity is used, meaning that the best attempt is counted. Also, any peak which has not been targeted above the minimum intensity threshold is treated as having a score of 0, to only include those peaks which would count for peak coverage. The intensity coverage is the sum of these $[0, 1]$-bounded scores, and so is an aggregate measure of actual maximum acquisition intensity compared to a theoretically possible maximum. Intensity coverage is a generalisation of coverage, so to obtain coverage from intensity coverage, one simply thresholds each peak's individual score to be 1 if it is non-zero and 0 otherwise.

To describe the equation for intensity coverage, let all symbols be as they were in Equation 3.1, but additionally define λ_p to be MS1 intensity at a given point (or precursor) p, such that $\underset{p}{max}(\tau, \lambda_p)$ gives the maximum observed MS1 intensity for any target peak τ. Then intensity coverage is defined as in Equation 3.2.

$$intensity\_coverage = \frac{1}{|T|} \cdot \sum_{\tau \in T} \left(\frac{\underset{f}{max}(\tau, \lambda_f)}{\underset{p}{max}(\tau, \lambda_p)} \right) \tag{3.2}$$

For example, if this equation were to be applied to Figure 3.1A, then $\underset{f}{max}(\tau, \lambda_f)$ would be equal to the intensity of the cross representing the precursor of the MS2 scan. If applied to Figure 3.1B then $\underset{f}{max}(\tau, \lambda_f)$ would be equal to the intensity of the MS1 point from the prior scan overlapped by the interval. In both cases $\underset{p}{max}(\tau, \lambda_p)$ would be equal to the highest intensity of all the points contained within the box being considered.

One aspect to note here is that the target intensity is converted into a proportion of the precursor intensity per each individual spectra, prior to any aggregation. In essence, this flattens the scores each target spectrum receives onto the same $[0, 1]$ scale. This is important because if the raw intensities were summed prior to division, this would cause the calculation to be dominated by high-intensity peaks. Given the orders of magnitude in difference between peak intensity, this could cause a severe bias — and since these peaks would likely be acquired by a simple heuristic and would be easier to identify regardless of quality, we are typically more interested in the low-intensity peaks that would be neglected in this case. Performing the sum first would potentially also be prone to numerical error from floating point representations.

We also have not defined any specific procedure for handling adducts and isotopes within the scores. If the user is interested only in hitting one specific isotope then it suffices to de-isotope the data to leave only one peak (our processing with MZMine in Section 3.2 does this). Alternatively, if the user were happy to fragment any isotope of a given peak then it would suffice to take the maximum intensity targeting among any of those adducts/isotopes and carry that information forward into the target set prior to computing (intensity) coverage. We have not implemented any such behaviour, however.

3.1.1 Should Simulation Include Fragmentation Spectra?

An unusual property of these metrics is that they do not actually reference any fragmentation spectra produced: they only consider the geometric positions of MS1 points, MS2 points/intervals and peak bounding boxes. This provides a level of abstraction from the underlying mass spectrometry, which is both advantageous and disadvantageous. With these metrics it is more obvious what an algorithm is *attempting* to do, what it *intends* to target and therefore what obvious inefficiencies and redundancies it is committing to. If using a peak-picker, the peaks in the target list are computed based on intrinsic properties of the .mzML data, and do not rely on (potentially incomplete) external chemical knowledge, so answers to why a fragmentation strategy is successful on these metrics can largely be found

in the .mzML data itself. As a result of that we can also evaluate a method on whether it successfully targeted parts of the data that had a reasonable possibility of corresponding to a metabolite, but cannot actually be matched to any reference metabolites. A fragmentation strategy which uses otherwise wasted scans on exploring these parts of the data may prove more valuable in the long-term than one which does not, because our current knowledge is incomplete. This method of evaluation does, however, obscure how many annotations that could be revealed were someone to use it right now, so ideally this metric should be used to complement successful metabolite annotations.

Additionally, not requiring the use of fragmentation spectra has benefits for simulation studies. Firstly, compute efficiency: simulated experiments can use less computational resources because they do not have to generate realistic spectra. This is a realistic concern on ordinary hardware — when running a large experiment, a collection of tens of thousands of realistic spectra can have space requirements measured in Gigabytes. Running several of these experiments in parallel may therefore exceed system memory capacity (which will then incur time costs because of virtualisation etc). Therefore this can be useful when prototyping fragmentation strategies at high throughput.

More importantly, this allows simulated data to be generated using much weaker assumptions. In order to produce completely realistic fragmentation spectra for a whole dataset, it would be necessary to exhaustively collect fragmentation spectra for that dataset. Not only would this add an extra phase of acquisition to the experimental workflow, but it is particularly problematic when the issue to be resolved is the poor coverage of existing methods. The alternative would be to use synthetic datasets, and realism in this regard would require intensive (possibly flawed) modelling. Coverage and intensity coverage completely bypass this concern and allow high-throughput simulation studies using ViMMS re-simulation and peak-picking on fullscan data. For all of these reasons, we will only engage in realistic simulation of MS1-level information.

3.1.2 Interpreting Proportional Scores

The definitions of coverage and intensity coverage given in Equations 3.1 and 3.2 are proportional, that is, they use the percentage of the actual obtained coverage/intensity coverage against the theoretically obtainable coverage/intensity coverage. This is easier to interpret than the absolute numbers. But how do we know what would be theoretically obtainable, to use as the denominator in a proportional score? Each set of LC-MS/MS runs carries natural variation in which peaks appear, when, and at what intensity. Even if we created some form of shared "ground truth" list, it would be unfair to penalise a given strategy for not obtaining a peak that simply did not appear or was of unusually low intensity. Thus, the "theoretically obtainable" intensity coverage for a given target under a particular strategy is defined with

reference to the values which actually appeared in the strategy's outputs that we are trying to evaluate. In other words, the intensity coverage denominator for a given peak will be the maximum MS1 intensity that appeared within that peak in any of a set of runs being considered for evaluation. This has some unusual consequences that are worth highlighting.

Using the MS1 values that appeared within the runs themselves means that the results depend on the order in which the MS1 and MS2 scans are run. In some cases this may create a perverse incentive to avoid performing MS1 scans, to avoid observing new peaks which could potentially lower the overall score if they were not targeted. We will avoid this issue with the new strategies we will introduce in Chapters 4 6 by pre-committing to TopN-like duty cycles for a fixed N. But it is often still important to check that conclusions about coverage are consistent across both absolute and proportional values, especially in situations where the strategy has more control over the order of MS1 and MS2 scans.

Secondly, the use of maximum MS1 intensities that appeared in the set of runs in question means that later runs can affect the proportional (not absolute) score in previous runs. For example, suppose we do an experiment of one run and obtain a peak at the maximum intensity, meaning it gets an intensity coverage score of 1.0. If we then perform a second run, observe the same peak at a higher intensity without fragmenting it, and correctly align this peak to the first run, then the intensity coverage score for that peak will drop below 1.0. This mostly manifests in the form of small reporting artifacts where several methods may perform identical TopN runs in an identical simulation yet receive different proportional scores because they took different actions later [4].

3.1.3 Algorithms

Now we will consider the algorithms we use to compute coverage and intensity coverage. Computing evaluation information for a set of .mzMLs (where each .mzML is the output file for a single LC-MS/MS run) happens in two phases. First, some basic information from the .mzMLs must be associated with each peak in the target list. For each peak its maximum MS1 and MS2 precursor intensity in the given set of .mzMLs must be collected [5]. Each piece of information is listed in a 2D matrix for the aligned peaks in the target list — rows give a given aligned peak and columns give the parent .mzML. The second step is to compute metrics from this matrix.

Filling out the values in the matrix using the information in the .mzMLs can be done by iterating through the data scan-by-scan. Whenever we pass the start RT of a target box, we add it to a list of active boxes and remove from the list of active boxes any box whose end

[4] In this case the absolute number of peaks covered will be identical, even if the proportional value is not.

[5] Also, although not used for this calculation, it is useful to collect the number of times the peak was targeted.

RT we have passed. Next we iterate through each m/z in the scan. If it is an MS1 scan, then each active box with a point lying within its m/z bounds has its maximum MS1 intensity updated — we already know it must lie within the RT bounds of any active box. For an MS2 scan, the behaviour depends on whether we are evaluating in point or interval mode. If in point mode then the targeting attempt is assigned to a peak if the isolation window centre (the MS2 point) lies within the peak box, and the precursor is the highest intensity point from the last MS1 scan within the box and within a small (e.g. 10ppm) error tolerance. For interval mode, the isolation window must envelop the box on the m/z axis, and the precursor is the highest intensity point falling within the box in the last MS1 scan. In either case, the box's maximum MS2 precursor intensity is updated.

Since we are working with intervals and points on the m/z dimension, we use an interval tree for querying scans against the target set. An interval tree is a standard data structure for performing interval overlap queries which allows us to search in log(n) time. The interval tree is populated with the active boxes and contains their m/z bounds as intervals: it can then be easily queried for which of the active boxes are relevant to a given MS1 point, MS2 point or isolation window. Strictly speaking, since we know we are going to do a large batch of queries at once, it may be faster to sort the queries and target m/z intervals and iterate through them in order, remembering which are active. However, since this operation is not performance-critical this was not tested (as the implementation is slightly more involved).

Once the target n run intensity matrix is populated by this first step, computing relevant metrics can be done simply by operations on the rows and columns. For example, we can first compute the maximum observed MS1 intensity for each peak across all runs by taking the maximum across all columns. The cumulative intensity coverage can then be computed by dividing the original columns of the max MS2 precursor matrix by this 1D MS1 intensity vector, and then applying a cumulative maximum. In our implementation we use NumPy to efficiently vectorise these operations. As an aside, separating these two steps also has the convenient benefit of making the implementation more transparent (i.e. it is possible to examine the individual values in the matrix prior to aggregating them).

3.2 Peak-Picking

Computing coverage and intensity coverage is parameterised by a "target list" of interesting peaks we wish to fragment. Where we obtain the target list from is a highly consequential choice: how we evaluate a fragmentation strategy's performance may entirely change with the target list. As already mentioned, one of the advantages of coverage and intensity coverage is being able to estimate performance on samples that may contain unknown compounds, but this also requires the creation of a target list suitable for this evaluation. Creating a tar-

get list using only known metabolite identities would likely force careful choices in sample selection (so that the sample would simultaneously be well-characterised and representative in terms of difficulty) and a burden in terms of manual effort.

An alternative to this means of target list creation is to select the target list with an automated peak-picking tool. This allows evaluation to be performed at very high throughput on any sample of one's choice. As we explained in Section 2.2.4, peak-picking is a standard part of post-acquisition data processing. Although useful, peak-picking is complex and prone to producing messy results, so it is often only used as a preliminary stage to filter the vast quantity of data that has to go through expert evaluation. However, so long as the tool is "good enough" at capturing some important information in the intrinsic structure of the data then any difference in fragmentation strategy effectiveness should also be borne out in evaluation results.

The current gold standards of peak-picking are the software suites XCMS [74] and MZMine [75], and their $centwave$ [77] and ADAP algorithms [78], which rely on continuous wavelet transforms (see Section 2.2.4 for a review of these tools). These algorithms, however, have an extremely large list of parameters and are very sensitive to the choices made (and good choices vary between setups). Given the impact a single choice of peak-picking setup may have, we use several different setups for evaluating our experiments in the following chapters.

With MZMine, we use two different "restrictive" and "permissive" parameter sets processed with MZMine 2.53, which will appear throughout Chapters 4 to 6. These two parameter-sets have different quality thresholds, influencing the number of output peaks and hence which fragmentation strategy is most effective. MZMine 2 can run a sequence of processing steps via command-line invocation of a "batch mode" file, which is formatted as an .xml file (and is normally created via the MZMine GUI). To invoke MZMine via ViMMS, a "template" file is provided. This template file specifies all the processing steps and their parameters but uses placeholders for the names of input and output files. When batch mode is invoked by Python's subprocess module at runtime, a modified version of the template specifying input and output file names is supplied. The output is a .csv file containing a list of peaks, and this is what we use as the target list.

The "restrictive" and "permissive" parameter sets are each defined as "batch mode" templates. The "permissive" parameter set is an existing parameter set used in [23]. The "restrictive" parameter set is one with much higher quality thresholds defined in consultation with a mass spectrometry expert. Generally, peak-picking is configured to minimise the chances of any false negatives (i.e. real metabolites for which the peak is not picked up). Even the restrictive parameter set will typically produce many more peaks than there are known metabolites. However, a more permissive parameter set also provides a more difficult algorithmic challenge. If we can improve fragmentation strategies to acquire data for a

wider set of "interesting-looking" peaks then it may follow that the typical way to set quality thresholds will become less restrictive too.

A full description of parameter values can be found in Tables 3.1 and 3.2. Italicised value names indicate a change in parameter setting from the permissive parameter set to the restrictive. Most of the names of the steps are self-explanatory. The crop filter removes some data at the end of the RT and m/z ranges as there is a limited range for peaks associated with the actual analytes. The mass detector steps filter out some points below a certain intensity threshold, and centroid it (essentially averaging some points across the m/z dimension because signals also appear across a peak-like distribution on this axis). These two steps are primarily to reduce the processing necessary for the peak-picker [6].

After the crop filter and mass detector stages, the ADAP Chromatogram Builder builds Regions of Interest, and the Chromatogram Deconvolution filters these into final peaks. The Isotopic Peak Grouper de-isotopes the data, reporting only the monoisotopic peak for a given isotopic distribution across the m/z axis. The join aligner aligns different LC-MS/MS runs so that "the same" peak observed across multiple runs can be recognised as a single object in the underlying process. MZMine's join aligner algorithm simply overlays different runs one-at-a-time, and assigns peaks from the new run to whichever peak in the running list is closest. To define distance a parameter is given for how to weight m/z distance vs RT and there is a maximum tolerance for both RT and m/z distance.

The final output after all steps is a matrix where a row gives all attributes for a single aligned peak and a column lists a specific attribute of all the peaks appearing in a given input .mzML. Attributes include RT and m/z aggregated for the whole set of .mzMLs, but also the precise (rectangular) RT and m/z bounds of the peak appearing in the .mzML and whether the peak was detected in that .mzML, estimated to be there or of unknown status (i.e. not detected).

Note that there is one change between our permissive parameter set and the set used in previous work [23]. MZMine requires both a Dalton and ppm value to be specified for m/z tolerances, and will use the maximum of these two values. In order to ensure the ppm value will always be used, we set a dummy value of $10^{(-8)}$ for the absolute tolerance. Also, initially m/z tolerance values were set to 5 ppm for the restricted parameter set (and the experiments in Chapters 4 and 5) this change was later permanently reverted to a value of 10 ppm (used for the experiments in Chapter 6).

Now we have defined the parameters, to construct target lists from this setup, we must give the peak-picker fullscans that are representative of the experimental data. In Chapters 4 and 5 this is one fullscan of each unique sample type in the experiment. In Chapter 6, we construct target lists in this way, but we also construct target lists by peak-picking and aligning a

[6]For example, XCMS took a few minutes to process for the experiments in Chapter 4 whereas MZMine required several hours of processing. The data in even a single .mzML is large and these algorithms are quite intensive.

MZMine Parameters (Permissive)	
Raw Data Import	
Crop Filter	
Retention time	0.5 to 30 minutes
m/z	50 to 1060 Da
Mass Detection (MS1)	
Mass detector	Centroid
Noise level	1000
Mass Detection (MS2)	
Mass detector	Centroid
Noise level	0
ADAP Chromatogram Builder	
MS level	1
Min group size in # of scans	3
Group intensity threshold	500
Min highest intensity	5000
m/z tolerance	$10^{(-8)}$ Da or 10.0 ppm
Chromatogram Deconvolution	
Algorithm	Wavelets (ADAP)
S/N threshold	10
S/N estimator	Intensity window SN
Min feature height	5000
Coefficient/area threshold	1
Peak duration range	0.10 to 5.00
RT wavelet range	0.10 to 1.00
m/z centre calculation	Median
Range for MS2 scan pairing (m/z, RT)	0.01 Da, 0.5 minutes
Isotopic Peaks Grouper	
m/z tolerance	$10^{(-8)}$ Da or 10.0 ppm
Retention time tolerance	0.1, absolute (minutes)
Monotonic shape	False
Maximum charge	2
Representative isotope	Most intense
Join Aligner	
m/z tolerance	$10^{(-8)}$ Da or 10.0 ppm
Weight for m/z	80
Retention time tolerance	0.1, absolute (minutes)
Weight for RT	30
Require same charge state	False
Compare isotope pattern	False
Compare spectra similarity	False
Export to CSV	

Table 3.1: Our set of "permissive" MZMine 2 batch mode parameters.

MZMine Parameters (Restrictive)	
Raw Data Import	
Crop Filter	
Retention time	0.5 to 30 minutes
m/z	50 to 1060 Da
Mass Detection (MS1)	
Mass detector	Centroid
Noise level	1000
Mass Detection (MS2)	
Mass detector	Centroid
Noise level	0
ADAP Chromatogram Builder	
MS level	1
Min group size in # of scans	3
Group intensity threshold	*2000*
Min highest intensity	5000
m/z tolerance†	$10^{(-8)}$ Da or 10.0 ppm
Chromatogram Deconvolution	
Algorithm	Wavelets (ADAP)
S/N threshold	*3*
S/N estimator	Intensity window SN
Min feature height	5000
Coefficient/area threshold	1
Peak duration range	*1.0 to 7.00*
rt wavelet range	*1.00 to 5.00*
m/z centre calculation	Median
Range for MS2 scan pairing (m/z, RT)	0.01 Da, 0.5 minutes
Isotopic Peaks Grouper	
m/z tolerance†	$10^{(-8)}$ Da or 5.0 ppm
Retention time tolerance	0.1, absolute (minutes)
Monotonic shape	False
Maximum charge	2
Representative isotope	*Lowest m/z*
Join Aligner	
m/z tolerance†	$10^{(-8)}$ Da or 10.0 ppm
Weight for m/z	80
Retention time tolerance	*0.5, absolute (minutes)*
Weight for RT	20
Require same charge state	False
Compare isotope pattern	False
Compare spectra similarity	False
Export to CSV	

Table 3.2: Our set of "restrictive" MZMine 2 batch mode parameters. Italics indicate a change from Table 3.1. † indicates the value is 5 ppm for Chapters 4 and 5 but was (permanently) reverted to 10 ppm for Chapter 6.

Centwave Parameters		Join Aligner Parameters	
ppm	15	mzVsRtBalance	10
peakwidth	(15, 80)	absMz	0.2
snthresh	5	absRt	15
noise	1000	kNN	10
prefilter	(3, 500)		
mzdiff	0.001		

Table 3.3: Parameters used for XCMS peak-picking and alignment.

different fullscan per run, including duplicates of the same underlying sample type. So if an experiment runs four sample types in some arbitrary order, e.g. 1-2-3-4-1-2-3-4..., we give the peak-picker one instance of a fullscan for each of sample types 1, 2, 3 and 4. It is important to use a fullscan because data with fragmentation information level scans will have fewer MS1 scans and thus less rich MS1-level data, worsening peak-picking results. A fullscan can also be used to seed a re-simulation, so for our re-simulations the peak-picking and re-simulation share the fullscan .mzML used.

In addition to the two MZMine parameter sets, in Chapter 6 we will also use XCMS for peak-picking, with its own parameter set. These parameters were (like the permissive set) based on an existing set of (extremely liberal) parameters, and produce a similar number of peaks to the permissive MZMine parameters. We use this set of parameters in Chapter 6 because we will use DsDA [136] as a baseline for our methods to compare against, and it uses XCMS internally, so if MZMine produced different results for the evaluation this may not necessarily be fair. This also has the benefit of showing that our results do not depend on the idiosyncrasies of one particular peak-picking implementation.

The parameters for XCMS are given in Table 3.3. It is worth noting that XCMS does not implement ADAP, so we use the older $centwave$ algorithm which also makes use of continuous wavelet transforms. Our alignment step uses $do\_group\_chromatographic\_peaks\_nearest$ which is "inspired by the correspondence algorithm of mzMine" [84] (i.e. the MZMine Join Aligner). This is to make the results more consistent with the MZMine results (and DsDA hardcodes a very similar idea, checking that peaks overlap in RT and are within the ppm m/z used for $centwave$).

3.3 Simulation Using ViMMS

In the forthcoming chapters we will be validating fragmentation strategies partially by using ViMMS [26] to re-simulate experiments so it is important to understand how the re-simulation process works. Effectively, ViMMS lets you "replay" part of a dataset by extracting groups of $(rt, m/z, intensity)$ triples. The master list of these traces can then be

sampled to produce the basis for new simulated LC-MS/MS experiments. In this thesis every time we re-simulate we populate this master list with a single fullscan (a "seed" file) per each run to reproduce. All traces meeting certain criteria (minimum intensity threshold, length) are used in the generated run. This for the most part means the data is replayed exactly as it was observed.

One area of difference occurs when the scan times of the simulation do not match the scan times of the original data. As MS2 scans typically do not have the same length as MS1, re-simulating a fragmentation strategy from a fullscan run (as we do in all cases) will trivially imply that the output scans will appear at different times. Obviously we cannot exactly replay data we haven't observed, so in this case each point in an MS1 scan is linearly interpolated from points in the two neighbouring MS1 scans in the seed data. If such a point cannot be found in one of those scans, the RoI is considered to have ended and it is dropped. The traces themselves are constructed via the $centwave$ [77] RoI-building algorithm [7].

As we mentioned in Section 3.1, it is not necessary to simulate accurate fragmentation spectra to compute coverage and intensity coverage (and doing so would introduce many complications). Thus, using a fullscan for the re-simulation offers a richer view of the data on which to perform these interpolations more accurately.

Because we are replaying the data exactly, there is also the question of whether to seed each run of the same sample type with a single fullscan or to use a different fullscan (of the same sample type) each time. There is some natural variation between each run so the latter procedure is arguably more realistic. However, that variation adds significant complexity when combined with peak-picking — because the number of output peaks will change with the number of fullscans given to the peak-picker even with the same sample type, an experimentalist must now control an additional variable. Specifically we have to ask how results will change as the total number of runs changes. Because it is possible with additional runs that new peaks will be observed or existing peaks will be observed at higher performance, then in the most extreme case the addition of more runs may make performance measured in proportional terms go *down* rather than up. It is significantly simpler to have a single target set that does not depend on the number of technical (rather than biological) repeats. Using multiple fullscans also, of course, requires the collection of multiple fullscans and thus valuable instrument time. This can be a serious overhead when running a large simulation. Some of the consequences of this choice will be illustrated in Chapter 6 when we use both procedures, and this is also why we peak-pick once per run in that chapter.

Another unusual property of re-simulation is that scan times are completely controllable. While on an actual instrument a physical process governs how many ions can be collected

[7] But basically all data is included, so this does not force simulated data into an unrepresentative "RoI-like" structure.

for a scan and thus a certain variable dwell time is required to produce said scan, simulations do not have this constraint. A consequence of this is that we can choose scan lengths in simulation to be fixed constants. We will frequently set scan lengths to the average on a real instrument in order to make it easier to reproduce these results exactly. Another caveat is that long fragmentation strategy processing times after a scan is produced are not usually accounted for in simulation. In a lab experiment this time would typically be wasted, but in a simulation the data generation process can be paused for as long as the simulated instrument controller needs. While most methods we will discuss have very short processing times, some particularly long-running methods will have timings presented in Section 5.5.

3.3.1 Summary

To illustrate how the entire re-simulation and evaluation procedure fits together, an overview is shown in Figure 3.2. To produce re-simulated data, *seed fullscan .mzMLs* are provided to "replay" the data. They are then interpolated according to a *scan schedule* and a *fragmentation strategy*. This scan schedule is a list of RT and MS level pairs and may be produced dynamically on a per-scan basis (e.g. in response to a DDA method) or be set in advance. The dashed arrows from the scan schedule and seed .mzMLs represent that some fragmentation strategies (e.g. pre-scheduled methods) may have some knowledge of the data and/or the scan times whereas others (e.g. TopN) may have none.

These steps produce re-simulated .mzMLs mimicking the output of the fragmentation strategy for a lab experiment of those sample types. These .mzMLs are then evaluated by their

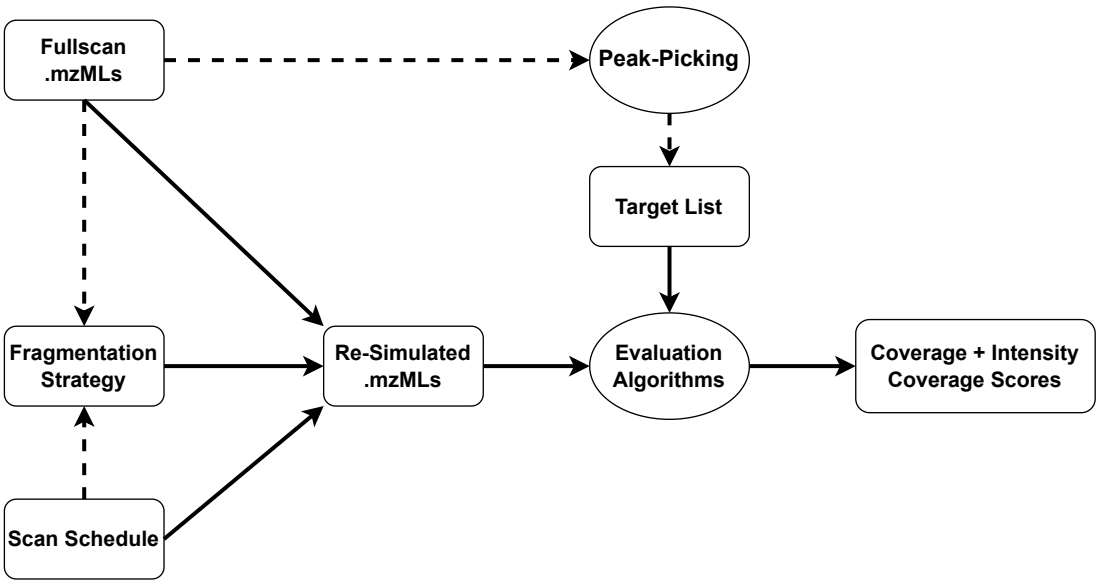

Figure 3.2: Diagram showing data-flow for re-simulation using ViMMS. Dashed arrows represent optional dependencies.

performance against a *target list* which details the targets for which it would be desirable to collect fragmentation spectra. This is used to compute an *evaluation metric*. In our case the evaluation metrics are (peak) coverage and intensity coverage, so the target list contains one $(rt, m/z)$ bounding box per peak we wish to acquire. In our experiments, this list is produced by a peak-picking program with access to the same list of fullscan .mzMLs used to seed the re-simulation. However, the evaluation algorithm will accept a target list created by any other means provided it is formatted as a list of $(rt, m/z)$ bounds (e.g. the list could be created manually by an expert with an existing workflow).

3.4 Datasets

The experiments described in the following chapters primarily use a common dataset of ten store-bought beers. Beer samples are inexpensive to obtain, so this makes it easier to reproduce our experiments. These samples also avoid ethics and biosafety concerns that might be associated with, for example, human samples. However, beer is also complex enough a sample type to produce data challenging for fragmentation strategies. It is also straightforward to obtain "related" samples by taking different beers or different families of beer. As all samples are beer, they should still overlap to some extent, but they should also have the differences between individual types of beer reflected in their metabolite population. This combination of easy obtainability, intrinsic challenge as data and ability to find related samples makes beer data a desirable choice for our investigations.

Table 3.4 lists a total of ten beers. In our subsequent experiments, we will often use only a subset of this list per experiment but most are drawn from this common pool. Generally, each experiment uses the beers with the lowest index available (so if only one beer is used, the Raspberry Sour was used). An exception is the replication experiments we will explore in Section 5.2. The first replication experiment used one of each of the ten beers: the other randomly sampled six beers from the list ten times. For easy lookup of the indices used we list them here in Table 3.5.

3.4.1 TopNEXt Dataset

Due to a long-running data acquisition being more prone to serious failure, data acquisition was split into four batches which were run across four separate days. A first day of runs produced data for another experiment along with testing that new methods would actually run on the instrument through the Thermo IAPI. The second, third and fourth days each ran a subset of the fragmentation strategies tested in Chapters 4 and 5. For all batches some

Index	Name	Type
1	Raspberry Sour by Vault City	Sour
2	Cacao & Hazelnut Broken Dream Twisted Breakfast Stout by Siren	Stout
3	Tennents	Lager
4	Life and Death by Vocation	IPA
5	Silence is... Citra by Overtone	Pale Ale
6	Punk AF by Brewdog	Alcohol-Free IPA
7	The Hop - Single Hop Series Simcoe Edition by Salt	IPA
8	42 DDH Pale Ale by BBNO	Pale Ale
9	Aoraki by Vocation	DIPA
10	Punk IPA by Brewdog	IPA

Table 3.4: A complete list of the beers collected for our experiments. Note that although the specific choices of sample were arbitrary, we varied the class of beer in the "Type" column in an attempt to obtain varied metabolic profiles.

Observation Num.	Beer Indices
1	(2, 1, 8, 4, 5, 10)
2	(9, 4, 5, 7, 10, 1)
3	(3, 8, 9, 5, 1, 2)
4	(9, 4, 2, 7, 8, 5)
5	(5, 6, 4, 7, 9, 2)
6	(7, 4, 8, 9, 10, 5)
7	(8, 5, 10, 2, 9, 3)
8	(7, 4, 5, 10, 1, 2)
9	(8, 6, 10, 3, 7, 2)
10	(7, 1, 2, 10, 9, 3)

Table 3.5: The random samples used for the 10 replications in the 6-4 replicate experiments in Chapter 5.2.

fullscans of the data were also produced as part of quality-control (which were also used for peak-picking and re-simulation for data from that day).

The actual data acquisition was performed by an expert technologist according to the following procedure. Monophasic sample extraction was done by adding chloroform and methanol in a ratio of 1:1:3 of beer:chloroform:methanol (v/v/v) and mixing with a vortex mixer. The extracted solution was then centrifuged to remove protein and other precipitates, and the supernatant was stored at -80°C. Chromatographic separation with HILIC was performed on all samples by injecting 10 μL beer extract with a Thermo Scientific UltiMate 3000 RSLC liquid chromatography system and a SeQuant ZIC-pHILIC column. A gradient elution was carried out with 20 mM ammonium carbonate (A) and acetonitrile (B), starting at 80% (B) and ending at 20% (B) over a 15 min period, followed by a 2 min wash at 5% (B) and a 9 min re-equilibration at 80% (B). The flow rate was 300 μL/min and the column oven temperature was 40°C. Mass spectra data was generated using a Thermo Orbitrap Fusion tribrid-series mass spectrometer controlled by Thermo IAPI via ViMMS. Full-scan spectra were acquired in positive mode with a resolution of 120,000 and a mass range of 70-1000 *m/z*. Fragmentation spectra were acquired using the orbitrap mass analyser at a resolution of 7,500, with precursor ions isolated using a 0.7 *m/z* width and fragmented using a fixed HCD collision energy of 25%. The AGC was set at 200,000 for MS1 scans and 30,000 for MS2 scans. Each beer extract was injected a maximum of 6 times from the same vial before moving to a new aliquot of the same beer extract, in order to minimise over-sampling of the same vial. Over-sampling can introduce re-sampling bias in the data due to differences in the head-space volume, septum degradation, and solvent evaporation with each successive injection.

3.4.2 Matching Dataset

Chapter 6 conducts a different kind of simulated experiment from Chapters 4 and 5, where two different sets of fullscan were used to seed simulated experiments. A different fullscan was used for each run to increase realism, and one set of fullscans was used by the pre-scheduled method introduced there to create a plan, and the other set was used to seed the simulation. This experiment used the same beers shown in Table 3.4, but because the data in Section 3.4.1 had only one fullscan per beer type, it was necessary to recollect data to have one fullscan per run in each of the two sets. No instrument with the appropriate API was available, so it was not possible to run methods directly on the instrument for lab experiments. Seed data was instead collected using a Thermo QExactive Orbitrap mass spectrometer. Additionally, the positive ionisation mode was unavailable, so fullscan spectra were acquired in negative mode with a resolution of 70,000 and a mass range of 70-1050 *m/z*, and the AGC was set at 1,000,000 for MS1 scans. Samples were retrieved from the freezer

(having being previously extracted as described in Section 3.4.1) and details of the liquid chromatography and vial sampling remained the same.

3.5 Conclusion

In this chapter we have introduced much of the basis of the work in the chapters to come. In Section 3.1 we introduced our performance measures, coverage and intensity coverage. Coverage is a proxy for how many candidates we have obtained an identifying fragmentation spectrum for, and intensity coverage is an aggregate measure of these spectra's quality. These measures act as a (complementary) alternative to metabolite annotations which evaluate fragmentation strategy performance more directly from the LC-MS/MS data rather than a final biological evaluation. This allows us to understand fragmentation strategies from a more algorithmic perspective. We have also described how to compute these performance measures.

In Section 3.2 we have described our procedure for peak-picking, which allows us to construct lists of desirable targets to compute coverage and intensity coverage. To minimise reliance on a single parameter set, we have described two alternative parameter sets in MZMine and one in XCMS. The combination of a peak-picker with our performance metrics gives us a high-throughput means to judge a fragmentation strategy's performance on given data.

Section 3.3 describes the process by which we created simulated experiments using ViMMS. MS1 data is replayed from a "seed" fullscan file, with a linear interpolation between adjacent MS1 scans in the seed file when scan times do not align exactly. Finally, in Section 3.4 describes the beer datasets we will use throughout Chapters 4-6, and the details of their collection. The experiments in Chapters 4-6 draw on the same total set of 10 beer sample types. Chapters 4 and 5 form a unit and both simulated and lab experiments were produced for them from a single round of data collection. Although Chapter 6 uses only simulated experiments on the same sample types, a second round of data collection was necessary to produce sufficient numbers of fullscans for the types of experiments conducted.

Having described how data was collected for our experiments, how we ran simulated experiments and how we evaluated our results, we have covered the shared background for our experiments going forward. We can now move forward to introducing novel fragmentation strategies and evaluating them.

Chapter 4

Introducing TopNEXt

Earlier in Section 2.3 we discussed the strengths and weaknesses of various fragmentation strategies. In particular, DDA strategies like TopN have significant difficulty obtaining comprehensive sample coverage. This is due to DDA methods' simple, heuristics-based behaviour which does not retain relevant information about previous choices. Some of the newer DDA methods we mentioned like SmartRoI, WeightedDEW [23] and TopN Exclusion [122] incorporate more context from previous scans or runs, but obtaining sufficient coverage remains a significant challenge for DDA methods. In this chapter we will introduce TopNEXt to address this weakness of DDA methods. TopNEXt is a framework for implementing multi-run DDA methods using a small set of modular features, and is implemented as a submodule of ViMMS. Our experiments will show that TopNEXt can be used to improve sample coverage and acquisition intensities for multi-run (and multi-sample) experiments.

To allow for modularity, TopNEXt defines each fragmentation strategy as a scoring function. The values of these scoring functions can be computed with simple computational geometry, similar to that used in Chapter 3 for the evaluation algorithm (if slightly more advanced). TopNEXt also implements two new concepts which can be used as terms in these scoring functions. The first is "intensity weighting". In a method like TopN Exclusion, exclusion is binary: if a precursor would fall within an exclusion window then it cannot be targeted again. Conversely, with intensity weighting, a DDA method can target MS1 signals even if they would be covered by an exclusion window, as long as the intensity of the MS1 signal is significantly increased from the previous MS2 precursor intensity. Allowing re-targeting in these instances allows the method to increase acquisition quality, and thus the interpretability of fragmentation spectra, particularly over large numbers of runs. We will see that this change leads to increased intensity coverage i.e. it increases the maximum intensity each spectrum is acquired at (see Section 3.1 for a precise definition).

Secondly, "RoI area weighting" allows area-based comparisons between different RoIs (Regions-

of-Interest). In TopN Exclusion, exclusion is again binary for a given precursor, dependent on whether the $(rt, m/z)$ point for that precursor falls within any exclusion window. In TopNEXt, we instead track entire traces of points across RT in real-time via an RoI-building algorithm. Similarly to SmartRoI, decisions are made whether or not to fragment an RoI at its latest point, rather than considering each precursor point in isolation. This allows us to implement RoI area weighting, where the score is weighted by the proportion of the area of the candidate RoI which is not shared by any exclusion window. This allows decisions to be made with slightly more information about the local structure of the data and using a smoother function.

Additionally, existing methods have been retroactively rephrased as scoring functions or terms in scoring functions to allow their use in TopNEXt. For example, SmartRoI and WeightedDEW can be combined with other TopNEXt methods by substituting a term in the main scoring function. Definitions of the TopNEXt methods, including the scoring functions for each, are included in Section 4.1. Section 4.1.5 explains the algorithms used to compute these scoring functions. Furthermore, to demonstrate that TopNEXt is able to improve the number of fragmentation spectra acquired and their average intensity (peak coverage and intensity coverage) we conducted a combination of simulated and lab experiments. Section 4.2 explains how we selected parameters for the DDA controllers in these experiments. The results of these experiments are given in Section 4.3.

This work was published in *OUP Bioinformatics* [27]. Large parts of this chapter (including text, tables and figures) were directly adapted from that publication. I performed most of the design, implementation and analysis. Some of the implementation work (in particular some of the modularisation of SmartRoI and WeightedDEW) and the running of lab experiments were performed by other parties (Joe Wandy and Stefan Weidt).

4.1 Definitions

TopNEXt is a modular extension to ViMMS for defining DDA fragmentation strategies. In ViMMS, DDA fragmentation strategies are implicitly represented by a *scoring function* [1]. For each MS1 scan, this scoring function maps a list of precursor points to a list of scores. The higher the score, the higher the priority of each for targeting with an MS2 scan in the rest of the duty cycle. Typically, each strategy uses the same duty cycle from TopN, with the same global N parameter determining how many MS2 scans should follow an MS1 scan in a duty cycle (provided there are sufficient valid targets). So, if we have $N = 10$ then after each MS1 scan there should be up to 10 MS2 scans, targeting the 10 highest intensity precursors

[1] Although we say that scoring functions are inherent to DDA in ViMMS, they were not formally described prior to this work.

Method	Multi-Sample	RoI DEW	Multi-Sample RoI Exclusion	Intensity Weighting	RoI Area Weighting
TopN					
TopN Exclusion	✓				
TopN RoI		✓			
TopN Exclusion RoI	✓	✓			
Hard RoI Exclusion	✓	✓	✓		
Intensity RoI Exclusion	✓	✓	✓	✓	
Non-Overlap	✓	✓	✓		✓
Intensity Non-Overlap	✓	✓	✓	✓	✓

Table 4.1: A breakdown of which fragmentation strategies incorporate which features. The column "RoI DEW" describes whether the within-run exclusion is tied to RoIs, whereas "Multi-Sample RoI Exclusion" shows whether between-runs exclusion is tied to RoIs. "Multi-Sample" indicates whether it carries over information between runs and " intensity weighting" and "RoI area weighting" show whether the between-runs exclusion uses intensity changes or RoI area, respectively. All methods below the line break are implemented using the TopNEXt framework: those above are implemented elsewhere in ViMMS. The last three strategies in the table implement the primary features introduced by TopNEXt: the others are primarily introduced to control for minor implementation changes introduced by the use of RoIs.

which are not excluded. A precursor is considered to be an invalid target if it has a score of zero or less, which will typically occur if, for example, it is below the minimum intensity threshold.

As a result of having modelled acquisition strategies as scoring functions, we can modularly combine different scoring features by using them as terms in an appropriate scoring function. TopNEXt allows certain aspects of LC-MS/MS data (e.g. RoIs) to be modelled as simple geometric objects and have geometric operations performed using them. We will use this to define useful scoring features based on RoI area and intensity comparisons. These operations are performed on a geometry of points, intervals (1D intervals) and rectangles (2D intervals). These geometric objects might represent, respectively, precursor points, isolation windows and RoIs built in real-time. This shares some commonalities (and some of the underlying implementation) with the evaluation algorithms described in Section 3.1. However, as we will see, some of the operations between rectangles are substantially more complex.

The features TopNEXt provides have been used in combination to implement several improved DDA fragmentation strategies for multi-run experiments [27]. Table 4.1 shows which DDA fragmentation strategies implement which features. In the following subsections, we will describe each in more detail.

4.1.1 Pre-Existing DDA Strategies

TopN is the most basic DDA strategy and all the strategies we will explain subsequently follow the same structure used by its scoring function, so we will describe TopN's scoring function first. Note that this is the same TopN described in Section 2.3 in terms of behaviour, but we are now defining it as a scoring function in order to fit it into the ViMMS framework. The scoring function for TopN can be seen in Equation 4.1.

$$score(p, Ex, \lambda_{min}) = I_{ex}(p, Ex) \cdot I_{\lambda}(\lambda_p \geq \lambda_{min}) \cdot log(\lambda_p) \tag{4.1}$$

This equation has three components. $I_{ex}(p, Ex)$ is the exclusion filter and implements the DEW (see Section 2.3 for details on TopN and the DEW). It evaluates to 0 if the precursor p falls within any exclusion window in the set Ex, and 1 otherwise. Similarly, $I_{\lambda}(\lambda_p \geq \lambda_{min})$ is the user-defined minimum intensity filter, which ensures that all acquisitions are of a usable quality. For a given precursor p, it evaluates to 1 if the precursor intensity λ_p is greater than or equal to the fixed intensity threshold parameter λ_{min}. The third component is the score itself. TopN scores each precursor by its intensity only, so it is simply the log intensity of the precursor, $log(\lambda_p)$. The combination of these three elements causes precursors to be prioritised by highest intensity first, and to be completely excluded if they fail either filter's criteria.

TopN Exclusion is a strategy where TopN can be extended for use in a series of multiple runs and/or sample types by remembering the DEWs between runs. Although the implementation details are different, the scoring function is exactly the same — the contents of Ex simply change to contain different exclusion windows.

RoI-Building

An RoI (Region of Interest) is a rectangular region in $(rt, m/z)$ space indicating the presence of a possible analyte (see Section 2.2.4 for more detail). RoIs are normally constructed as candidate areas for chromatographic peaks in peak-picking, prior to further filtering. By using real-time RoI-building, ViMMS attempts to link the design of DDA fragmentation strategies with the typical way data is processed post-acquisition (again, see Section 2.2.4). As each scan arrives current RoIs are extended and then the collection of active RoIs are considered for potential targeting. In this real-time RoI-tracking scheme, RoIs also subsume the responsibility of DEWs. Rather than checking precursors against a separate list of DEWs, each RoI remembers the last time it was targeted by MS2, and is forbidden from being targeted again for some fixed length in RT. Real-time RoI building is performed using the $centwave$ RoI-building algorithm described in Section 2.2.4. Although the ADAP RoI-builder is in principle more advanced, it requires knowledge of all MS1 scans so it can sort

them highest-intensity first. This is not possible if we build scans as we receive them, so ViMMS uses the $centwave$ RoI-builder which considers MS1 scans in order of RT [2].

Real-time RoI-building provides a foundation for introducing more complex methods in TopNEXt, but the fact that RoIs now take responsibility for dynamic exclusion also has a minor effect on performance. We therefore define **TopN RoI** as a separate method which implements the TopN strategy but which replaces the normal DEW with real-time RoI-tracking. The equation for TopN RoI is shown in Equation 4.2.

$$score(r, Ex, \lambda_{min}) = I_{ex}(r, Ex) \cdot I_{\lambda}(\lambda_r \geq \lambda_{min}) \cdot log(\lambda_r) \tag{4.2}$$

Equation 4.2 is almost identical to Equation 4.1. The sole change is that we have substituted all instances of the precursor p with the RoI containing it, r. This represents that the RoI is now the basic unit of exclusion and targeting, rather than having separate precursor points and DEWs. The list of precursors is still accessible (they are the last point(s) in their RoIs) but information from previous points in the trace is now also available to more complex methods.

Consequently, I_{ex} still handles the exclusion filter, but the semantics have been shifted so that DEW exclusion is handled by the argument r, as we are remembering the last time the *RoI* was fragmented rather than checking if an individual precursor point falls within a previous exclusion window. Ex now only handles exclusion between runs (as now the RoI handles the within-run exclusion). TopN does not remember anything between LC-MS/MS runs in a sequence, so for TopN RoI Ex is always empty. The question of redefining TopN Exclusion to use RoI-building is more complex and leverages TopNEXt so we will address it in Section 4.1.2.

SmartRoI and WeightedDEW

In Section 4.1.2 we will introduce methods that replace the scoring element with a "modified intensity" that lets us build on TopN with more complex reasoning. However, this is not the only way to modify the scoring function. It is also possible to add additional filters or replace the existing ones. For example, we can replace the exclusion filter I_{ex} with a SmartRoI or WeightedDEW filter to hybridise methods. The original SmartRoI and WeightedDEW methods [23] were published as complete, standalone methods — but this is equivalent to replacing the DEW filter in the TopN scoring function.

The basics of SmartRoI and WeightedDEW were explained previously in Section 2.3.1 but here we will focus on their definitions as components of a DDA scoring function — Equation

[2] We are not certain of whether this could imply adverse effects when using ADAP RoI-building to pick peaks for the evaluation, but if so, the effect is very minor.

4.3 shows the filter function for SmartRoI.

$$I_s(r) = \begin{cases} 0 & \text{if } r \text{ has been MS2 targeted and } (\frac{\lambda_r}{\lambda_f} < \alpha) \text{ and } (\frac{\lambda_r}{\lambda_{max}} > \beta) \\ 1 & \text{otherwise} \end{cases} \tag{4.3}$$

Like the DEW function I_{ex}, the SmartRoI filter I_s is an indicator function (i.e. it has binary 0 or 1 output). For an RoI r, λ_r is its current intensity. λ_f represents the highest precursor intensity the RoI has been targeted at previously, and λ_{max} is the highest MS1 intensity belonging to the RoI since that targeting. α and β are, respectively, the parameters dictating how much the RoI must have fallen or risen in intensity since these points. If the RoI has been previously targeted, and neither the rise nor fall is large enough proportionally, the RoI is excluded from targeting in the current duty cycle. This works as a simple drop-in replacement for the standard DEW filter, I_{ex}.

Now let us consider the filter function for WeightedDEW in Equation 4.4.

$$f_{ex}(p, Ex) = \begin{cases} 1 & \text{if } p \text{ is not within a DEW} \\ \frac{t_p - (t_f + d_0)}{d_1} & (t_f + d_0) < t_p < (t_f + d_0 + d_1) \\ 0 & t_f < t_p \leq (t_f + d_0) \end{cases} \tag{4.4}$$

Because it can produce any values in the continuous range $[0, 1]$, f_{ex} is *not* an indicator function and the notation has changed to reflect this. WeightedDEW splits the DEW into two intervals in $(rt, m/z)$ space. d_0 denotes the RT-length of the first interval, and d_1 denotes the length of the second. Both d_0 and d_1 are global parameters chosen by the user. The first interval begins at the RT of its parent MS2 t_f, and the second interval begins as soon as the first ends. Therefore $t_f + d_0$ gives the endpoint of the first interval and $t_f + d_0 + d_1$ gives the endpoint of the second. The output filter value is 0 if the time of the precursor t_p lies within the first interval, and 1 if it falls within the bounds of neither. The linear weighting is applied when t_p falls within the bounds of the second part of the DEW, and is the proportion of how far along it is in RT-terms within that second interval.

However, the fact that the WeightedDEW filter function is not an indicator function causes the position of logarithms in the overall scoring function to become quite important. The log identity $x \cdot log(\lambda) = log(\lambda^x)$ implies that in general precursors will not have the same ranking under $log(f_{ex} \cdot \lambda_p)$ and $f_{ex} \cdot log(\lambda_p)$. Applying the logarithm first implies that RT considerations will more strongly downweight the intensity-based score component — this is useful given the large order of magnitude differences between different peaks. However, this does indicate care should be used when composing more complex RT-filtering and precursor-scoring functions.

4.1.2 TopNEXt DDA Strategies

TopNEXt introduces two major concepts for fragmentation strategies: intensity weighting and RoI area weighting. These concepts are used for exclusion in multi-run LC-MS/MS experiments. All methods in this section behave identically to TopN RoI on their first LC-MS/MS run in a sequence. On subsequent runs, precursors are downweighted if they interact with an exclusion window. In contrast to most of the existing exclusion methods we have described as background, methods using intensity weighting and RoI area weighting perform weighting using a smooth function rather than binary exclusion. They produce a "modified intensity" between 0 and the original precursor intensity. Thus, although scores will be downweighted further as the same region receives more MS2 targetings, it is possible to make more complex decisions about when to revisit these regions when they have changed substantially.

First, though, let us define TopN Exclusion within TopNEXt. Introducing RoI-DEWs creates some ambiguity in what it means to "remember the DEW between LC-MS/MS runs". One option is to perform the DEW-exclusion with RoIs but remember the fixed-sized DEW that a non-RoI scheme would have used, for use in future LC-MS/MS runs. The other is to be consistent and remember the final bounds of the RoI as the exclusion window for future runs. While conceptual consistency is appealing, a change here could have unintended consequences. Therefore, to control for the effect this may have, we define these both as separate methods, **TopN Exclusion RoI (TopNEx)** and **Hard RoI Exclusion**, respectively. Note that, although the implementations are different, these controllers can both be represented conceptually by the earlier Equation 4.2 for TopN RoI. All three handle DEW-exclusion within RoIs, but while for TopN RoI Ex is empty, for TopN Exclusion RoI it contains memories of fixed-size "DEW" windows, and for Hard RoI Exclusion it contains memories of RoIs to use for between-run exclusion. In other words, each exclusion window in Hard RoI Exclusion has the same boundaries as the final boundaries of an RoI that was fragmented in a previous run. Note that the methods we will describe subsequently are assumed to store RoIs as exclusion windows in the same way as RoIs are stored by Hard RoI Exclusion.

An example of how Multi-Sample RoI Exclusion works can be seen in Figure 4.1: some points in the second run fall within the area labelled ab and hence inside a, so under Hard RoI Exclusion would not be considered for targeting.

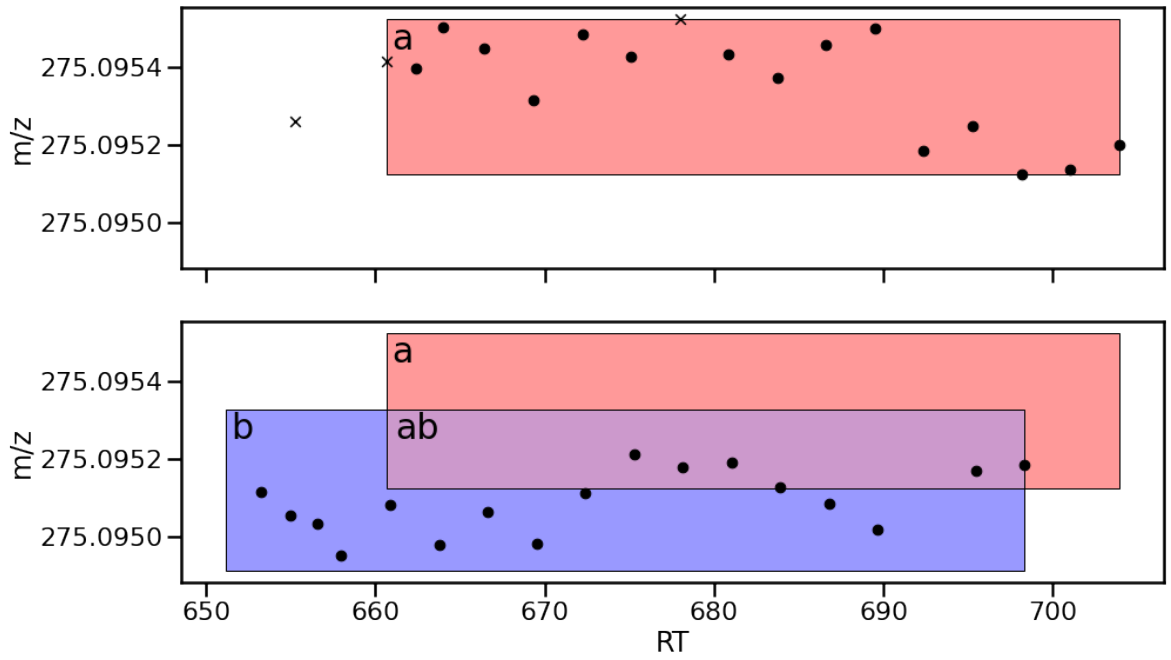

Figure 4.1: An illustration of RoI-tracking when using RoIs as exclusion windows, where from top to bottom each subplot represents a successive LC-MS/MS run. The points are individual observations in MS1 scans. A cross represents an MS2 precursor. On the first run, the RoI a is drawn. On the second run, a persists as an exclusion window, while b is drawn around the new points, forming the overlapping area ab. Note that a and b are drawn here after all points were observed, but as RoIs would be dynamically extended to the right to cover the points as we observed them in real-time.

4.1.3 Multi-Sample RoI Exclusion

Intensity Weighting

Intensity weighting allows retargeting of RoIs that appear across runs even if they would otherwise be excluded, provided the precursor intensity has increased. This allows fragmentation spectra to be recollected for an RoI in the same region of $(rt, m/z)$ space if this would allow a higher-quality spectrum or if it appears to be differentially expressed between related samples. This is similar to the intent behind SmartRoI's α parameter, but between LC-MS/MS runs rather than within the same RoI.

However, rather than a binary exclusion based on a threshold parameter, we replace the precursor intensity in the TopN scoring function with a *modified intensity* which is equal to the difference between the current precursor intensity and the highest MS2 precursor intensity from a previous run. To determine the previous precursor intensity, each candidate precursor is checked against RoIs from previous runs. Its previous precursor intensity is the highest precursor intensity any of those RoIs were targeted at. This defines **Intensity RoI Exclusion** which is shown in Equation 4.5.

$$\begin{aligned} score(r, Ex, \lambda_{min}) =& I_{ex}(r, Ex) \cdot I_{\lambda}(\lambda_r \geq \lambda_{min}) \\ & \cdot max(0,\, log(\lambda_r) - log(\phi(r, Ex))) \end{aligned} \tag{4.5}$$

ϕ is the function which computes the highest previous precursor intensity. If the latest precursor from r is *not* contained in any exclusion window ϕ evaluates to 1, meaning the log intensity will be 0. This means that the modified intensity will be treated as equal to the precursor intensity if the RoI has not been previously targeted, and will be lower otherwise.

Note also that logarithms are applied to the intensities before subtracting them. The log identity $log(x) - log(y) = log(x/y)$ implies this is equivalent to dividing the original intensity values. Again, this is because intensity values span many orders of magnitude. A "minor" upwards fluctuation on a very dominant high-intensity peak may drown out an untargeted low-intensity peak otherwise. This would cause Intensity RoI Exclusion to engage in TopN-like redundant behaviour. The logarithms are therefore placed to ensure that Intensity RoI Exclusion will place low value on intensity increased if they are not increases in order of magnitude (and thus the result of the division will also be low in value).

RoI Area Weighting

Rather than comparing individual precursors to exclusion windows, RoI area weighting compares entire RoIs to the exclusion windows. By comparing two objects that are alike, we produce a more nuanced scoring function and avoid scenarios where outliers are not associated with a previous RoI. As the name "RoI area weighting" suggests, our similarity measure is the overlapping area between the RoI and exclusion windows.

If an RoI shares little or no area with exclusion windows, then it is likely we have not targeted it in a previous run. **Non-Overlap** computes a weighting using the ratio of the candidate RoI's area which does not overlap with any exclusion window over its total area. This can be thought of as "cutting out" any parts of the RoI which overlap with an exclusion window, computing the area of that leftover shape, and then taking its area as a proportion of the original rectangle.

When "cutting out" an axis-aligned rectangle from another axis-aligned rectangle, the result must be a rectilinear polygon (informally, a shape with all sides parallel to a set of Cartesian coordinate axes). Similarly, cutting a rectilinear polygon out of another will result in another rectilinear polygon (rectangles are themselves a special case of a rectilinear polygon). Furthermore, a rectilinear polygon can be represented using a collection of rectangles. For example, either of the input rectangle a's edges protruding into b in Figure 4.1 can be extended to split b into a composite of two rectangles. This has the convenient property that the

area of a rectilinear polygon can be calculated by summing the areas of the rectangles making it up. The area of a collection of rectilinear polygons can also be found by summing the areas of individual rectilinear polygons [3]. We will use the term "subregion" to refer to such a collection of rectilinear polygons corresponding to a particular combination of original rectangles (and not intersecting *any* other rectangles not included in that combination).

Dividing the area of a "cut-out" subregion by its parent rectangle gives us a ratio bounded in the range $[0, 1]$. Once we have this ratio, we multiply the log precursor intensity by it to compute the modified intensity. This is defined in Equation 4.6.

$$prop(a) = \frac{\sum_{i=1}^{|a|} area(a_i)}{area(a)} \tag{4.6}$$

Here $area(a)$ denotes the area of the parent rectangle a (assume this function is only defined for rectangles, so that more complex shapes are always written as a summation of rectangles). a_i denotes the ith rectangle being used to represent the subregion a, and $|a|$ is the number of rectangles used to represent subregion a. The entire Non-Overlap scoring function is given in Equation 4.7.

$$score(r, Ex, \lambda_{min}) = I_{ex}(r, Ex) \cdot I_{\lambda}(\lambda_r \geq \lambda_{min}) \cdot prop(r) \cdot log(\lambda_r) \tag{4.7}$$

Thus, for example, when deciding whether a point in the original RoI b should be fragmented in Figure 4.1, we would raise the intensity of the point λ_r to the power of the blue area b as a proportion of the total area of the original RoI b (i.e. area b where only b is present, plus area ab where a is also present).

Also, once again, the multiplication applies after the application of the logarithm, to stop downweighting being dominated by scale differences from high-intensity peaks.

4.1.4 Combining Area and Intensity Weighting

Although both area weighting and intensity weighting are simple, combining them requires a more complex generalisation. Consider that for any pair of overlapping rectangles, there will be an overlapping rectangle, and up to two rectilinear polygons left over for the non-overlapping areas [4]. If there is a third rectangle which overlaps with both of the others, there is one rectangular overlapping area, and potentially six rectilinear polygons, up to three per each individual rectangle and up to three per each pair.

[3] To see why we would need a collection of rectilinear polygons, imagine a candidate RoI being bifurcated by an identical exclusion window rotated 90 degrees, forming a cross shape. The remaining two rectangles in the candidate RoI could then be cut into some non-rectangular shape.

[4] There will be only one if one rectangle completely envelops the other, and zero if the rectangles have exactly the same bounds.

Clearly the number of unique combinations expands superlinearly. For Non-Overlap we were concerned only with maintaining one of those rectilinear polygons, and it would get continually smaller as more exclusion windows were added. But if we seek to add intensity exclusion, we now care about the intensities associated with those exclusion windows, and how they overlap with the candidate RoI. For example, first consider a case where a candidate is completely enveloped by a low-intensity exclusion window but has a small overlap with a high-intensity window. Then, consider the inverse where the overlap is large with the high-intensity peak and low with the low-intensity peak. For a simple example like this it would be possible to use the value from the highest area of overlap, but in general we are interested in the granularity between many exclusion windows from different runs potentially overlapping our peak.

Intensity Non-Overlap aggregates each of these subregions into a single modified intensity. If we denote our rectangles as $a, b, c, d...$ then the subregion where only a and b are present can be written as ab, the subregion where only a, b and c are present can be written as abc and so on. Given a rectangle a, let $\mathbb{B}$ denote all possible combinations of overlapping rectangles (the power set of all rectangles touching a) i.e. $\mathbb{B} = \{\epsilon, b, c, bc, d, bd, bc, bcd...\}$. Furthermore let $B \in \mathbb{B}$ denote one potential combination of overlapping rectangles. aB then denotes an arbitrary subregion where a overlaps one of the combinations of rectangles contained in $\mathbb{B}$. B also contains a null case where no rectangles are present, denoted ϵ.

Each rectangle also has an intensity associated with it. The intensity of the candidate RoI (in this case assumed to be a) is the intensity of the last precursor contained within it. The intensity of the exclusion windows (any other single letter used in B) is the highest intensity they were fragmented at. The intensity of any subregion with a as a member (i.e. any aB) is defined as the difference between a's intensity and the maximum intensity of any exclusion windows belonging to that subregion, floored to 0 i.e. $\lambda_{aB} = max(0, \lambda_a - \underset{b' \in B}{max}(\lambda_{b'}))$, where b' represents a single exclusion window making up the combination in B. So if $B = bc$, then b' may take the value b or c. When $B = \epsilon$ we have $\underset{b' \in B}{max}(\lambda_{b'}) = 0$. The modified intensity value for the parent rectangle a is the sum of the intensities for each aB weighted by area. This is shown in Equation 4.8.

$$\begin{aligned} score(r, Ex, \lambda_{min}) =& I_{ex}(r, Ex) \cdot I_\lambda(\lambda_r \geq \lambda_{min}) \\ & \cdot log\left(\sum_{B \in \mathbb{B}} max\left(0,\ \lambda_{rB}^{prop(r,B)}\right)\right) \end{aligned} \tag{4.8}$$

The function $prop(a, B)$ gives the area of the particular subregion defined by aB as a proportion of the area of a. Here summing over all $(aB)_i$ means to sum over all rectangles constituting a, including those touching exclusion windows. Note that this is different from

iterating over $b' \in B$. aB denotes the intersection of the original rectangles b' while simultaneously the union of rectangles $(aB)_i$ is used to represent the area of aB.

Furthermore, here the "RoI area weight" is not a weight in the normal sense: it is applied as a power. This matches the definition in Non-Overlap due to log identities — but in order to sum modified intensities rather than multiplying them it is necessary to apply the logarithm at the end. The scoring function is given in Equation 4.9.

$$prop(a, B) = \frac{\sum_{i=1}^{|aB|} area((aB)_i)}{area(a)} \tag{4.9}$$

To decide whether to fragment b in Figure 4.1 using Equation 4.8, b and ab would have their intensity calculated as λ_b and $\lambda_b - \lambda_a$ respectively. Each of these values would then be taken to the power of the ratio between their subregion's area and the total area of b, then finally the results would be summed.

4.1.5 Algorithms

In its most general terms, the algorithmic problem for Intensity Non-Overlap is: given a set of axis-aligned rectangles, map each element B in the power set of combinations of those rectangles $\mathbb{B}$ to a collection of rectilinear polygons which exactly covers the area of B and which intersects none of the other remaining axis-aligned rectangles [5]. In other words, the problem is to report all (non-empty) subregions. Each reported subregion must also track some information from its parent rectangles (intensity values) so we can perform some operation with them (arithmetic). Non-Overlap only requires reporting the subregion where a specific rectangle is present and no others are, and no data needs to be associated with this subregion. This restriction allows more efficient implementations, but for the most part we will treat this as a special case of Intensity Non-Overlap. Hard RoI Exclusion and Intensity RoI exclusion only need to find which exclusion window(s) a given precursor point lies within, so basic point containment logic (similar to Section 3.1.3) can be used.

However, the problem we are interested in has some quirks not captured by this generalisation. There are two separate classes of rectangle: RoI and exclusion window. Exclusion windows are added essentially offline in a large batch (at the end of each LC-MS/MS run) but RoIs are considered online, in a time-critical way. RoIs are also essentially independent of each other: exclusion windows interact with RoIs and each other, but RoIs should only overlap in edge-cases, if ever.

Crucially, whether the problem is online or offline decides our strategy for tackling it. For

[5] While the concept is the same, B here is not paired with an extra box which gets special treatment as a did in the previous section.

a totally offline problem it is possible to use a "global" algorithm which exploits having all the data at once. For example, a line-sweep algorithm [142] sorts the whole set of data and benefits from its orderedness. For an online problem the rectangles are received in batches so it is not possible to efficiently sort them all at once and it may not be useful to do so with an already partially-processed dataset.

A minimal example with only two overlapping rectangles is very easy to solve. The overlapping subregion between any arbitrary number of rectangles is always a rectangle. Once this rectangle is "cut out" of its parents, there are up to two remaining rectilinear polygons. It is straightforward to compute the bounds of the overlapping rectangle, too: its bounds will each be drawn from one of the parents' bounds. Following that, the area of the overlapping rectangle can be computed with the trivial $length * width$ formula and the area of the two non-overlapping rectilinear polygons can be calculated by subtracting that area from the areas of the two parent rectangles. For example, for Figure 2.4, the areas of parent rectangle b and intersecting rectangle ab can be calculated as $length * width$, and then the area of rectilinear polygon b is the area of parent rectangle b with the area of ab subtracted, and similarly for a.

Pairwise comparisons are thus very straightforward, but the problem becomes more complex once three or more rectangles overlap. The overlapping area of all three is still a rectangle (if it exists) but there are up to three rectilinear polygons where two parent rectangles are present, and again three where only one is present. A naive algorithm which separately considers each k-combination of input rectangles, including combinations where it is not feasible for all to overlap, will have 2^n possibilities to consider (the cardinality of the power set for n input rectangles) and will not be feasible at relatively low input numbers, let alone at the hundreds of thousands of rectangles we might have it consider.

An obvious algorithm is to find all subregions where a maximal combination of input rectangles intersects (e.g. with a backtracking search) and then work backwards to assign areas to subregions with progressively fewer parent rectangles. However, at least in a naive implementation, this forces recomputation of the entire chain for every query RoI considered, redundantly performing work for the set of exclusion windows [6].

Another approach we have already mentioned in Section 4.1.3 is to note that a rectilinear polygon can be represented exactly by a collection of rectangles covering the same area. Consequently each set of overlapping rectangles can be transformed into an equivalent set of non-overlapping rectangles. A completely online algorithm might reduce a set of parent rectangles to non-overlapping rectangles by maintaining a set of current non-overlapping rectangles and then updating it with the parent rectangles one-at-a-time. Each overlap between a split rectangle and a newly-added rectangle will produce one rectangular overlapping

[6] For example, suppose multiple query RoIs touched abc. To answer every query, you would have to start from abc and work backwards before being able to answer any question about the query RoI.

area and 0-4 non-overlapping rectangles to cover 0-2 rectilinear polygons. Then, as before, we obtain the areas of each subregion by summing the areas of their constituent rectangles.

This approach also has the nice property that exclusion windows can be pre-processed between LC-MS/MS runs by splitting them to be non-overlapping. Because each RoI is independent, too, this means each comparison during the time-critical online part will be between one RoI and this pre-processed set of non-overlapping rectangles. Because none of that set of rectangles overlap, at most two rectangles (the query RoI and a split exclusion window) can simultaneously overlap. This effectively reduces the query to finding which split exclusion windows the RoI overlaps with, and then performing a pairwise comparison with each, using the same simple method we described previously. The remaining problem to solve for the online step is how to quickly find overlapping rectangles.

Filtering Exclusion Windows

For all controllers we must perform a rectangle overlap or point containment check in real-time for every precursor in a given MS1 scan. An MS1 scan takes around 0.6 seconds on our instrument (see Table 5.6 for approximate timings) so we would ideally like processing time to be on the order of $10^{(-2)}$ or less. However, given that over the course of 20 successive LC-MS/MS runs the number of exclusion windows can reach the 100,000s, and we must search this set of windows for *every* precursor in the current scan, a naive containment check where we simply iterate through every (precursor, window) pair is not practical. Therefore we separate the space into a discrete grid of fixed-size boxes, and each of these grid-box stores which of the exclusion windows overlap it. Simple arithmetic then computes which of these grid-boxes a precursor falls into, and then we need only manually check that the precursor is contained within the small subset of exclusion windows associated with its grid-box. This data structure is updated entirely offline, between runs, when all exclusion windows for that run are added to the total.

A simple example is to imagine dividing the space into four, then for each precursor only checking the exclusion windows overlapping the quadrant the precursor falls into. If the exclusion windows were evenly distributed, then this would divide the number needing to be searched by four. However, when querying an entire RoI for area exclusion it may lie across multiple grid-boxes, and we must take the union of their contents before checking them for intersection. But in our simple example, even if the query RoI lay across two quadrants, then we would still only need to search half the number of exclusion windows for intersection.

By quartering the space, we have split it once on each dimension, but we extend this idea and split it more times to reduce the set of containment/intersection checks further. However, more splits are not always better. Increasing the number of grid-boxes also creates a

memory overhead (and processing overhead in managing that memory) and making the grid-boxes smaller also increases the number of them that exclusion windows will be duplicated between, requiring additional processing for the query RoI to filter them down into a single set.

This is only a heuristic measure with a performance increase bounded from above by some constant factor, and is sensitive to the number of splits and the distribution of exclusion windows. More splits also implies a quadratic growth in the number of grid-boxes and hence overhead. Still, we found this grid performed better in practice than using Python's *intervaltree* package once for each dimension, or the *r-tree* package. Faster implementations may be possible with alternative implementations of these data structures, segment trees or other principled methods of searching 2D space for point containment/rectangle overlap.

One further consideration is that it should also be possible to speculatively calculate which exclusion windows an RoI *could* overlap with if it were extended in the next MS1 scan given we have a reasonable estimate of the next RT (e.g. a maximum cycle time set on the instrument)[7]. In the $centwave$ RoI-builder each RoI is recentred on the mean m/z of its constituents whenever one is added. However, we know the maximum range the next point can lie in to be considered a part of that RoI. Consequently, we also know the maximum distance the RoI can move, and can construct a larger query rectangle using its minimum and maximum possible bounds. When we have the actual query RoI, the (short) list of rectangles returned by this query can be filtered by, say, naive overlap checks. This would reduce the time-critical search component to just this final check while the majority of the search can be performed while the instrument is executing a duty cycle.

Dissecting a Set of Rectangles

At the beginning of Section 4.1.5 we described a completely online algorithm for dissecting our set of parent rectangles into equivalent non-overlapping rectangles, by adding rectangles one-at-a-time. However, since exclusion windows are added in large batches it makes more sense to process them with an algorithm that exploits this. The number of split boxes produced by the simple online scheme will also depend on the order parent rectangles are considered and will not in general be minimal. An offline algorithm has more information available to minimise the number of split rectangles used and thus reduce processing costs for the online stage.

This problem of "cutting out" parts of a shape where it overlaps with others is known in computational geometry as "clipping". Its most notable use is in computer graphics, where the visible parts of a shape are determined by clipping using any shapes further in the foreground

[7] Our implementation does not (yet) do this.

— this saves resources drawing these shapes only for them to be overwritten. Many efficient algorithms are known for general polygons [143, 144, 145, 146], but since they are for general polygons implementations do not leverage the fact that all polygons we are interested in are rectilinear.

In addition to simplifying area calculations, pre-dissecting everything into rectangles has another theoretical benefit. Each rectangle can be represented with exactly two corner points, so should require an equal or lesser number of vertices to represent a rectilinear polygon compared to representing it as a general polygon. However any performance benefit for our algorithm compared to algorithms for general polygons depends on a myriad of factors. Firstly how close to minimality the set of split rectangles is, secondly on implementation details like efficient layout in memory and thirdly on the distribution of the data. These topics are extraordinarily complex and require separate empirical studies to justify, so we shall leave them aside for this thesis. It is known to us that covering rectilinear polygons with a minimal number of non-overlapping rectangles can be performed in polynomial time [147] but we do not make any algorithmic comparisons — the focus of this thesis is on how solving this problem can be used for LC-MS/MS acquisition strategies, and we list the details of our algorithm primarily so our results can be reproduced.

To perform the splitting we use a form of line-sweep algorithm [142]. Line-sweep is a technique common to many geometric algorithms where objects are sorted and iterated over on each axis in turn. Conceptually, this is like sweeping a line across that dimension so that processed data lies behind the line and unprocessed data in front of it. In our case, split rectangles lie behind the line, and parts of exclusion windows we have yet to split lie in front of it.

For each rectangle, there are two endpoints on each dimension where the rectangle begins and ends (i.e. each rectangle is the Cartesian product of two 1D intervals, and each one of its four edges is an endpoint on one of these intervals). We first sort all x-endpoints (i.e. edges with a length on the y-dimension) and then iterate through this ordered list. As rectangles begin, we store them as "active" rectangles, and as they end, we remove them from this storage. Whenever we would update the list of "active" rectangles, we can then sort their y-endpoints (i.e. edges with a length on the x-dimension) and iterate through them similarly. For each iteration through the y-endpoints, minus one, we are able to emit one rectangle, by using the currently active pairs of x-endpoints and y-endpoints as its four edges. If we do this, we will end up with a set of split non-overlapping rectangles — it is not possible for necessary splits to our rectangles to appear in locations where no original rectangle had an edge. However, this will lead to obvious cases where rectangles will be split unnecessarily (and thus, processing overhead). For example, if two rectangles intersected on the x-dimension but not the y-dimension (i.e. a vertical line could be drawn intersecting both, but not a horizontal one), then the one that begins first would be partitioned into smaller rectangles when the other

began (and potentially when it ended) despite this being unnecessary.

To reduce the number of rectangles emitted, we must do bounds-checking of active rectangle y-bounds, and in order to quickly query the active rectangles we store them in an interval tree. However, when we receive a new x-endpoint, it may only partially overlap with other active intervals in the interval tree. In this case, to add the x-endpoint to the tree we will need to split it and any intervals it overlaps with into non-overlapping intervals (assume that the intervals contained in the tree are already non-overlapping). For each interval in the tree our new x-endpoint overlaps with, we can emit one rectangle with y-bounds equal to where the interval and x-endpoint overlap, and x-bounds equal to the last x-coordinate the interval was updated at and the current x-coordinate. The existing intervals touched by this new x-endpoint are truncated so as to not overlap with the rectangles we have just emitted. Finally, the tree is repopulated with new intervals in the space that has been cleared by truncating the existing intervals.

Repopulating the tree comes with an additional challenge, though. For the non-split intervals, it is fairly easy to know when they are active: they have two x-endpoints and if the current time is between them then they are active. The split intervals, however, can be formed by the overlap of several active intervals, and only cease to be active once all of those parent intervals cease to be active. To make repopulating our interval tree filled with split intervals easier, we maintain a second interval tree which contains the original, unsplit intervals. This will quickly tell us which intervals are active at any given time. By performing a y-sweep of those unsplit intervals which overlap the new x-endpoint we can emit new split intervals to repopulate the split interval tree with.

To describe the entire algorithm in the geometric metaphor of the line-sweep, we can think of a vertical line sweeping along the x-endpoints. As it contacts them, a horizontal line performs a vertical sweep only along the length of the new x-endpoint, and after that, the vertical line "fills" the rectangles behind it. When endpoints only partially overlap, these fills split into two. By continuing this procedure to its end, we eventually fill in all the new rectangles i.e. obtain a set of non-overlapping rectangles covering the same area as the exclusion windows.

In our case we typically treat the RT as the x-axis and m/z as the y-axis. However, note that each x-endpoint implies a vertical sweep and the emission of at least one split rectangle. Given RT has a continuous domain, x-endpoints at approximately the same RT may have infinitesimal differences and thus create performance overhead or numerical error for a tiny slice that is otherwise unlikely to affect area calculations. We therefore round all values in the geometry. Values are already represented by finite-precision floating-point values so this in effect simply reduces their precision[8].

[8] You can also visualise this as an evenly spaced grid on the axes, with rounding making the "spokes" of the grid slightly larger.

A further potential optimisation is that each time we are asked to query an RoI, we previously queried it at the last RT, and thus, we already know which split rectangles the RoI overlapped with prior to that point in RT and how large the overlap was. It is possible to store this information and only calculate overlapping areas for the latest "slice" between the current MS1 scan and the previous one. For example, by having each RoI store a hashmap which has the keys be split exclusion boxes, and values be the sum overlapping area (absolute, not proportional) seen so far. We then only need to update these with the most recent slice and then compute Intensity Non-Overlap as normal using the contents of the hashmap.

However, we have not used this optimisation as it is not compatible with the fact that RoIs do not have a stable position in the $centwave$ RoI-builder due to being recentred on the mean m/z each time a point is added. Essentially, most times we receive an MS1 scan existing RoIs will move around on one of the axes and change which exclusion windows they overlapped with at previous RTs. Our running count would therefore not match the current bounds of the RoI. This may still produce acceptable quality results, but this would require more validation. We nonetheless mention this optimisation as it may be useful in future if one uses an RoI-building algorithm (or a similar technique) which does not move RoIs around on the m/z axis.

4.1.6 Worked Example

We will now illustrate the concept of Non-Overlap with a small (not necessarily realistic) example. Figure 4.2 shows an example of three partially overlapping boxes a, b and c. In the left subfigure these are coloured red, blue and yellow respectively, and where they overlap the colours are merged. The right subfigure shows an intensity heatmap where the colour shifts from yellow to orange to red as intensity increases and where overlapping subregions use the highest intensity of their constituents. a has been assigned the lowest intensity, and c the highest. Figure 4.3 follows the same format, but only the subregion used in calculating the numerator of the Non-Overlap score for c is marked in colour, and the others are marked in grey. In this scenario, a and b would be exclusion windows which had their RoI targeted in a previous LC-MS/MS run (and their intensity is the maximum intensity they were targeted at) and c would be a currently active RoI (and its intensity is the precursor intensity of the last MS1 point).

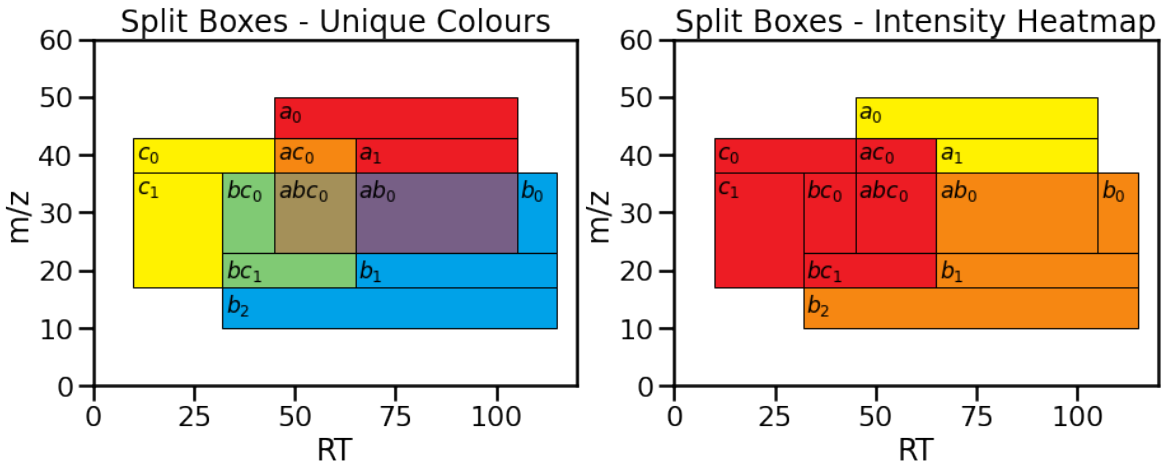

Figure 4.2: *Left:* A dummy example of how three overlapping boxes can be split into subregions and then rectangles, for illustration purposes. The three original boxes are coloured red, blue and yellow, and their shared areas interpolate their colours. *Right*: A heatmap of the intensities of the same boxes.

Non-Overlap only uses the area of the query RoI which is uncovered by exclusion windows to calculate the numerator, so only the single small subregion where only c is present is marked in colour. The rectangles constituting it are labelled on the figures as c_i. The numerator is equal to the sum of the individual areas of each c_i. Conversely, the denominator is the total area of c. This is also equal to the sum of the areas of all rectangles including c in the name but is more straightforwardly obtained from c directly. Letting λ_{mod} be the log modified intensity, $\lambda(c)$ the intensity of c, and all other symbols as previously shown, we have:

$$\lambda_{mod} = log(\lambda_c) \cdot \frac{\sum_{i=1}^{|c|} area(c_i)}{area(c)}$$

$$\lambda_{mod} = log(\lambda_c) \cdot \frac{area(c_0) + area(c_1)}{area(c)}$$

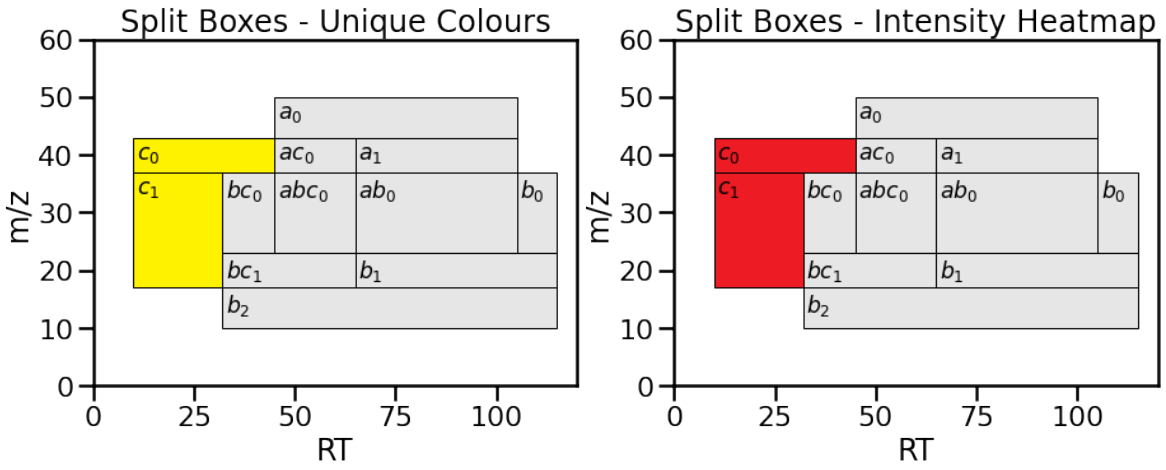

Figure 4.3: Example of how the Non-Overlap score would be calculated for c (the leftmost, yellow box) in Figure 4.2. Anything unused for the calculation of the numerator is marked in grey. Non-Overlap exclusively uses the subregion where only c is present, and uses its intensity unmodified.

Having obtained the Non-Overlap modified intensity value, we need only apply our within-run exclusion filters (DEW/SmartRoI/WeightedDEW) and intensity filter to get the final score.

In addition to these values, Intensity Non-Overlap will also use the subregions containing a and b, subtracting their intensities from the intensity of the subregion only containing c (thus all boxes with c in their label will be used in both the numerator and denominator). To assign intensities to these subregions, for example, λ_{cab} is the difference between the current precursor intensity of the RoI c and the maximum MS2 precursor intensities of exclusion windows a and b. This term may be negative (due to a or b being targeted at a higher intensity) so we floor its contribution to the equation to zero. That the intensities of the overlapping subregions are reduced can be seen in Figure 4.4 — intensities below the value represented by the yellow colour have had their colour interpolated between that yellow colour and grey. Now:

$$prop(c, B) = \frac{\sum_{i=1}^{|cB|} area((cB)_i)}{area(c)}$$

$$\lambda_{mod} = log\left(\sum_{B\in\mathbb{B}} max\left(0,\ \lambda_{cB}^{prop(c,B)}\right)\right)$$

$$\begin{aligned} exp(\lambda_{mod}) = &\ max\left(0,\ pow\left(\lambda_c, \frac{area(c_0)+area(c_1)}{area(c)}\right)\right) \\ &+ max\left(0,\ pow\left(\lambda_{ca}, \frac{area(ac_0)}{area(c)}\right)\right) \\ &+ max\left(0,\ pow\left(\lambda_{cb}, \frac{area(bc_0)+area(bc_1)}{area(c)}\right)\right) \\ &+ max\left(0,\ pow\left(\lambda_{cab}, \frac{area(abc_0)}{area(c)}\right)\right) \end{aligned}$$

Finally, once we apply the filters to this modified intensity we will get the complete Intensity Non-Overlap score.

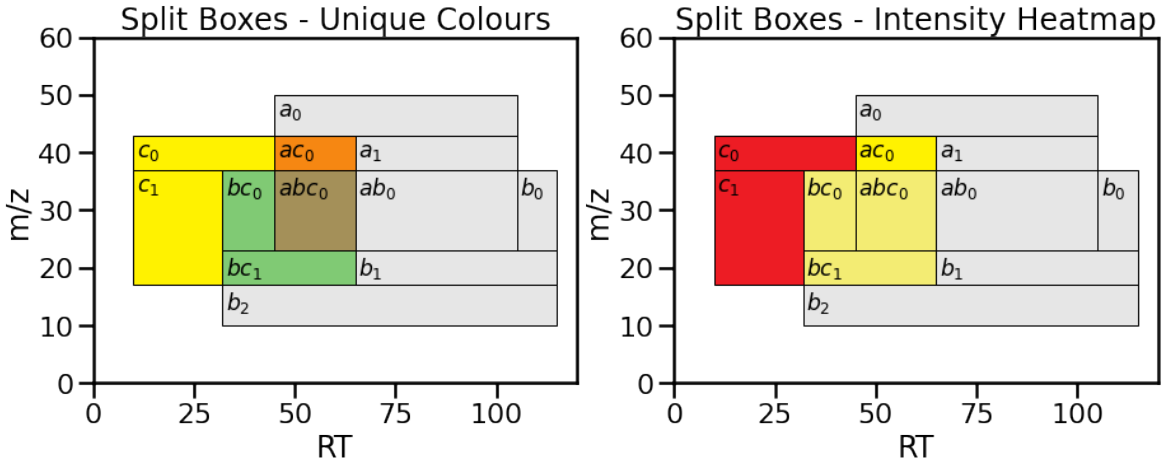

Figure 4.4: Example of how the Intensity Non-Overlap score would be calculated for c (the leftmost, yellow box) in Figure 4.2. Anything unused for the calculation of the numerator is marked in grey. Note that all of the boxes touched by c are used in the intensity area calculation. Also note that the overlapping areas used in Intensity Non-Overlap, but not Non-Overlap, are at decreased intensity compared to Figure 4.2 as they use the difference between c (query RoI) and a and b (exclusion windows) where they are present.

4.2 Parameter Optimisation

TopN and TopN-like DDA methods require the input of many parameters, including the length of the duty cycle N, the size of the DEW, the minimum intensity threshold and so on. RoI-building also requires some parameter choices (e.g. minimum RoI length to not be

ignored and discarded) and SmartRoI and WeightedDEW have several parameters to control their exclusion, too [9]. Although the new TopNEXt controllers do not require any further parameters, it is still necessary to choose plausible values for all controllers.

Shared Parameters	
Ionisation mode	Positive
Isolation width	1
Min MS1 intensity (λ_{min})	5000
mz_tol	10
min_roi_length	3

Table 4.2: Shared controller parameters that were used throughout all our parameter optimisations and experiments. For the real experiments we instead used an isolation width of 0.7 — this is the minimum our instrument supports.

For some parameters a sensible default could be chosen - for example, all controllers used an isolation window width of 1, as this would be unlikely to affect the comparison. These shared parameters can be seen in Table 4.2. However, the optimal values of N and the rt-length of the DEW (and SmartRoI and WeightedDEW equivalents) have a strong impact when comparing different fragmentation strategies and may also impact each other. We therefore used a re-simulated experiment to grid search parameter values — that is, we used re-simulated combinatorially test every parameter combination generated from small lists of valid values for each parameter. Parameters were searched by using an Intensity Non-Overlap variant (we reasoned it would have the most complex behaviour) on a smaller version of the experiment in question (i.e. fewer runs). An experiment using TopN Exclusion to search for parameter values instead can be found later, in Section 5.1. Searched values can be seen in Table 4.3. The final values used for our experiments are highlighted in bold.

Note that some parameters are given different names in the ViMMS code than in the equations. Where applicable we first list the name given in the code and then the symbolic name in parentheses. Note also that the length of the DEW is implemented in a slightly different way than we described in Section 4.1.1: rt_tol expresses the total rt-length of window d_0 plus d_1. Therefore invalid combinations caused by rt_tol having a strictly lesser value than $exclusion\_t\_0$ (i.e. d_0) were excluded from the search and are not listed in Table 4.3. Finally, mz_tol indicates the ppm mass tolerance used for RoI-building/the fixed size of exclusion windows in the case of TopN Exclusion. min_roi_length indicates how many points an RoI must have at minimum before it is discarded.

In total there were 227 searched cases. In each case we had ViMMS generate a set of chem-

[9] There are also instrument-specific parameters like the AGC mentioned in Section 3.4.1

Grid Search Parameter Values	
Standard DEW	
N	1, 3, 5, 10, **20**
rt_tol	15, 30, **60**, 120, 240
SmartRoI	
N	1, 3, 5, 10, **20**
rt_tol	**15**
$intensity\_increase\_factor$ (α)	2, **3**, 5, 10
$drop\_perc$ (β)	0, $\mathbf{10^{-3}}$, 10^{-2}, 10^{-1}
WeightedDEW	
N	1, 3, 5, 10, **20**
$exclusion\_t\_0$ (d_0)	**1**, 10, 15, 30, 60
rt_tol ($d_0 + d_1$)	15, 30, **60**, 120, 240

Table 4.3: Searched parameter values for each DEW variant. All combinations for a given variant were tried: marked in bold are the highest-scoring values, which we used for our actual experiments.

ical objects from the real fullscan .mzMLs generated for beers 1, 2 and 3. Then for each parameter combination, we ran a re-simulated experiment with 6 total runs following the order 1-2-3-1-2-3. All of these parameter combinations were then sorted by their proportional intensity coverage, and the parameter values used in our actual experiments were the ones which showed the highest proportional intensity coverage in this grid search. The fullscans used were generated as part of our lab experiment on the first day, and thereafter we used the parameter values optimised on these. Additional fullscans were generated for use with the fragmentation runs on other days, but the length of the optimisation procedure prevented these being used to optimise parameters for the data generated on the same day.

4.3 Results

Our main experiments testing TopNEXt will use a set of six beers drawn from the larger pool of 10 described in Section 3.4. The important facts for these experiments are that these are complex biological samples which produce dense LC-MS/MS data, but all being beers there should be some similarity between these outputs. Nonetheless, a full list is given in Section 3.4.1. The controller parameters are as given in Section 4.2, and the peak-picking uses the "restrictive" set of parameters from Section 3.2.

We will show both re-simulated and real results and we will test both repeated runs of the same individual beer sample type (multi-run, single-sample) and runs of different beer sample types, with repeats (multi-sample). The re-simulated experiments are low-cost to run

and so allow a more detailed comparison on methods, whereas running them on an actual machine gives a realistic experimental validation.

For a single-sample multi-run experiment, roughly the same peaks are encountered each time at the same *m/z* and RT position in a run, causing it to be relatively more predictable and straightforward for a fragmentation strategy. A multi-sample experiment may have sample types with partial overlap in metabolites, and accounting for this may allow them to be targeted less redundantly compared to individual single-sample, multi-run experiments. However, this also means optimally targeting peaks in the correct sample types is more challenging. Running both of these scenarios in simulation and in the lab gives a total of four experiments. The first aim of these experiments is to show that multi-sample exclusion methods (particularly TopNEXt methods) continue to improve in coverage as more runs are performed, in contrast to TopN. The second is to show that intensity methods can continue optimising intensity coverage with more runs even after coverage stops increasing.

Sections 4.3.1 and 4.3.2 contain simulated results combining all strategies built on top of TopNEXt with the three DEW variants (regular DEW, SmartRoI, WeightedDEW). We also present TopN, TopN RoI and TopN Exclusion as baselines. To differentiate the non-RoI implementation of TopN Exclusion with the RoI-based implementation within TopNEXt we denote them "TopN Exclusion" and "TopNEx" respectively. These simulated results were produced using the fullscans from our fourth day of lab experiments, and MS1 and MS2 scan lengths were fixed to be 0.59 and 0.19 seconds respectively — the average times from the same instrument in Table SI-6 of the original SmartRoI publication [23] — so they could be more exactly reproducible.

Sections 4.3.3 and 4.3.4 contain the results of the lab experiments. These have a significant instrument time cost to run, so for the multi-run experiment (during the third batch of our experiments, see Section 3.4.1 for details) we only present comparison of TopN Exclusion, Non-Overlap and Intensity Non-Overlap. These three were chosen to compare performance of an intensity method to a non-intensity TopNEXt method and a baseline method as sample coverage becomes exhaustive. For the multi-sample experiment we tested all the main variants — TopN Exclusion, Non-Overlap and Intensity Non-Overlap in the second batch, and TopN, Hard RoI Exclusion and Intensity RoI Exclusion in the fourth. This shows performance on a complex and realistic scenario. In the lab experiments, all TopNEXt methods use WeightedDEW exclusion. WeightedDEW was found to have the best performance when optimising parameters on all three in simulation (Section 4.2).

4.3.1 Single Sample Repeated (Simulated)

Figure 4.5A contains results from re-simulating an experiment of the same beer repeated 20 times. As expected, TopN is a completely flat line which does not improve beyond seeing

the same sample type once as no RT noise was introduced during simulation. The other controllers are all roughly competitive on coverage, with the gap being at most around 2% between the best and worst performing variants. The best performing variants are the different implementations of TopN Exclusion, and after ten runs most methods have converged to near-complete coverage of the sample type. Despite gaining coverage the fastest, in intensity coverage the TopN exclusion variants perform the worst by a significant margin, which increases up to around 3% behind the worst new TopNEXt-based method, Hard RoI Exclusion. Intensity RoI Exclusion has significantly better intensity coverage than any non-intensity method and Intensity Non-Overlap is again better than Intensity RoI Exclusion by a significant margin, performing the best on this metric. For most methods in this example, but particularly the intensity methods, the SmartRoI variants are especially effective, with Intensity RoI Exclusion being approximately 6% higher in intensity coverage compared to Non-Overlap, and Intensity Non-Overlap being approximately 5% beyond that. In total In-

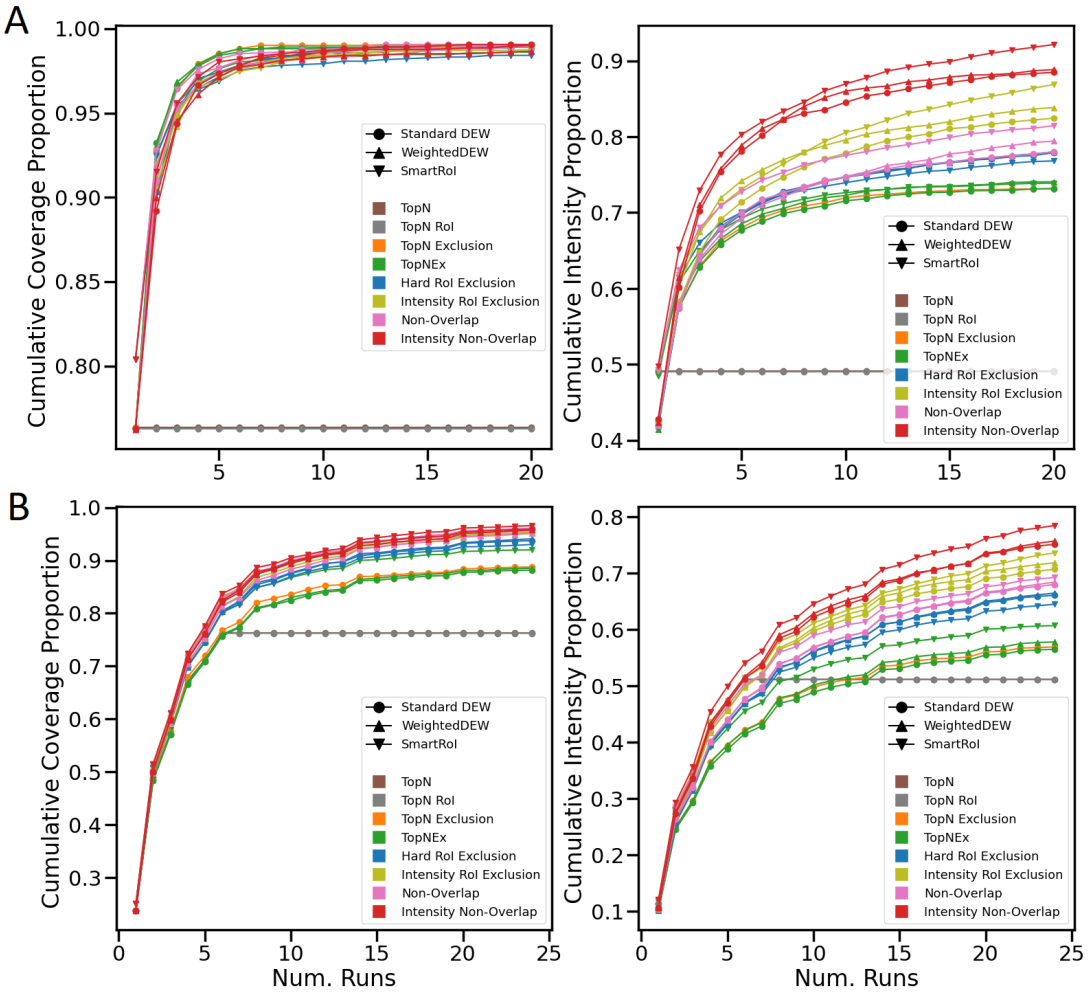

Figure 4.5: Comparison of new TopNEXt DDA methods vs baseline DDA methods in terms of coverage and intensity coverage. A: simulated experiment with the same beer repeated for twenty runs. B: simulated experiment with six different beers each repeated four times.

tensity Non-Overlap has an intensity coverage of 92%, a nearly 20% difference ahead of the nearest TopN Exclusion variant. Importantly, we can also observe that the TopN exclusion variants plateau in intensity coverage shortly after doing so in coverage, and that intensity methods especially maintain a significant slope even as their coverage does not, plateauing later in the process.

4.3.2 Multi-Sample (Simulated)

For the multi-sample experiment we repeat six unique beers (labelled 1-6) four times each for each controller, in round-robin order (i.e. 1-2-3-4-5-6-1-2-3-4-5-6...). This means that at any point no beer sample type has been repeated more than once more than another. This should allow the strategies to firstly collect all shared metabolites, then collect those exclusive to some sample types later, rather than potentially missing them permanently. When this ordering is applied to the lab experiment, it has the potential of causing experimental issues on the real instrument (e.g. through retention time drift) but we decided the benefit of additional collection opportunities outweighed the risk.

Figure 4.5B shows the results of this 6-4 (6 sample types, 4 repeats) experiment. Once again, TopN stops gaining any coverage or intensity coverage once new beers cease to be introduced, and the multi-sample methods all have a significant advantage over it. However, the methods rank differently in coverage this time, and while all new methods are roughly competitive in this respect, most of the TopN exclusion variants trail by a significant margin of around 4% behind the least effective of these methods, Hard RoI Exclusion. TopNEX SmartRoI is very close to Hard RoI Exclusion, but it still nonetheless ranks below all of the new methods. The intensity methods perform *best* on coverage here, with Intensity Non-Overlap being the best controller overall, with around 8% increase in coverage from the baseline TopN exclusion implementation to Intensity Non-Overlap SmartRoI. The differences in intensity coverage remain mostly similar to the same beer experiment, with Intensity Non-Overlap SmartRoI being roughly 21% ahead of baseline TopN Exclusion.

4.3.3 Single Sample Repeated (Lab)

In the simulated results given in Sections 4.3.1 and 4.3.2, coverage was often exhausted significantly before 10 runs, so we ran only 10 runs for the multi-run experiment on the actual instrument. Figure 4.6A shows that all the multi-sample methods are competitive on coverage, with Intensity Non-Overlap being the lowest by a slight margin (as it is focusing on reacquiring peaks at higher intensities, i.e. intensity coverage). Both overlap methods have significantly better intensity coverage (with Intensity Non-Overlap having a further advantage) — but most notably it can be observed that the curves of TopN Exclusion and

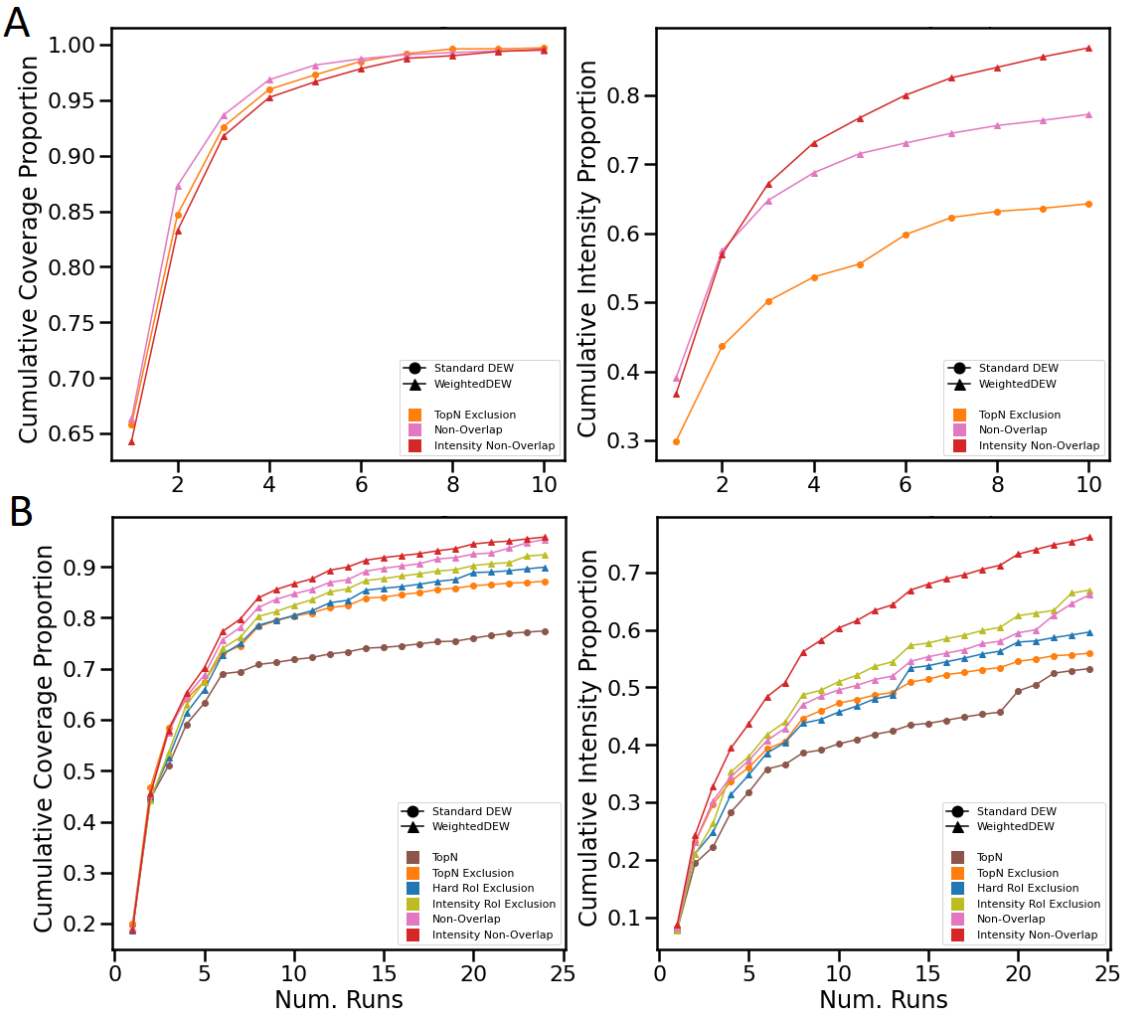

Figure 4.6: Comparison of new TopNEXt DDA methods vs baseline DDA methods in terms of coverage and intensity coverage. A: lab experiment with the same beer repeated for ten runs. B: lab experiment with six different beers each repeated four times.

Non-Overlap flatten as they run out of new peaks to acquire, but the Intensity Non-Overlap curve flattens at a decreased rate. This demonstrates the advantage it has in continuing to reacquire peaks at higher intensities even once coverage gains cease.

4.3.4 Multi-Sample (Lab)

The 6-4 experiment was the most complex and representative of real-use, so we ran this again without changing the setup of runs or beers from simulation [10]. Figure 4.6B shows that in both coverage and intensity coverage TopN is again the weakest of the methods and TopN exclusion trails behind the new TopNEXt methods. The new methods are competitive with each other in terms of coverage: the "intensity" methods clearly improve the intensity

[10]However, the comparisons of RoI and non-RoI methods have been dropped.

coverage by a large margin. There are some particularly large spikes in intensity coverage (especially around run 20) which are likely a consequence of noise in the instrument or the sample itself. The overall trend, however, reassuringly matches the simulated results.

4.4 Conclusion

In this chapter we have introduced TopNEXt, an extensible framework for implementing novel DDA methods. The main features of TopNEXt are intensity weighting and RoI area weighting. Intensity weighting allows an approximation of the value of redundantly targeting peaks across different LC-MS/MS runs to be considered against the value of obtaining a new peak by comparing a function of their intensities. RoI area weighting performs similarly but allows "peak-like" Regions of Interest to be compared for similarity by using their area, allowing more dissimilar peaks to be targeted more often.

We have also shown through experiments on a collection of beer samples introduced in Chapter 3 that DDA strategies using these features can obtain increased performance. Use of either intensity weighting or RoI area weighting allows collection of similar or greater numbers of unique fragmentation spectra at higher intensities. We saw that by using both during multiple runs of a single beer sample type, we ultimately collected a similar number of spectra at close to 20% more of the total intensity. For multiple sample types, the improvements were especially pronounced: we collected nearly 10% more of the total spectra at up to 20% more of the total available intensity. However, we would expect that when intensity methods reacquire a spectrum at higher intensity they also lose an opportunity to acquire a new spectrum. Indeed, in the multi-run same beer results, although the final coverage is nigh-identical, it rises slightly more slowly. But in the mixed-beers 6-4 experiment we saw that the intensity methods in fact have slightly better coverage compared to their non-intensity counterpart, and Intensity Non-Overlap has the highest overall.

What causes the coverage increase? In the original SmartRoI publication [23], SmartRoI and WeightedDEW exchanged some intensity coverage on certain spectra for increased overall coverage against TopN. Although in theory all the scans can be allocated for optimal (intensity) coverage, in a noisy real-world process some degree of redundancy may be desirable. We only see the TopNEXt coverage increase in the multi-sample case, so one potential explanation is that peak-picking has detected different peaks across sample types in similar portions of the space, allowing extra coverage when revisiting these locations. A similar argument could be made for SmartRoI variants performing best in our experiments. SmartRoI produces fewer fragmentation events overall — see the "efficiency" metric in Table 2 of the SmartRoI publication [23] — so it may create exclusion windows less prematurely in a multi-sample context. This would allow these locations to be visited later compared to other

DEW methods, when they would be most relevant. This is one of several questions we will explore further in Chapter 5.

One further point of interest is that in the lab experiments we have used the WeightedDEW variants. This is because while optimising parameters we found that the intensity coverage ranking was in the order WeightedDEW, regular DEW, SmartRoI. However, in the final results, *SmartRoI* consistently performed best. Parameter optimisation did significantly improve results across all controllers (mainly due to increasing the value of the "N" parameter from its initial value of 10) but it was not an exhaustive search. As this procedure can be quite time-intensive, the individual domains for the parameters we considered were relatively small, and the 3-2 experiment might behave differently from a 6-4 or 1-20 setup. This is likely the cause of the different performances of SmartRoI and WeightedDEW. This may indicate that WeightedDEW is more effective on lower run numbers, and SmartRoI on higher run numbers, but we leave exhaustive comparisons to confirm this to future work.

Overall, these results demonstrate that TopNEXt is able to partly address DDA's traditional weakness in obtaining comprehensive sample coverage. It may therefore replace TopN in multi-sample workflows, to provide better (intensity) coverage and therefore greater insight into samples of interest. Future work might involve adaptation to different experimental contexts. For example, rather than only being interested in the absolute number of peaks we can acquire spectra for, a specialised method for a case-control setup might value having pairs of spectra from both case and control. Alternatively, we also use RoIs as "peak-like objects" which can be compared for similarity against exclusion windows, but we could instead use a different RoI building algorithm or a more complicated similarity measure than shared area. These developments or others could be flexibly switched out in the TopNEXt framework with the rest of the procedure working as before, easing future fragmentation strategy development. Additionally, the TopNEXt family of methods can be used as-is through ViMMS to perform metabolomics experiments if separate bridging code exists between ViMMS and the mass spectrometry instrument model. This is currently limited to instruments exposing a Thermo Fisher IAPI, but in future bridges to other instrument models may be created.

Chapter 5

Is TopNEXt Robust?

In Chapter 4 we introduced TopNEXt and showed through experiment that DDA methods using it could obtain more coverage and intensity coverage than standard methods like TopN and TopN Exclusion. However, there are many variables which could change the outcome. Could it be the case that if we were to test TopNEXt on another dataset, or changed the peak-picking parameters, we would get a completely different outcome? In this chapter we aim to address this question.

Each section investigates the configuration of a different part of the setup. Each DDA controller in the previous experiment was given the same parameter values, so Section 5.1 tests different values. Section 5.2 contains a large replication experiment to show that results did not depend on the choice of a specific beer or ordering of beers. Section 5.3 exchanges the previously used "restrictive" set of peak-picking parameters for a harder (though likely noisier) "permissive" set. Section 5.4 goes a step further than Section 5.2 by re-running the original experiment from Chapter 4 with completely separate beer and urine data from the literature with both parameter sets from Section 5.3. Finally Section 5.5 reports inter-scan delays for the lab experiment from Chapter 4 as processing delays may affect the interpretation of results.

This work was included in the Supplementary Information of the TopNEXt publication [27] in *OUP Bioinformatics*. Much of this chapter, including text, tables and figures, was directly adapted from that publication.

5.1 Reoptimised Parameters

The fragmentation strategies we have described all use the same parameter set as TopN, so in all our experiments we fixed them to have identical values to each other. However, it is possible that the individual behaviours of each controller predisposes one to prefer a

different parameter set over another. In our parameter optimisation step (described in Section 4.2) we chose to optimise parameters for Intensity Non-Overlap, reasoning that as the most complicated controller, it would have the most complex interactions with parameters and thus would likely be the most strongly affected by the choice.

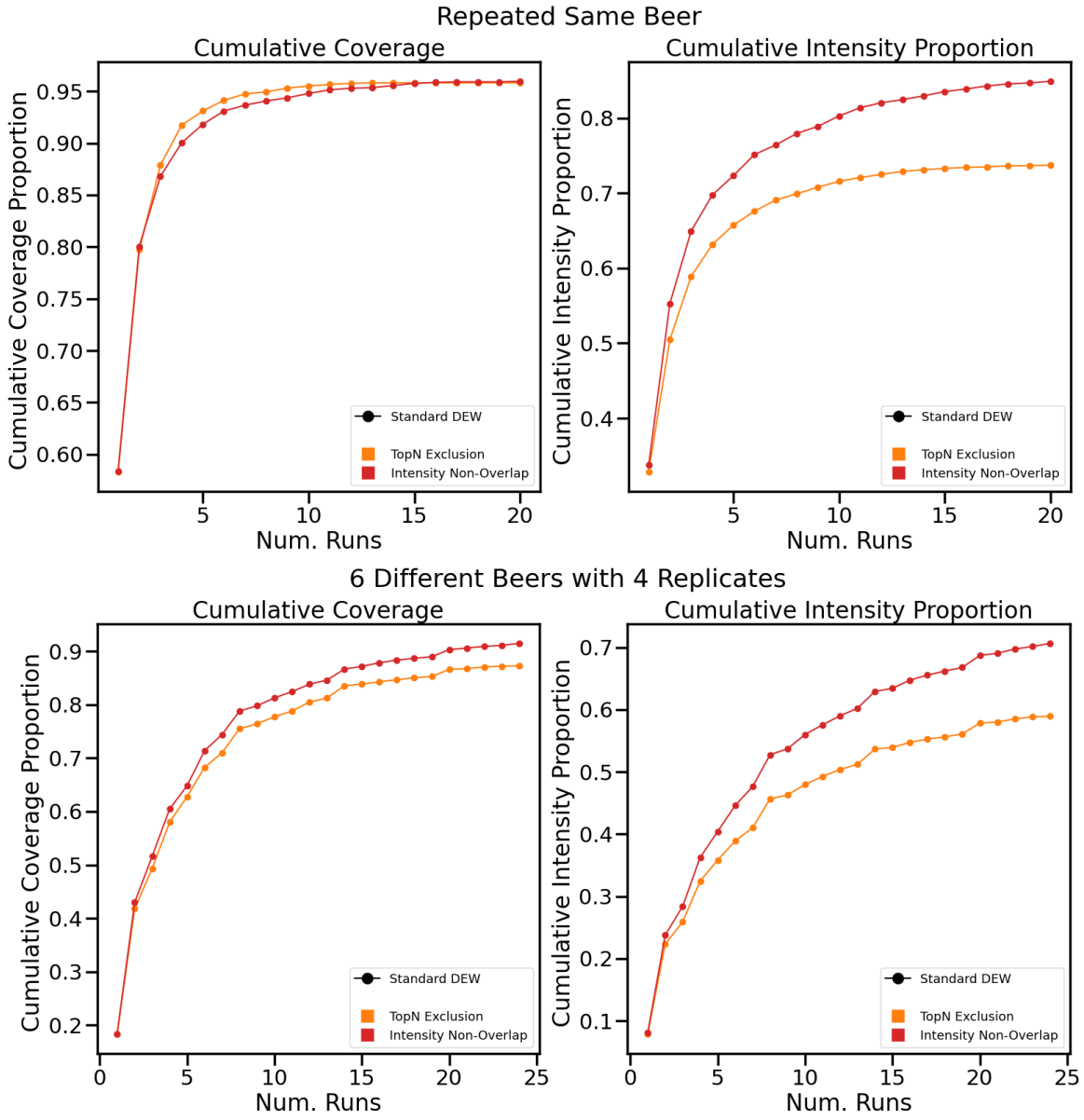

Figure 5.1: Simulated comparison between TopN Exclusion and regular DEW Intensity Non-Overlap using optimal values for TopN Exclusion and restrictive peak-picking. Top: same beer repeated. Bottom: 6-4 beers.

However, our baseline, TopN Exclusion, is the most dissimilar controller from Intensity Non-Overlap (TopN notwithstanding). It is therefore possible that this optimisation procedure favours Intensity Non-Overlap at the expense of TopN Exclusion. To control for this possibility we performed the parameter optimisation again but optimised for TopN Exclusion

this time. The optimal value of $N = 20$ did not change, but TopN Exclusion instead had a slight preference for $rt\_tol = 30$ rather than $rt\_tol = 60$.

Figure 5.1 shows the result of a (re-simulated) comparison between TopN Exclusion and Intensity Non-Overlap when both use $rt\_tol = 30$. From it, we can observe that the change in parameters does not significantly change the results of the evaluation. Both methods are still competitive in coverage on a single sample type repeated, obtaining effectively exhaustive coverage. In the multi-sample case, Intensity Non-Overlap still obtains slightly more coverage. In both cases, Intensity Non-Overlap obtains significantly more intensity coverage.

It therefore appears to be the case there are only minor differences between these fragmentation strategies for optimal parameter choice. A higher value of N is effectively always favoured for any realistic value, and the significance of rt_tol is low enough that an attempt to assign meaning to it would likely be overinterpreting our data. For the rest of the chapter we will use our default setup of $rt\_tol = 60$.

5.2 Replication Study

Although the main results from Section 4.3 are very promising, they consider only two sample type orders — Beer 1 run repeatedly, and Beers 1-6 run in round-robin order. It is hypothetically possible that the results are a product of the specific sample types we used, and will not generalise even to our other beer sample types. To eliminate this concern we have also performed a simulated replication experiment. We repeated each experiment using ten times and plotted the distribution as boxplots in Figures 5.2 and 5.3.

The choice of sample type used for the same beer experiment is simple — we have ten beers, so we repeat a different one each time. Because we only display the final value, each replication used ten runs to allow comparison to both the previous simulated and lab experiments. For the repeated beers experiment, there are 720 possible permutations [1] of beers, so we randomly sampled six beers and ran them in the same 6-4 order as before for each of the ten replications. The specific orders drawn are listed in Table 3.5 in Chapter 3.

The same beer replication experiment is shown in Figure 5.2. All methods except TopN finished with near-total coverage, and spread is minimal due to all methods approximately converging on this point. We can also observe that the final intensity coverage for the intensity methods significantly jumps for both intensity methods, especially for Intensity Non-Overlap, which spikes by around 10% compared to non-intensity methods. The spread on intensity coverage is slightly larger but still quite small. The size increase is expected given performance has not totally converged, but the variation in results between different beer sample types represented by this spread is overall quite small.

[1] Sampling permutations also eliminates run order effects.

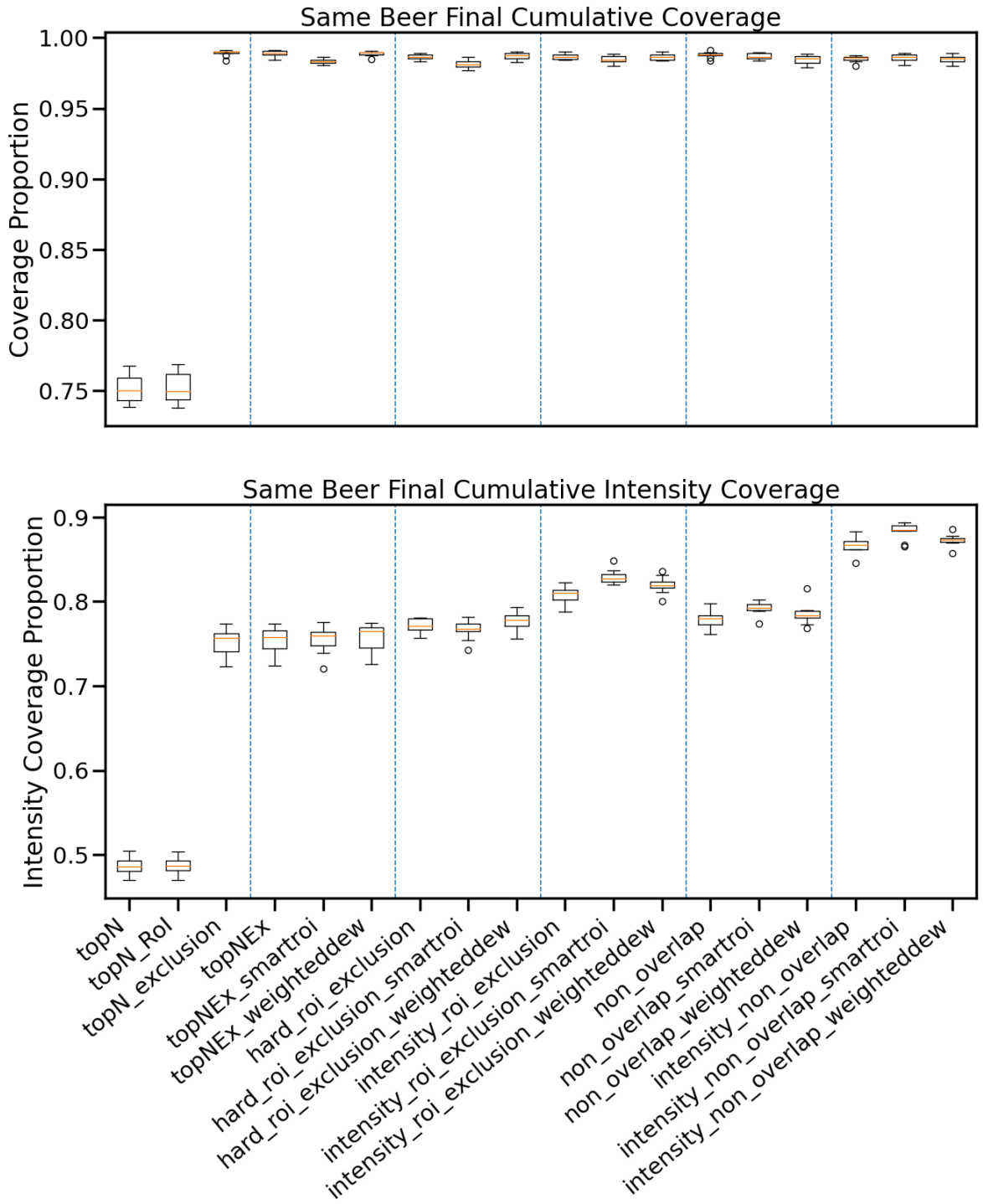

Figure 5.2: Simulated replications of an experiment of ten repeats of a single beer. Each box shows the spread of ten replications for a given controller being run on these samples. Each replication used a different beer sample type to repeat ten times within that experiment.

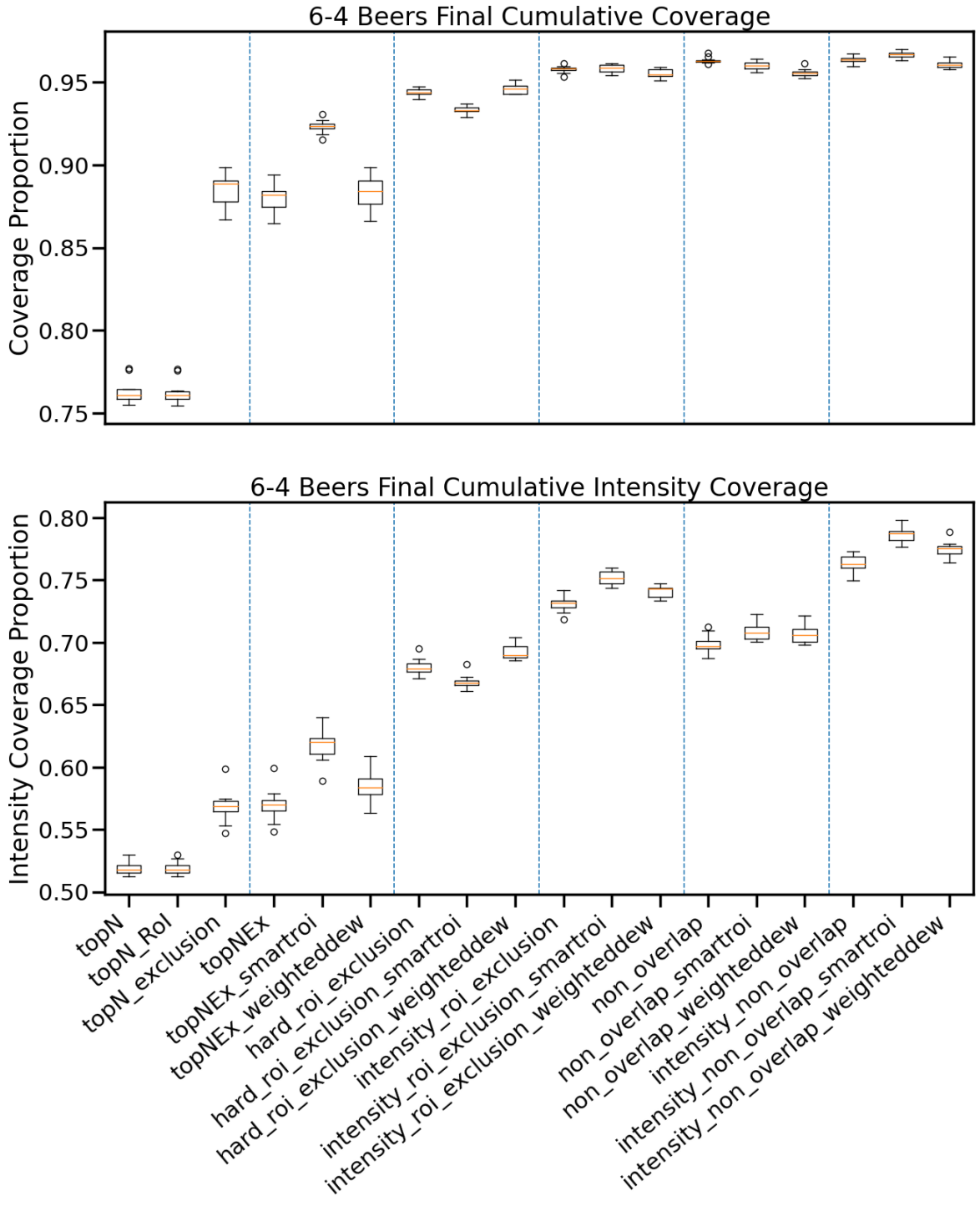

Figure 5.3: Simulated replications of an experiment of ten repeats of a single beer. Each box shows the spread of ten replications for a given controller being run on these samples. Each replication randomly sampled six beers sample types in a random order from a shared set of ten beers, and ran them four times each in round-robin order.

The different beer experiment is shown in Figure 5.3. As in the main experiment, we observe that TopNEXt methods offer a substantial improvement over TopN Exclusion, with TopNEx SmartRoI being a notable outlier which almost reaches the coverage of the weakest TopNEXt method. Coverage also increases as TopNEXt features are added. We also see again the intensity methods perform best at intensity coverage, and that the difference between Intensity Non-Overlap and the TopN Exclusion variants is almost a fifth of the total intensity coverage that can possibly be obtained. Spreads are again extremely narrow for values nearing 100% and overall quite narrow otherwise. Some spreads are a bit wider — TopN Exclusion variants vary by around 10% in some cases, but even the best case does not reach the performance of most of the TopNEXt methods.

This experiment is quite large-scale — running these experiments produced $1800 + 4320 = 6120$ .mzMLs across forty-seven hours, using 20 computer cores. For comparison, if we were to run these experiments on our mass spectrometer at 26 minutes per .mzML produced, assuming no downtime between runs, then these experiments would take a total of 110.5 mass spectrometer days. Performing an experiment of this many cases almost necessitates the use of simulation. Seeing that the results tightly reproduce under all of these conditions we can therefore be confident that they hold across our selection of beers.

5.3 Alternative Peak-Picking

Now we examine the effect our choice of peak-picking parameters has on the results. Peak-picking methods are known to be quite sensitive to parameter choices, and the peak-picking is used to construct the entire set of target peaks for the evaluation, so parameter choices may be quite impactful. We will therefore re-examine the main results, produced with the "restrictive" MZMine parameter set, by using the "permissive" parameter set instead (see Section 3.2 for details on the two parameter sets). The values of the restrictive set were chosen in discussion with our mass spectrometry expert and by design filter out a lot of noise, though they will still likely produce significantly more peaks than there are actual annotatable metabolites. Using the permissive parameter set defines a much greater number of peaks as interesting. This creates a much denser and harder problem, but some of those additional "interesting" peaks may represent a low-intensity metabolite. And if we could get 100% coverage on the permissive parameter set, we would have done so on the restrictive set as well. Therefore we treat this as a proxy for the behaviour of our controllers on sample types that even with a restrictive parameter set still end up dense with interesting peaks.

To give an idea of the actual numbers of peaks involved, restrictive peak-picking on the first beer only produced 2148 identified peaks, but permissive parameters produced 10939. Similarly, for the six beers used in the main 6-4 experiment, restrictive peak-picking parameters

produced 6490 identified aligned peaks, and permissive parameters produced 22516.

Figure 5.4 shows the result of the lab experiments when using the permissive set of MZMine parameters for the evaluation. In the experiment with 10 repeats of the same beer (top), TopN Exclusion has a noticeable coverage advantage of around 5% over Intensity Non-Overlap, and the intensity coverage disadvantage has shrunk to around 8%. A similar change can be observed with the 6-4 beers (bottom), where TopN Exclusion has a coverage lead of up to around 3% throughout most of the experiment, but is eventually barely overtaken by the overlap methods near the end of the experiment. Intensity Non-Overlap finishes with only around 7% more intensity coverage.

Additionally, we can see that (as we would expect) scores for all methods are lower compared to the restrictive parameter set. For the same beer experiment with the restrictive parameters, all methods quickly obtained comprehensive coverage, with TopN Exclusion effectively no longer increasing in coverage by the 8th run. With the permissive set, TopN Exclusion (the method with the highest coverage) finishes only slightly above 80% coverage. With the restrictive set, Intensity Non-Overlap had around 86% intensity coverage, but here has only approximately 63%. A similar result can be seen for the 6-4 beers. TopN has only approximately 49% coverage compared to the approximately 77% it was able to obtain with the restrictive parameters, and has only roughly 35% intensity coverage compared to the restrictive set's 53%. Similarly the strongest methods only break 70% coverage and 60% intensity coverage compared to the thresholds of 90% and 70% seen with the restrictive parameters.

It is clear from the results in Figure 5.4 that the permissive parameter set makes the coverage problem significantly harder, and that it gives TopN Exclusion a coverage advantage compared to the non-overlap methods — the performance of the non-overlap methods relative to each other has not changed much. The intensity coverage advantage of the new methods shrinks as well, but it is important to note that intensity coverage is not independent of coverage: if a peak is not covered, then it counts as having an intensity coverage of 0%.

Thus we recompute intensity coverage to only include those peaks which the method successfully covered in the calculation of the denominator. The normal calculation includes any peak that appeared targetable above the intensity threshold — if it was not targeted above the threshold then it is given a score of 0. This alternate calculation removes those with a score of 0 entirely (so they do not contribute to the denominator) and reports the average target intensity for only those peaks we have targeted so far, rather than the average for the whole dataset.

This is shown in Figure 5.5. For the same beer case (top) we see that the difference in this form of intensity coverage between Intensity Non-Overlap and TopN Exclusion is around 13%, but in the 6-4 case it is only approximately 10%. We can also see that in the 6-4 exper-

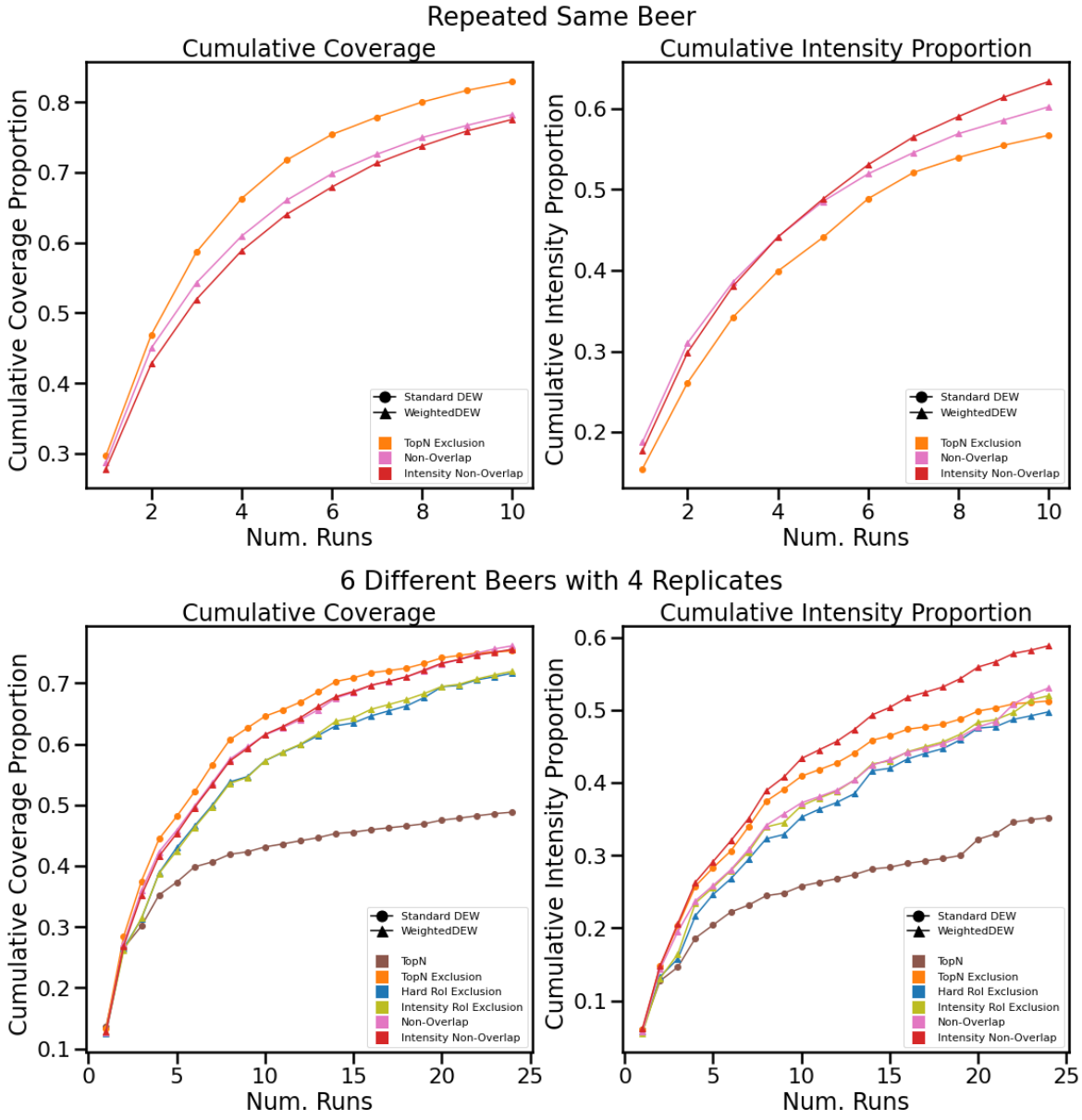

Figure 5.4: Lab experiments evaluated using the permissive parameter set. Top: same beer repeated ten times. Bottom: six different beers each repeated four times in round-robin order.

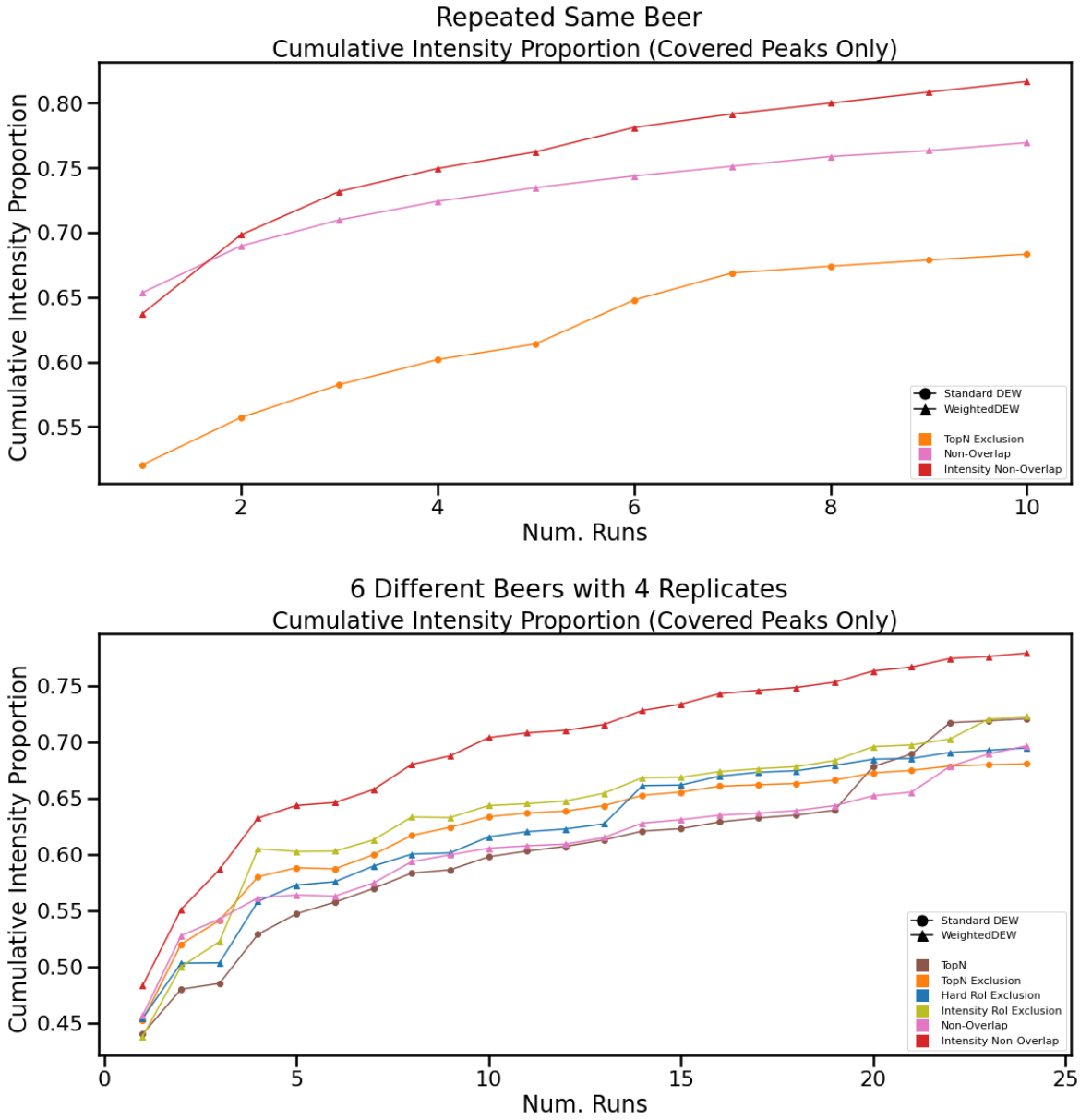

Figure 5.5: Lab experiment evaluated using the permissive parameter set, but showing the intensity proportion calculated by only including those peaks we have covered. Top: same beer repeated ten times. Bottom: six different beers each repeated four times in round-robin order.

iment TopN scores better than anything other than the intensity methods on this alternative intensity coverage measure. This is because this measure simply ignores those peaks which TopN did not cover, which is more than half of the total number. Consequently, the fact that TopN has neglected those peaks to repeatedly target those it did cover benefits TopN when scoring it in this way. As previous work has shown [23] having a lot of MS2 scans targeted at a single peak increases the chance that one will fall at the optimal time for targeting — whether these MS2 scans target similar locations within-run or between-runs.

However, even though the relative coverage of Intensity Non-Overlap compared to TopN Exclusion improves from the same beer experiment to the 6-4 beers, the gap in quality of the spectra they do have narrows from 13% to 10%. That is, the spectra we do have with Intensity Non-Overlap are significantly higher quality compared to TopN Exclusion, and this difference is *larger* in the case where the coverage is worse. This most likely indicates that when TopN Exclusion broadens it coverage by targeting a "weak" peak not included in the restrictive parameter set, the non-overlap methods are instead increasing intensity coverage by repeatedly targeting "strong" peaks. Intensity Non-Overlap was designed to do this to some extent, but it seems to be the case for Non-Overlap as well.

This decrease in coverage and increase in per-spectra coverage for the non-overlap methods could be explained by a tendency to revisit regions of the space compared to TopN Exclusion. By targeting already explored "strong" peaks instead of targeting new "weak" peaks or stopping the duty cycle early and scheduling another MS1 scan, we increase the chance that get a higher intensity coverage on those "strong" peaks or obtain similar peaks across multiple sample types. If relatively "strong" peaks are the only things included in the evaluation (as with the restrictive parameter set) then naturally this confers an advantage.

To substantiate this hypothesis that the new methods revisit potential peaks more often, Tables 5.1, 5.2 and 5.3 show counts of MS2 scans per controller [2]. As the number of runs increases, the number of valid targets for MS2 scans decreases and so MS1 scans are scheduled instead of MS2 scans. In the simulated experiments we can see that over the course of the experiment TopN Exclusion eventually almost entirely stops scheduling new MS2 scans, while Non-Overlap decreases significantly and Intensity Non-Overlap remains nearly constant.

The likely implication of this is that TopN Exclusion is running out of targets because it does not revisit them, whereas Non-Overlap and Intensity Non-Overlap do and thus continue to have targets to schedule. A further piece of evidence is shown in Figures 5.6 and 5.7 which show counts of the number of runs each peak was covered in for both experiments with both peak-picking parameter sets. In each case it can be seen that coverage is distributed more

[2]The simulated experiments use the WeightedDEW variant of the non-overlap controllers to allow for direct comparisons with the lab experiments

Scan Counts — Same Beer						
	TopN Exclusion		**Non-Overlap**		**Intensity Non-Overlap**	
Run Index	MS1s	MS2s	MS1s	MS2s	MS1s	MS2s
1	377	6293	328	6433	328	6433
2	1277	3556	342	6418	329	6485
3	1568	2590	352	6354	335	6384
4	1681	2186	368	6205	346	6320
5	1759	1839	358	6368	354	6289
6	1820	1589	378	6256	352	6331
7	1892	1388	393	6233	360	6306
8	1984	1126	399	6158	360	6299
9	2084	879	413	6103	368	6254
10	2189	619	412	6139	361	6269
11	2250	467	415	6068	363	6268
12	2313	308	483	5845	359	6229
13	2349	218	463	5809	368	6159
14	2364	198	469	5922	365	6226
15	2384	147	451	5951	363	6188
16	2388	123	513	5717	363	6186
17	2400	108	452	5918	360	6257
18	2403	95	456	5906	365	6235
19	2404	81	530	5705	353	6255
20	2408	79	546	5513	360	6173

Table 5.1: Number of scans per run during the same beer simulated experiment, separated by the fragmentation strategy used.

Scan Counts — Repeated Different Beers						
	TopN Exclusion		Non-Overlap		Intensity Non-Overlap	
Run Index	MS1s	MS2s	MS1s	MS2s	MS1s	MS2s
1	377	6293	328	6433	328	6433
2	980	4470	329	6480	328	6489
3	1384	3193	328	6467	328	6478
4	1360	3256	334	6464	329	6455
5	1594	2397	332	6375	328	6420
6	1686	2227	344	6420	331	6469
7	1653	2244	368	6348	336	6409
8	1588	2464	355	6362	338	6419
9	1905	1447	331	6396	328	6437
10	1687	2021	344	6326	330	6399
11	1891	1431	352	6244	331	6388
12	2155	806	379	6263	334	6423
13	1926	1310	375	6223	337	6418
14	1793	1789	404	6177	336	6409
15	2231	518	340	6344	329	6406
16	1919	1287	380	6053	331	6276
17	2147	752	380	6175	331	6365
18	2360	221	404	6158	340	6395
19	2151	686	401	6151	338	6336
20	1966	1306	436	6039	337	6345
21	2358	177	340	6285	331	6335
22	2127	748	390	6128	332	6262
23	2299	333	401	6084	330	6314
24	2402	104	419	6113	337	6391

Table 5.2: Number of scans per run during the repeated different beers simulated experiment, separated by the fragmentation strategy used.

Scan Counts — Same Beer						
	TopN Exclusion		Non-Overlap		Intensity Non-Overlap	
Run Index	MS1s	MS2s	MS1s	MS2s	MS1s	MS2s
1	348	6330	299	5941	298	5922
2	1043	4124	297	5863	299	5923
3	1322	3199	301	5848	298	5918
4	1450	2823	296	5866	308	5857
5	1521	2559	295	5858	297	5891
6	1054	4262	294	5833	298	5901
7	1354	3347	299	5812	307	5821
8	1627	2277	333	5641	296	5861
9	1626	2253	296	5864	294	5839
10	1659	2198	328	5692	293	5808

Scan Counts — Repeated Different Beers						
	TopN Exclusion		Non-Overlap		Intensity Non-Overlap	
Run Index	MS1s	MS2s	MS1s	MS2s	MS1s	MS2s
1	337	6350	298	5909	293	5819
2	757	5038	298	5914	298	5903
3	1221	3568	298	5918	298	5915
4	1281	3374	299	5923	300	5959
5	1456	2804	300	5945	299	5924
6	1496	2504	295	5843	294	5823
7	1447	2744	296	5881	295	5850
8	1387	2993	309	5807	299	5836
9	1539	2508	305	5851	295	5849
10	1504	2609	301	5874	297	5880
11	1599	2318	298	5892	296	5874
12	1702	1856	303	5776	293	5718
13	1586	2307	299	5834	291	5776
14	1476	2702	295	5851	291	5761
15	1607	2272	297	5861	292	5783
16	1539	2488	299	5888	293	5811
17	1645	2160	322	5776	292	5778
18	1901	1245	293	5793	284	5632
19	1623	2169	307	5767	287	5681
20	1518	2553	300	5800	287	5690
21	1645	2144	306	5737	286	5673
22	1588	2326	312	6188	289	5728
23	1702	1960	312	6183	290	5709
24	1988	960	310	6091	279	5534

Table 5.3: Number of scans per run during the lab experiments, separated by the fragmentation strategy used.

heavily towards repeated targeting for Non-Overlap and especially Intensity Non-Overlap. This confirms that these methods are more likely to revisit previously targeted peaks, and together with the MS2 counts suggests that TopN Exclusion neglects doing this in favour of scheduling MS1 scans.

Another possible explanation is that because TopN Exclusion more often schedules MS1 scans instead of MS2, then it is more likely to see opportunities where "borderline" peaks close to the decision threshold are the most appealing option. Also, while the evaluation for an individual controller does not include any peaks that do not have a single eligible precursor above the intensity threshold in one of the fragmentation runs tied to that controller, it is also still possible that the permissive parameter set allows many artifacts or otherwise unreachable peaks. Additional MS1 scans may allow these to be considered for targeting by the fragmentation strategy.

With both of these behaviours in mind, consider Tables 5.4 and 5.5. Table 5.4 divides the intensity range into powers of 10 (with a minimum cutoff at 5000, the minimum MS2 intensity) and then shows counts for the numbers of peaks that fell into each bin based on their maximum observed intensity (not MS2 precursor intensity) during the fragmentation run. It also gives means and medians of the number of LC-MS/MS each peak was covered in, again separated into these bins. Table 5.5 shows the same information, but reports means and medians for the number of times each peak was targeted in total (i.e. counting multiple targetings in the same run, unlike coverage).

Across both the same beer experiment and the 6-4 beer experiment and both restrictive and permissive parameters in both tables, we can see that compared to TopN Exclusion, Non-Overlap tends towards targeting an individual peak more times on average. The same can be said of Intensity Non-Overlap compared to Non-Overlap. This is congruent with the results in Figures 5.6 and 5.7 which give a different view of the data in Table 5.4, and can again potentially be attributed to the slower decay in MS2 scans per run seen in Tables 5.1, 5.2 and 5.3.

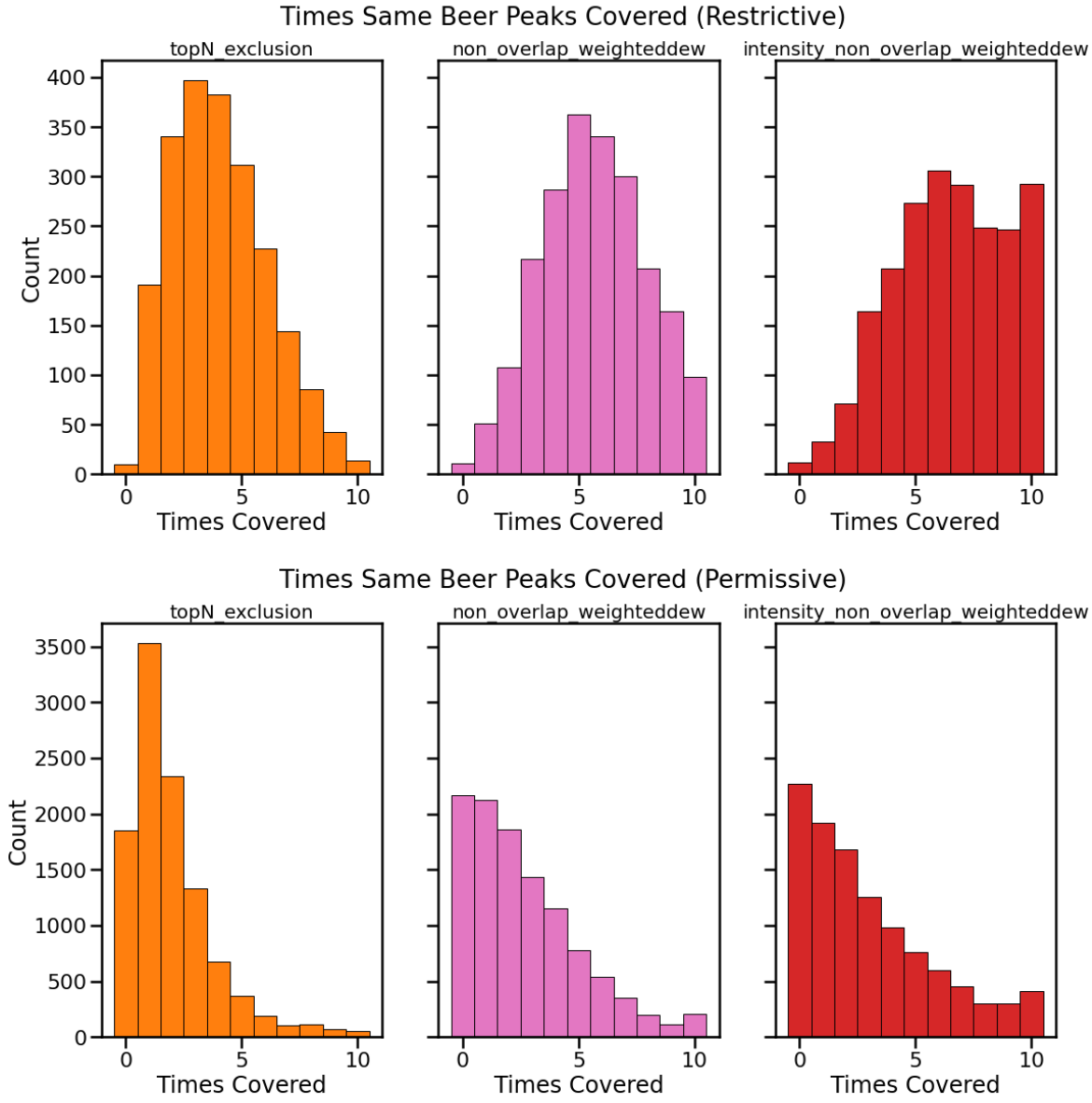

Figure 5.6: Counts of number of different runs each peak was covered in during the lab experiment where the same beer was repeated ten times. Top: restrictive peak-picking. Bottom: permissive peak-picking.

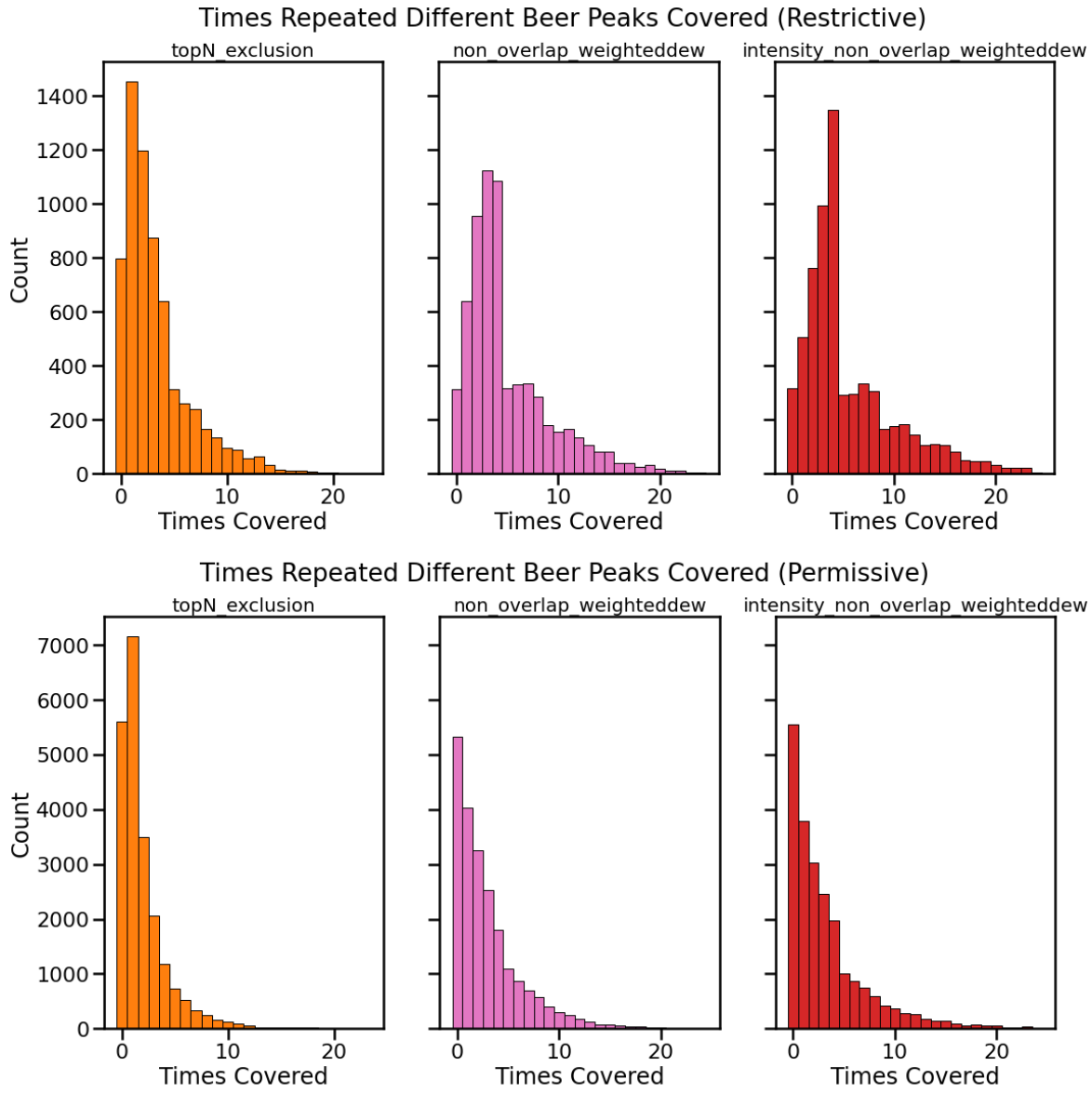

Figure 5.7: Counts of number of different runs each peak was covered in during the lab experiment where four beers were each repeated six times. Top: restrictive peak-picking. Bottom: permissive peak-picking.

Times Same Beer Covered by Intensity (Restrictive)									
	TopN Exclusion			Non-Overlap			Intensity Non-Overlap		
Range	Count	Mean	Median	Count	Mean	Median	Count	Mean	Median
$5000 - 10^4$	133	2.24	2.00	226	4.67	5.00	232	6.06	6.00
$10^4 - 10^5$	1219	3.89	4.00	1230	5.53	5.00	1217	6.15	6.00
$10^5 - 10^6$	633	4.42	4.00	551	5.97	6.00	557	6.87	7.00
$10^6 - 10^7$	136	5.57	6.00	120	7.12	7.00	121	8.10	8.00
$\geq 10^7$	22	7.95	8.00	16	9.69	10.00	16	9.88	10.00
Times Repeated Different Beers Covered by Intensity (Restrictive)									
	TopN Exclusion			Non-Overlap			Intensity Non-Overlap		
Range	Count	Mean	Median	Count	Mean	Median	Count	Mean	Median
$5000 - 10^4$	1087	0.91	1.00	1039	2.68	2.00	1118	3.69	3.00
$10^4 - 10^5$	3421	2.74	2.00	3338	4.23	3.00	3277	4.75	4.00
$10^5 - 10^6$	1687	4.88	4.00	1779	6.31	5.00	1777	6.93	6.00
$10^6 - 10^7$	273	8.27	8.00	309	10.94	11.00	297	11.89	12.00
$\geq 10^7$	44	12.89	13.00	47	15.96	17.00	43	17.51	19.00
Times Same Beer Covered by Intensity (Permissive)									
	TopN Exclusion			Non-Overlap			Intensity Non-Overlap		
Range	Count	Mean	Median	Count	Mean	Median	Count	Mean	Median
$5000 - 10^4$	1960	0.76	0.00	2104	2.08	1.00	2130	2.32	1.00
$10^4 - 10^5$	7382	2.03	2.00	7440	2.70	2.00	7382	2.98	2.00
$10^5 - 10^6$	1394	2.41	2.00	1229	3.07	3.00	1255	3.83	3.00
$10^6 - 10^7$	177	3.53	3.00	148	4.56	4.00	154	5.89	6.00
$\geq 10^7$	26	7.00	7.00	18	9.00	10.00	18	9.67	10.00
Times Repeated Different Beers Covered by Intensity (Permissive)									
	TopN Exclusion			Non-Overlap			Intensity Non-Overlap		
Range	Count	Mean	Median	Count	Mean	Median	Count	Mean	Median
$5000 - 10^4$	4989	0.55	0.00	4910	1.19	0.00	4509	1.69	0.00
$10^4 - 10^5$	14179	1.89	1.00	13863	2.77	2.00	14014	3.04	2.00
$10^5 - 10^6$	2980	3.73	3.00	3315	5.11	4.00	3565	5.37	4.00
$10^6 - 10^7$	329	5.20	5.00	383	6.73	6.00	387	8.24	8.00
$\geq 10^7$	49	10.55	11.00	55	12.55	12.00	51	14.76	15.00

Table 5.4: Table showing relationship between peak intensity and the average number of runs a peak was covered in. For each lab experiment and peak-picking parameter combination, peaks are binned by their maximum observed intensity (not target precursor intensity). Each bin lists the count of its peaks, and the mean and median numbers of LC-MS/MS runs in which any peak in its range was covered.

Times Same Beer Targeted by Intensity (Restrictive)									
	TopN Exclusion			**Non-Overlap**			**Intensity Non-Overlap**		
Range	Count	Mean	Median	Count	Mean	Median	Count	Mean	Median
$5000 - 10^4$	133	5.26	5.00	226	18.06	10.00	232	19.89	12.00
$10^4 - 10^5$	1219	8.92	7.00	1230	16.47	10.00	1217	17.55	10.00
$10^5 - 10^6$	633	9.35	7.00	551	15.15	11.00	557	17.58	12.00
$10^6 - 10^7$	136	11.05	9.00	120	17.92	13.50	121	19.88	16.00
$\geq 10^7$	22	37.64	24.00	16	71.94	38.00	16	83.69	39.50
Times Repeated Different Beers Targeted by Intensity (Restrictive)									
	TopN Exclusion			**Non-Overlap**			**Intensity Non-Overlap**		
Range	Count	Mean	Median	Count	Mean	Median	Count	Mean	Median
$5000 - 10^4$	1087	2.07	1.00	1039	13.03	6.00	1118	13.99	7.00
$10^4 - 10^5$	3421	6.29	4.00	3338	16.05	8.00	3277	16.66	8.00
$10^5 - 10^6$	1687	11.10	8.00	1779	18.54	12.00	1777	20.21	13.00
$10^6 - 10^7$	273	18.35	15.00	309	29.75	21.00	297	32.21	23.00
$\geq 10^7$	44	44.75	32.50	47	68.94	43.00	43	95.40	56.00
Times Same Beer Targeted by Intensity (Permissive)									
	TopN Exclusion			**Non-Overlap**			**Intensity Non-Overlap**		
Range	Count	Mean	Median	Count	Mean	Median	Count	Mean	Median
$5000 - 10^4$	1960	2.12	1.00	2104	6.55	2.00	2130	8.34	2.00
$10^4 - 10^5$	7382	4.56	2.00	7440	7.06	3.00	7382	7.86	3.00
$10^5 - 10^6$	1394	5.62	2.00	1229	8.52	3.00	1255	9.94	4.00
$10^6 - 10^7$	177	6.53	4.00	148	9.86	6.00	154	12.45	8.00
$\geq 10^7$	26	28.77	14.50	18	60.83	32.00	18	70.72	32.50
Times Repeated Different Beers Targeted by Intensity (Permissive)									
	TopN Exclusion			**Non-Overlap**			**Intensity Non-Overlap**		
Range	Count	Mean	Median	Count	Mean	Median	Count	Mean	Median
$5000 - 10^4$	4989	1.28	1.00	4910	4.62	1.00	4509	5.99	1.00
$10^4 - 10^5$	14179	3.96	2.00	13863	7.48	3.00	14014	8.71	3.00
$10^5 - 10^6$	2980	6.92	4.00	3315	11.24	6.00	3565	12.23	6.00
$10^6 - 10^7$	329	8.80	6.00	383	13.54	9.00	387	16.32	10.00
$\geq 10^7$	49	33.06	22.00	55	47.71	25.00	51	70.06	31.00

Table 5.5: Table showing relationship between peak intensity and the average number of runs a peak was covered in. For each lab experiment and peak-picking parameter combination, peaks are binned by their maximum observed intensity (not target precursor intensity). Each bin lists the count of its peaks, and the mean and median numbers of LC-MS/MS runs in which any peak in its range was targeted (including multiple times in the same run).

Additionally, in the same beer case, it seems that the count of peaks in the lowest range, $5000 - 10^4$, is lower for TopN Exclusion, but the counts in the $10^5 - 10^6$ range and above are higher, which suggests these peaks have been observed at higher intensities due to the increased number of MS1 scans. However, the opposite behaviour is seen in the 6-4 beers case so this may have just been an experimental artifact or it may still hold true but cannot be observed here due to the increased complexity of this experiment. Nonetheless this may explain the anomaly observable in the $5000 - 10^4$ row of the permissive same beer segment of Table 5.4 where the median number of times each peak in that range was covered is 0.

While we would ordinarily expect our new methods to cover a higher median number of peaks, the fact that TopN Exclusion has a zero in this row in contrast to the new methods raises the question of how it can attain higher coverage with permissive parameters. However, if some of the peaks that were covered were moved to a higher intensity bin and the coverage was a result of that, then this would make sense. There is no zero median in the same row of Table 5.5 but this may indicate, for example, that some identified peaks were targeted below the minimum fragmentation intensity by a coincidence of the isolation window covering them.

Finally, note that we can see from these tables that there is an extreme bias towards targeting peaks which appear at higher intensities. This is to be expected as all three methods build on TopN's assumption that high intensity implies an interesting target. However, collecting additional fragmentation spectra would be more useful for low-intensity peaks, so future method development may wish to account for this.

Based on these results we might tentatively conclude that the new methods are much more thorough in exploring relatively "strong" peaks compared to TopN Exclusion, and thus will get higher intensity coverage on these "strong" peaks in particular, as well as higher intensity coverage in general. In multi-sample cases (rather than just multi-run cases with the same sample type) revisiting may also confer an advantage in coverage. However, in cases where we must very rapidly obtain sample coverage on a very dense sample type without caring so much about the intensity coverage, TopN Exclusion may be preferable.

Trying to find the right balance of broadening coverage vs revisiting peaks in hopes of increasing coverage/intensity coverage may guide the development of future controllers. However, while this evidence may give us some indication of the final behaviour of our controllers, it is insufficient by itself. A real example of a sample type dense with interesting peaks likely has a very different distribution of intensities. Therefore in this case we might also expect Intensity Non-Overlap, etc to revisit the same targets less frequently, also broadening their coverage. Further work is needed to determine the specifics of the behaviour in this case.

5.4 Alternative Beer/Urine Data

Having drawn all results to this point from the same pool of data, we will now investigate how performance generalises on different data. While we cannot feasibly run another large-scale lab experiment, we can re-simulate any data we have fullscan .mzMLs for. For this we use a previously published data set [148], which we used to test our methods initially, prior to having any lab results to generate re-simulated data from. Although this dataset also uses beers, they are a different set from the set introduced in Section 3.4 which we use for most of our experiments. They were also run in a significantly different experimental context, with a different mass spectrometer from the main TopNEXt experiments (Q-Exactive) and with different instrument settings — the details can be found in the original publication [148]. We again fixed the scan lengths of our simulations to the average scan lengths of the lab TopN runs included with this dataset, in this case 0.28 (MS1) and 0.13 (MS2).

Figures 5.8 and 5.9 show results on these beers for both the same beers and 6-4 beers cases we have already explored. Figure 5.10 leverages the fact that this dataset contains 19 beers to perform a new kind of experiment where each beer is only run once. The equivalent with only six beers would be to read the 6-4 plot up to the sixth run. The top plots in Figures 5.8, 5.9 and 5.10 show the results evaluated with the restrictive peak-picking parameters; the bottom plots show them with the permissive parameters.

The same main conclusions can be observed once again. TopN only gains performance while new sample types continue to be introduced, and overall performs significantly worse than multi-sample methods even prior to this point. Intensity methods easily perform best in intensity coverage, with Intensity Non-Overlap clearly dominant in all six cases. One notable difference is that although we observed in Section 5.3 that the permissive parameter set increased the coverage of the TopN Exclusion variants relative to the new methods, the same pattern is not as pronounced here. In fact, it seems to only be clearly exhibited in the same beer case, but even in this case the Non-Overlap methods are close in performance, and in the other cases with multiple beer sample types they are significantly stronger. It would therefore seem that this effect generated by the permissive parameters and the overall performance of the controllers are significantly dependent on the input data, and perhaps future work can isolate the reason why, to guide appropriate method use.

Peak-peaking for the same beer case produced 1705 identified peaks with the restrictive parameters with the restrictive parameters, and 6901 with the permissive. For the 6-4 case, 5004 restrictive and 17471 permissive identified aligned peaks were produced. For the 19 different beers, 8474 restrictive and 27730 permissive identified aligned peaks were produced.

However, this dataset contains more than just beers. It also contains a number of human urines run in the same experimental context — now we can investigate the performance of

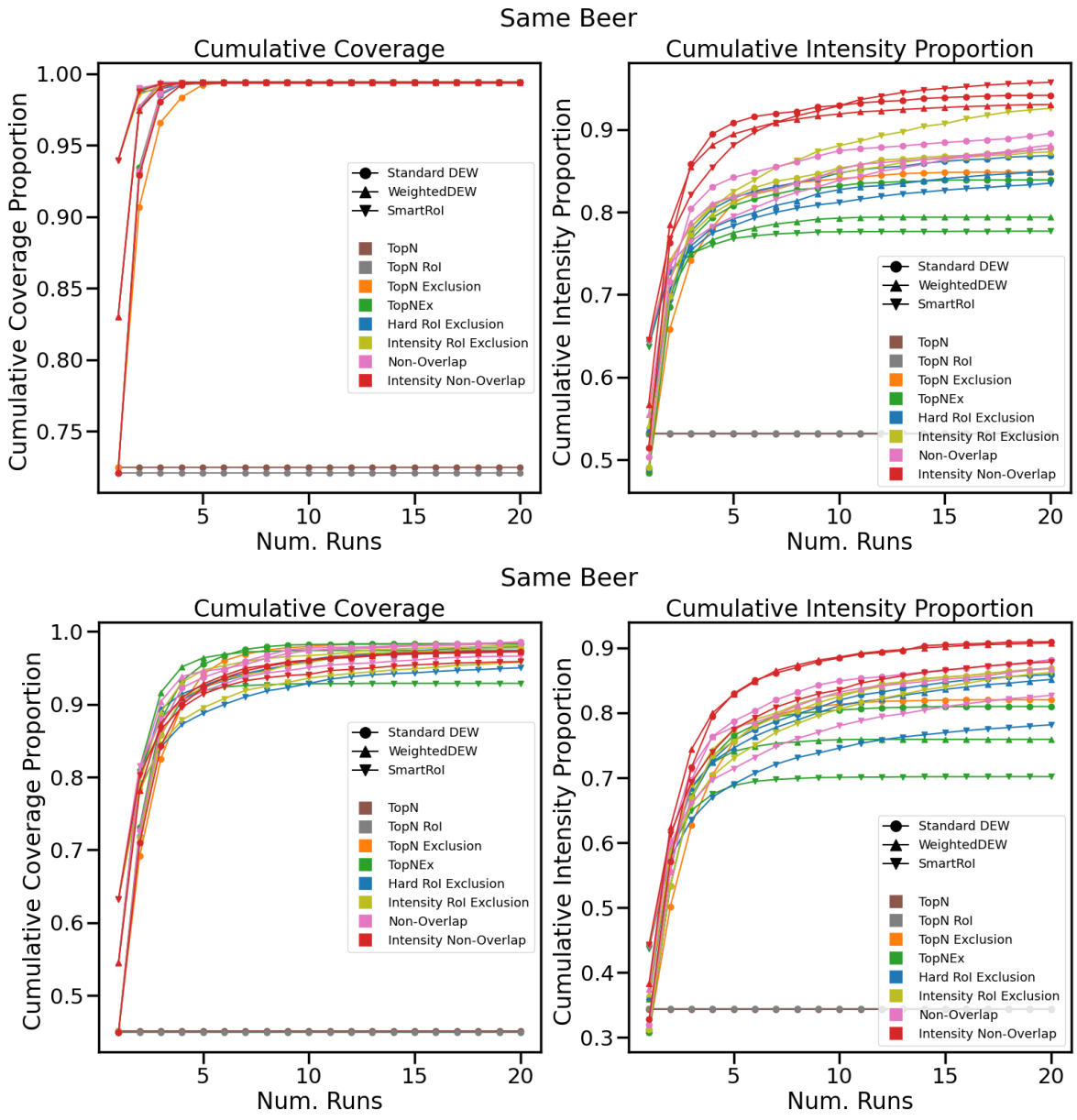

Figure 5.8: Simulated experiment using the alternative beers with the same beer repeated for ten runs. Top: restrictive peak-picking. Bottom: permissive peak-picking.

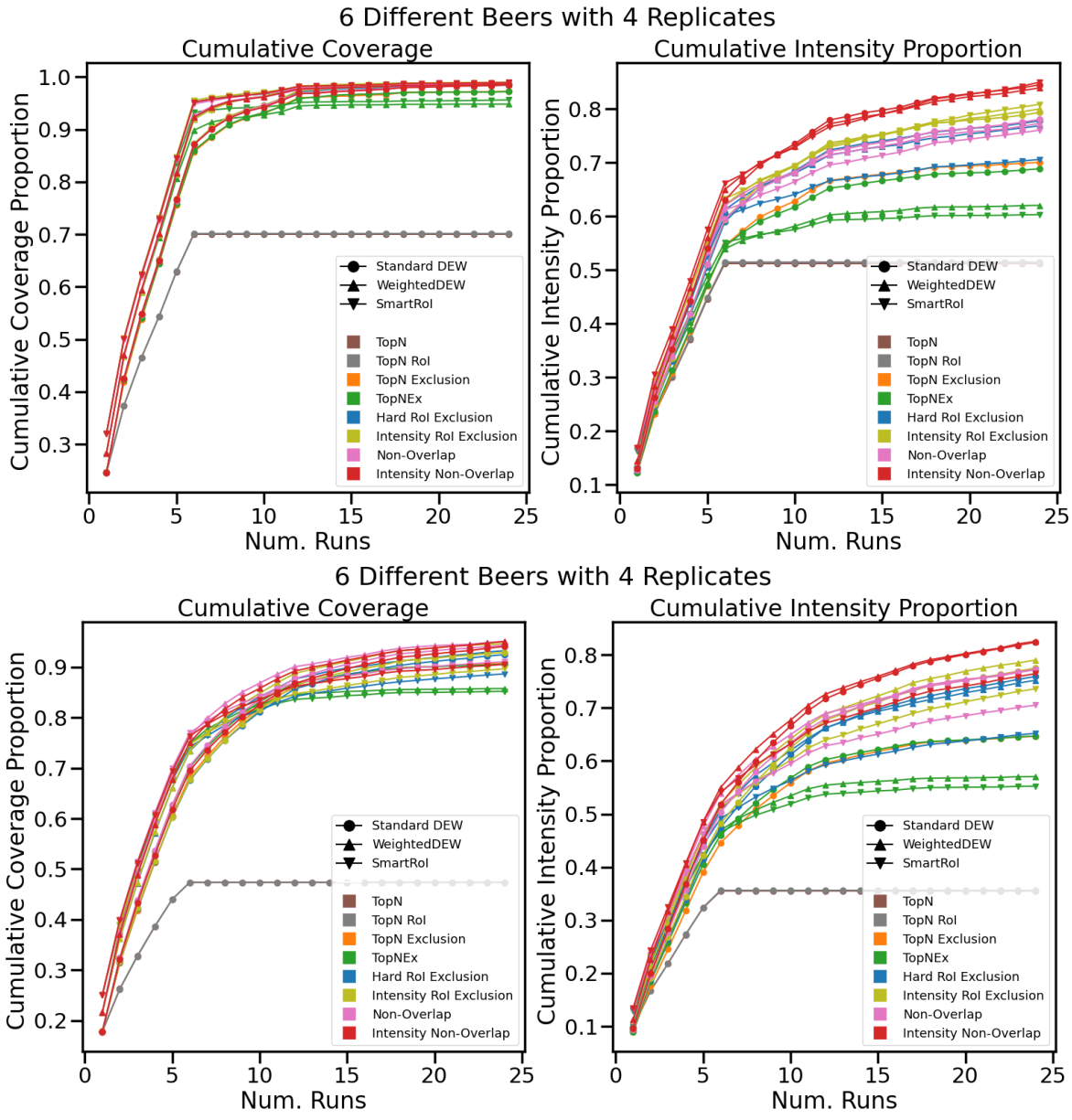

Figure 5.9: Simulated experiment using the alternative beers with six different beers each repeated four times in round-robin order. Top: restrictive peak-picking. Bottom: permissive peak-picking.

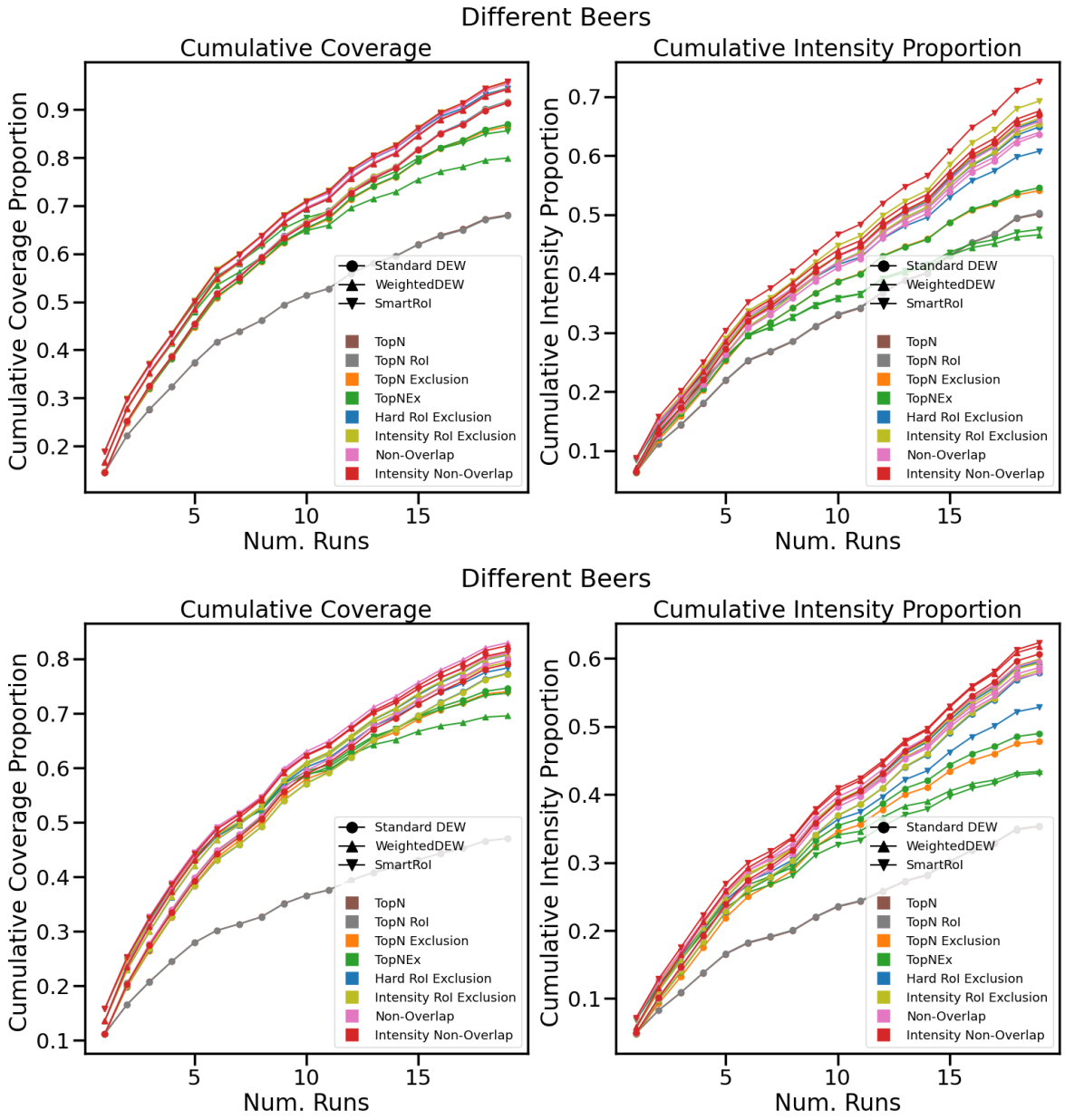

Figure 5.10: Simulated experiment using the alternative beers with nineteen different beers run once each. Top: restrictive peak-picking. Bottom: permissive peak-picking.

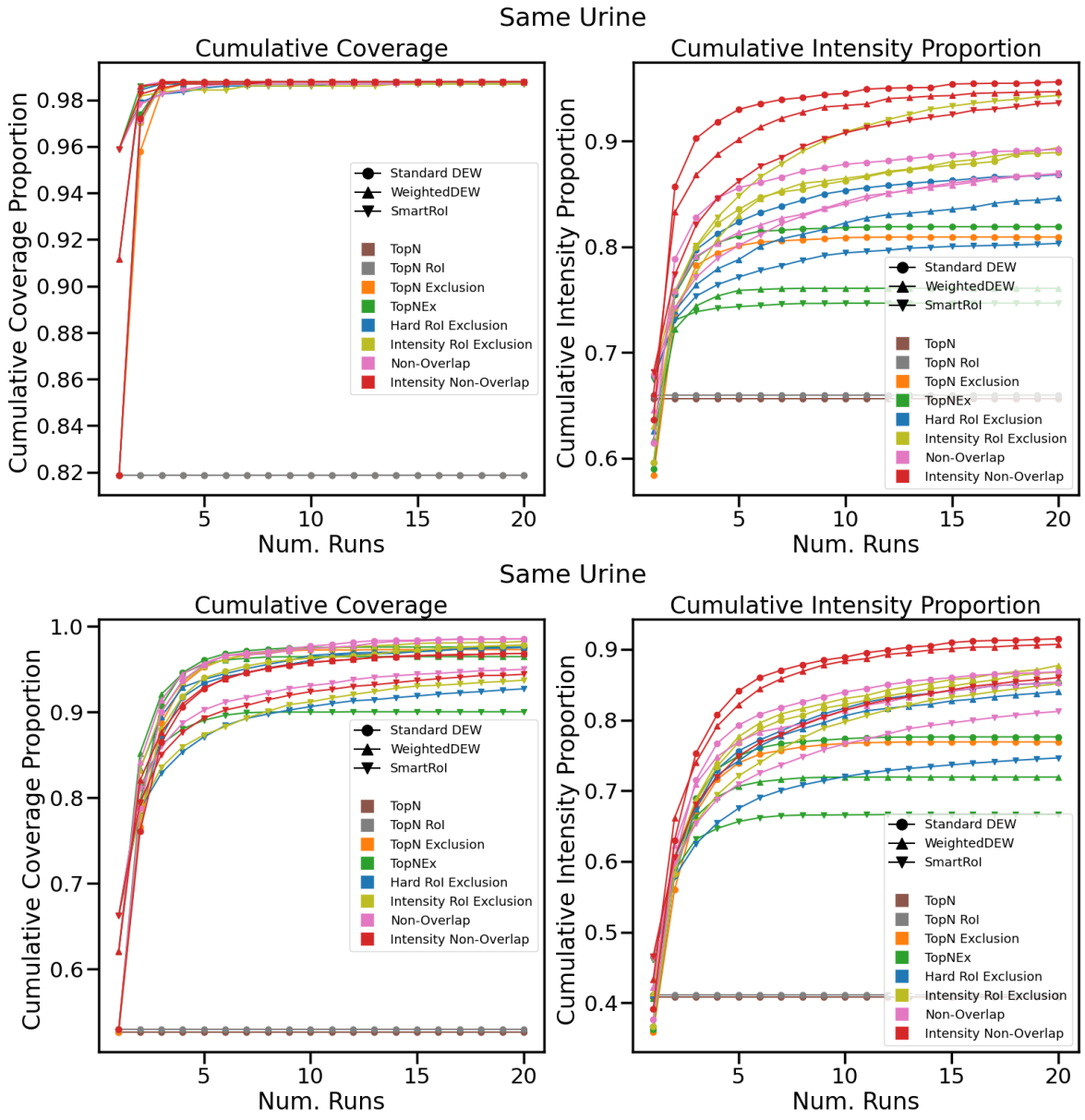

Figure 5.11: Simulated experiment with the same urine repeated for ten runs. Top: restrictive peak-picking. Bottom: permissive peak-picking.

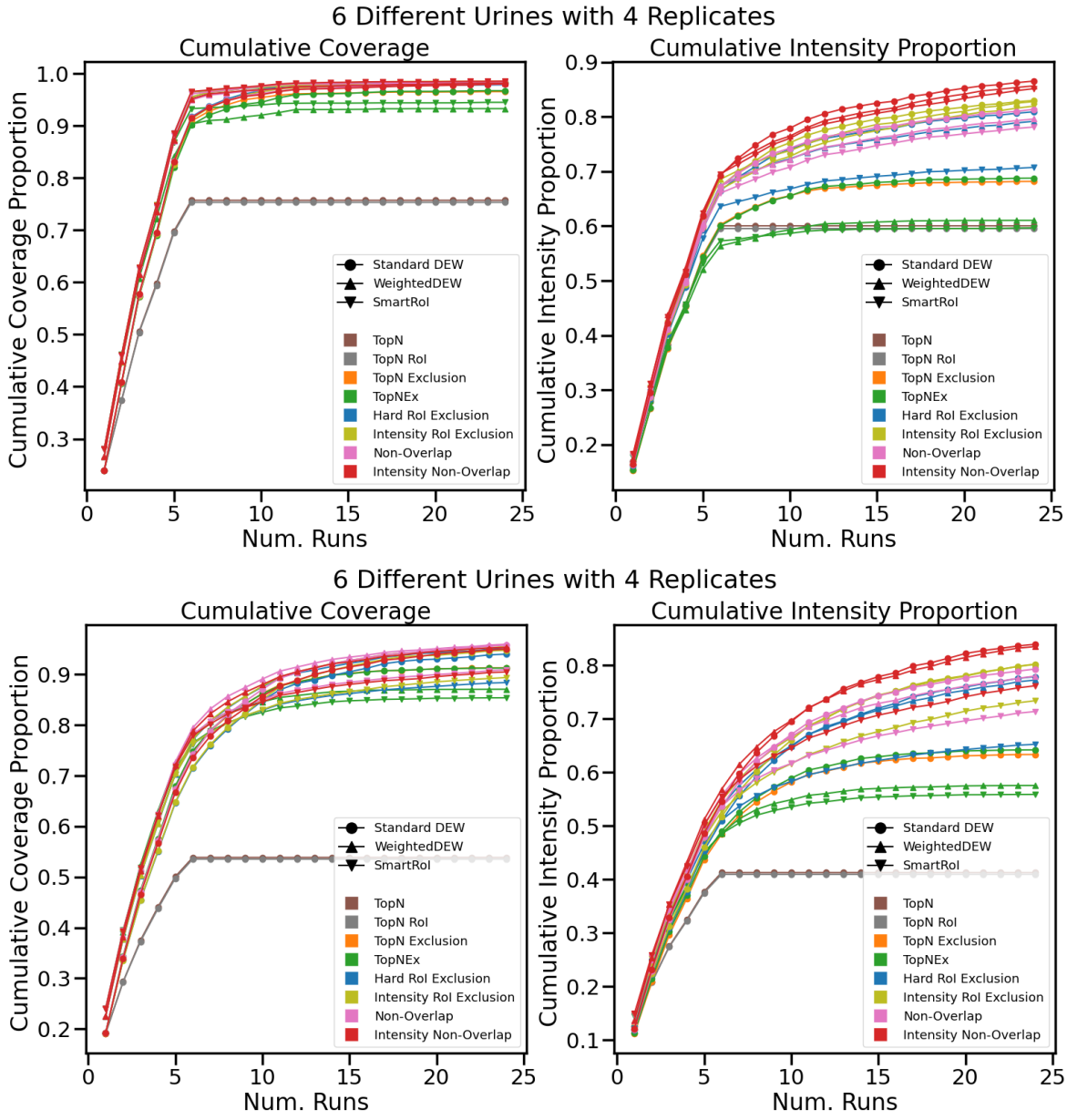

Figure 5.12: Simulated experiment with six different urines each repeated four times. Top: restrictive peak-picking. Bottom: permissive peak-picking.

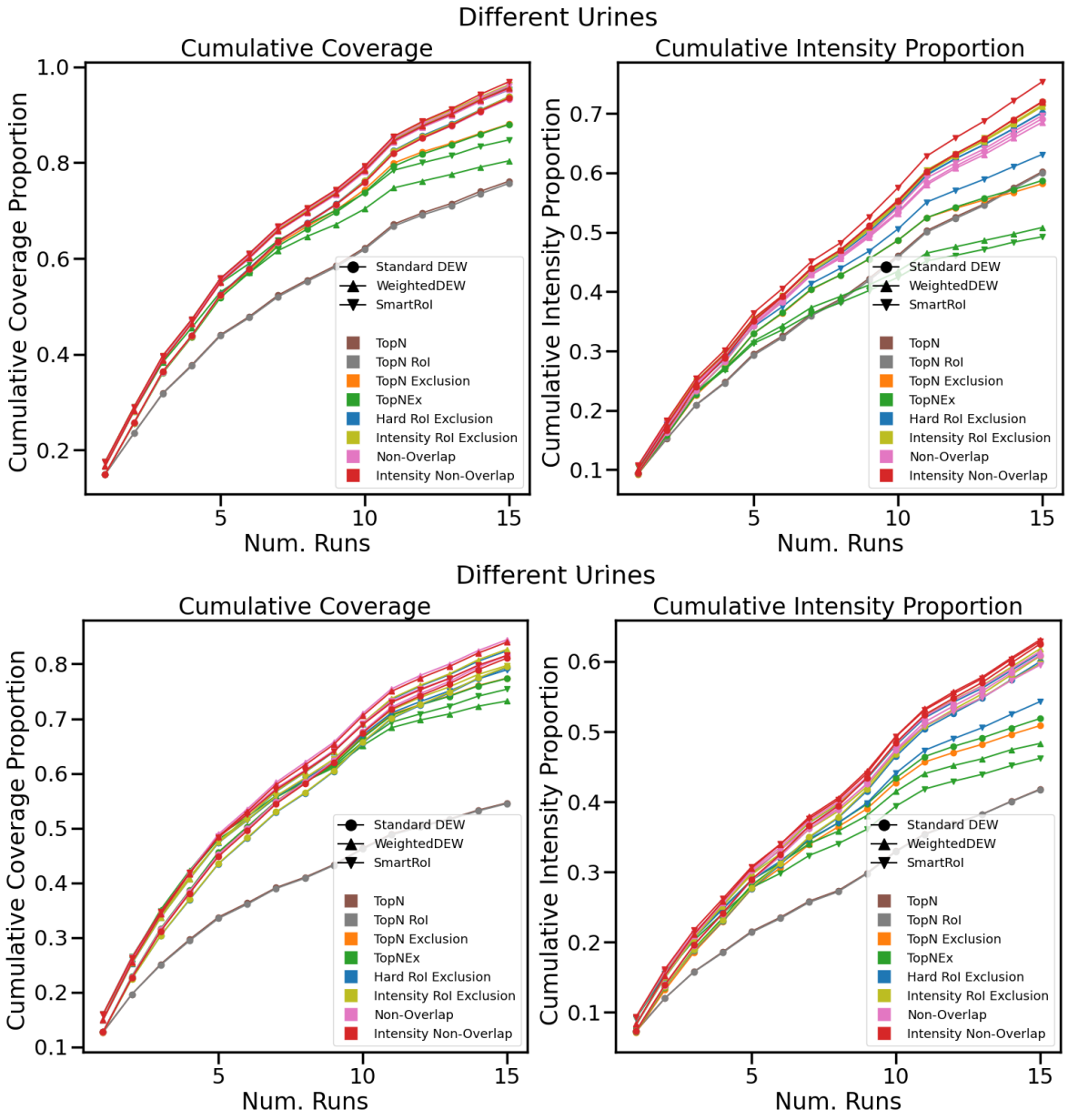

Figure 5.13: Simulated experiment with fifteen different urines run once each. Top: restrictive peak-picking. Bottom: permissive peak-picking.

our methods on a different kind of metabolic sample. Figures 5.11, 5.12 and 5.13 show the same experiments, but with urines instead of beers, and using 15 urines in the different urine case compared to 19 beers (because the dataset only includes 15). Again, the top plots in Figures 5.11, 5.12 and 5.13 show the results evaluated with the restrictive peak-picking parameters; the bottom plots show them with the permissive parameters. Despite using a different type of sample these results have a very similar profile to the beer results we have just seen, and the same conclusions seem to apply, though notably the best DEW variant seems to vary more strongly between them. Overall this suggests that our conclusions about TopNEXt should generalise to other kinds of data and experimental setups, and that the advantages of Intensity Non-Overlap are consistent.

Peak-peaking for the same urine case produced 1142 identified peaks with the restrictive parameters, and 5531 with the permissive. For the 6-4 case, 3903 restrictive and 15264 permissive identified aligned peaks were produced. For the 15 different urines, 7389 restrictive and 26682 permissive identified aligned peaks were produced.

5.5 Timings

Simulated results are easy to produce in large quantities, however there may be limitations to what factors they can realistically model and in this work we have used our lab results as a form of external validation against unknown sources of error. For example, the results we have presented did not model retention time drift in the simulated environment nor did we introduce other kinds of noise — when generated from a fullscan, each run of a sample type would appear exactly the same except for interpolating scans that appeared at different times as a result of the fragmentation strategy's choices. This relative predictability makes the results easier to interpret and reproduce exactly. More generally, in a simulated environment certain properties can be turned on or off or adjusted to investigate hypotheses — in this way simulated and lab results naturally complement each other. Our results being consistent in both a pure environment where we specified only relatively limited assumptions, and in a real-world setting where many uncontrollable sources of error are potentially present, suggests there is a deeper structure to the data accurately captured by the simulation and which our methods take advantage of.

However, for direct comparison, it may still be necessary to account for differences between the environments that have a minor net effect. An example of this is the processing times. In the simulated environment, processing time is ignored and the scan length used is instead drawn from a user-specified distribution. Naturally, in the real environment, we cannot just ignore processing overhead and it is added to the MS1 scan on each cycle. As we stated in the Results section of the main manuscript, in simulation our MS1 and MS2 times were fixed

Average Scan Lengths (seconds, 3 decimal places)		
Method	Avg. MS1 Length	Avg. MS2 Length
Fullscan (1)	0.568	N/A
Fullscan (2)	0.549	N/A
Fullscan (3)	0.550	N/A
Fullscan (4)	0.549	N/A
TopN (4)	0.585	0.193
TopN Exclusion (2)	0.628	0.191
Hard RoI Exclusion WeightedDEW (4)	0.959	0.191
Intensity RoI Exclusion WeightedDEW (4)	0.979	0.190
Non-Overlap WeightedDEW (2)	0.990	0.194
Intensity Non-Overlap WeightedDEW (2)	1.086	0.194

Table 5.6: Average length in seconds of each scan extracted from the lab experiment .mzMLs. Parenthetical numbers after method indicate which batch each method was run in. Each batch produced one fullscan for each beer sample type used, and the times reported here average the MS1 scan times for the first six beers for each batch (i.e. the six used in the 6-4 experiment). Non-fullscan methods were averaged over the 24 .mzMLs from the 6-4 experiment.

to be the average times previously measured in [23], 0.59s and 0.19s respectively, because results with consistent times are easier to interpret and reproduce. This eliminates the effect of what processing overhead there would be in a real environment and thus slightly overestimates the performance of our new, more processing-heavy methods. It would in principle be possible to adjust the scan times of each method to account for this, but it is hard to know what number to use exactly given this is hardware and implementation dependent, and a full empirical study on timings to produce representative figures would be quite involved.

The results in Table 5.6 show the average scan times extracted from the .mzMLs produced by our lab experiments. All our new methods show a significant slowdown of around 0.4s on an approximately 0.6s MS1 scan time, and Intensity Non-Overlap by an additional 0.1s, resulting in fewer scans and likely worse overall performance. It is not surprising that Intensity Non-Overlap would cause a slowdown given it is the most processing-heavy method, though a full analysis of this slowdown would require more than comparing these averages. This is because for Intensity Non-Overlap the processing times are dependent on the number of and positions of exclusion boxes and therefore the specific sample type(s) used and the number of runs (some of these effects can be seen in Table 5.7). However, the 0.4s jump starting from Hard RoI Exclusion is much larger, which is quite surprising given Hard RoI Exclusion is functionally very similar to TopN Exclusion other than the inclusion of RoIs. We can eliminate it being a batching issue given that e.g. Intensity Non-Overlap and TopN Exclusion were run on a separate day from Hard RoI Exclusion and TopN. Therefore the

Intensity Non-Overlap WeightedDEW Average Scan Lengths (seconds, 3 decimal places)				
Run Number	Sample Type	Repeat	Avg. MS1 Length	Avg. MS2 Length
1	1	1	0.998	0.197
2	2	1	1.004	0.193
3	3	1	0.996	0.193
4	4	1	0.975	0.193
5	5	1	0.987	0.193
6	6	1	1.03	0.195
7	1	2	1.039	0.194
8	2	2	1.039	0.193
9	3	2	1.042	0.194
10	4	2	1.032	0.193
11	5	2	1.036	0.193
12	6	2	1.101	0.195
13	1	3	1.099	0.194
14	2	3	1.11	0.194
15	3	3	1.097	0.194
16	4	3	1.086	0.193
17	5	3	1.104	0.193
18	6	3	1.194	0.195
19	1	4	1.185	0.194
20	2	4	1.172	0.194
21	3	4	1.188	0.194
22	4	4	1.156	0.193
23	5	4	1.153	0.194
24	6	4	1.281	0.196

Table 5.7: Average length in seconds of each scan in the lab experiments extracted from the .mzMLs for Intensity Non-Overlap WeightedDEW, and with times for each run presented separately.

most probable culprit is an implementation issue in the RoI-building (previously introduced in [23]) or some of the other basic scaffolding for our methods. This will likely be fixed in future versions of ViMMS but it is impossible to backport this fix to the lab results without re-running the experiments, so the results must be interpreted with this in mind. Therefore the lab results most likely slightly underestimate the performance of our new methods.

Taken together, the simulated results slightly *overestimate* the performance of our new methods by not accounting for processing them, and the lab results slightly *underestimate* them due to the likely inefficient implementation on which TopNEXt is based. Despite this the pattern we can observe between these two sets of results has not changed, so the effect size must be quite small. Consider a duty cycle of one MS1 of approximate length 0.59s and 20 MS2s of approximate length 0.19s each for a total of 4.39s. Then an increase of 0.5s, although nearly doubling the MS1 scan time, is only an increase of slightly over 10% to the length of a full duty cycle, which might explain why the negative effect to our new methods isn't more prominent. The additional slowdown to Intensity Non-Overlap in Table 5.7 is at most approximately 0.3s and therefore even more minor, meaning that while more precise setting of the timings in simulation may help explain minor differences, it is not necessary to understand the overall behaviour. Additionally, the value for $N = 20$ was chosen by *simulated* experiments, so our optimisation procedure in Supplementary Section 4.2 was not influenced by a bias to make the duty cycle longer and reduce the effect of processing time.

5.6 Conclusion

In this chapter we have tested various assumptions that the original TopNEXt experiment in Chapter 4 was based on. We have found the optimal choice of fragmentation strategy parameters does not differ significantly between the controllers tested. We have also found that the results were highly reproducible on both a large-scale replication using other beers from the same dataset, and on different beer/urine datasets collected previously in the literature on a different instrument model.

The assumption making the most significant difference appears to be the choice of peak-picking parameters. On the "permissive" parameter set we found that TopNEXt methods did not have as strong an advantage in intensity coverage, and were a net negative in terms of coverage for a single sample type repeated. We found that as the number of LC-MS/MS runs increases, TopN Exclusion schedules less MS2 scans and instead focuses on MS1 scans. This may be the cause of the differences in performance between peak-picking parameter sets: TopN Exclusion may move on to covering "new" peaks faster while the TopNEXt methods are revisiting them, or that the additional MS1 scans allow TopNEXt to target more peaks near the intensity threshold. Future work may isolate the reason why.

However, there are some caveats to this analysis. Firstly, the change in performance by peak-picking was not reproducible on the other beer/urine data. For this data Intensity Non-Overlap did not lose any significant performance on the permissive parameter set. This effect therefore seems to depend on the input data and more research must be done to understand its impact in practice.

Secondly, the "restrictive" parameter set is (currently) the more reflective of real-world use. The higher quality filter is more representative of "real" chromatographic peaks in the data and thus likely also metabolites. The "permissive" parameter set reflects the ethos that if the machine has nothing better to do with its time, it may as well collect things that are highly likely to be (but may not be) noise. As fragmentation strategies become more efficient, it is possible that the "permissive" parameter set will become more representative, but for now, the "restrictive" parameter set is a more realistic performance benchmark.

Overall, we have shown that the results produced in Chapter 4 were not the product of a specific assumption. We can now have greater confidence in the robustness of TopNEXt and its applicability to various scenarios.

Chapter 6

Maximum Bipartite Matching

With TopNEXt we introduced improved DDA methods, but now we will move on to another important category of fragmentation strategy. Pre-scheduled methods (see Section 2.3.3 for a fuller overview) operate on similar principles, and are often considered a subset of DDA. Like DDA they target individual peaks and so their ability to obtain comprehensive coverage is also governed by the same scheduling problem. But there are also some key differences. Being entirely offline means the pre-scheduled method can use a more involved, processing-heavy global algorithm that plans across the whole acquisition. However, it comes with the trade-off that the fragmentation strategy pre-commits to an assumption about what the data will look like, usually based on some representative data. Therefore pre-scheduled methods are less robust to run-to-run variability. Essentially, compared to DDA, pre-scheduled methods can make stronger assumptions in order to leverage more advanced algorithms and improve comprehensiveness at the expense of robustness.

Described like this, it may seem obvious from a Computing Science perspective that the wealth of research on resource allocation problems could be fairly directly applied here. But, perhaps because the highly interdisciplinary nature of this area creates a high barrier to entry, techniques of this nature have not seen much adoption. DsDA [136], a modern pre-scheduled method (whose introductory work also drew the distinction between “true” DDA methods and pre-scheduled methods) relies on simple DDA-like scoring heuristics and by design neglects peaks which are more rarely observed across the aligned data.

To address this gap we build on a recently-published technique to transform this allocation problem into an instance of maximum bipartite matching [23]. This technique transformed a fixed scan schedule and a set of target peaks into a bipartite scan-peak graph, and then was able to compute an allocation which maximised coverage using a standard algorithm implementation. However, this method was not implemented as an acquisition method in itself: it was instead used as a benchmark theoretical ceiling for the performance of DDA methods. We have therefore taken the next logical step, and implemented it as its own

fragmentation strategy, and in this chapter we will experimentally validate it.

Given a "target list" of rectangular windows that specify regions in $(rt, m/z)$ space to acquire, and representative data which is used as a reference for the intensities that will be observed in a given future scan, this method can produce an optimal schedule targeting as many peaks as possible. To produce the target list, we use peak-picking on fullscan data (which also provides reference intensities) but our matching method can be used with any desired means of generating a target list. Additionally, while the matching method may benefit from dynamic instrument control, this is not a requirement for its use (i.e. it can be run offline by processing its output into whichever format a given instrument type accepts).

We have also implemented several new extensions to the bipartite matching technique that are necessary for its usability in practice. Firstly, the original technique was designed for a single LC-MS/MS run, so we have extended it to be capable of computing a global schedule across multiple LC-MS/MS runs (and multiple sample types). Secondly, we use a maximum weighted bipartite matching algorithm to optimise the precursor intensity of each target, where the previous technique did not consider the quality of individual assignments. Thirdly, we have extended it so that leftover scans can be redundantly assigned in order to improve the robustness of the technique. Finally, since this technique might be used to feed an inclusion window workflow as well as being a standalone pre-scheduled method, TopNEXt [27] has been updated to be able to use inclusion windows.

We will demonstrate through experiment that this is a state-of-the-art method for data acquisition, but we will also illustrate some of the trade-offs involved in choosing a completely offline pre-scheduled method over a DDA method or vice versa. Section 6.1 describes the maximum bipartite matching problem, and how LC-MS/MS acquisition control can be mapped to an instance of the maximum bipartite matching problem, including our improvements. Section 6.2 describes parameter settings for the controllers we used in our experiments. Section 6.3 gives our experiments, with Sections 6.3.1 and 6.3.2 describing a best-case scenario for pre-scheduling and Section 6.3.3 describing a harder and more realistic scenario. Section 6.5 gives recorded runtimes of different parts of the matching workflow.

At the time of writing, this work is being finalised for publication. The collection of lab data was again performed by a technologist (Stefan Weidt) but otherwise the work in this chapter is my own.

6.1 Definitions

As we have indicated, to optimally assign MS2 scans to targets by bipartite matching, we model scans and target peaks as a bipartite graph. A bipartite graph is an abstract formalism

allowing us to capture relationships between two disjoint sets of objects. Objects are represented as "vertices", and relationships as an "edge" joining two vertices. A maximum matching is the largest set of pairs of those objects where none of those objects appear in more than one pair. In our scan-peak graph, the presence of an edge represents the ability of an MS2 scan to target a chromatographic peak. Thus solving for the maximum bipartite matching on this graph effectively gives a list of MS2 targets to use for expected maximum performance. The mapping of this problem to a well-understood graph theory problem [149, 150] allows us to solve it efficiently with a standard algorithm (in this case the Hopcroft-Karp algorithm [151]).

In this scan-peak scheme, a bipartite graph can be created from a scan schedule, a target list and a list of representative fullscan .mzML files, one per LC-MS/MS run to plan for. The scan schedule is a user-specified list of RTs and MS-levels for scans to be run. The scan schedule is set in advance by the user rather than being optimised algorithmically because doing so would be a significantly more challenging problem — a change to a given scan's RT would affect the RTs of its successors and therefore their potential targets, causing significant combinatorial blowup in the number of solutions. But while we require that the scan schedule is fixed in advance, there is no requirement that the schedule used is entirely uniform. For example, one could lower the number of MS2s used relative to MS1s as the number of runs increases. However, in this work we have opted for the simplest method of maintaining a TopN-like duty cycle for a fixed N throughout all runs as this makes it easier to fairly evaluate the performance of our methods.

The target list specifies acceptable $(rt, m/z)$ bounds for each target you would like to acquire. In this work it is produced by peak-picking software (due to the ease of this method) but it can be produced by any method of the user's choice, as long as it is provided in the correct format. The MS1 values in the representative data are used to populate the expected MS1 intensities of each target in the new scan schedule — this is necessary to populate the edges of our scan-peak graph. For representative data we use fullscan .mzML files of the same sample type we want to plan on, as fullscan files have the richest intensity information. The maximum matching computed on the graph produced from the scan schedule, target list and representative data can then be used directly to create a pre-scheduled strategy, or converted to inclusion windows (by taking a small window around each planned MS2) and given to a compatible DDA strategy. An overview can be seen in Figure 6.1.

In Section 6.1.1 we describe the background of the scan-peak bipartite matching technique as it was previously used [23]. We contribute the ability to actually execute this schedule in an MS run and several improvements described in the following sections. Section 6.1.2 describes how a scan-peak graph can be constructed for an experiment of multiple MS runs using either a single sample type or multiple sample types. Section 6.1.3 describes how we assign MS2 scans to the highest expected precursor intensity available per peak and therefore

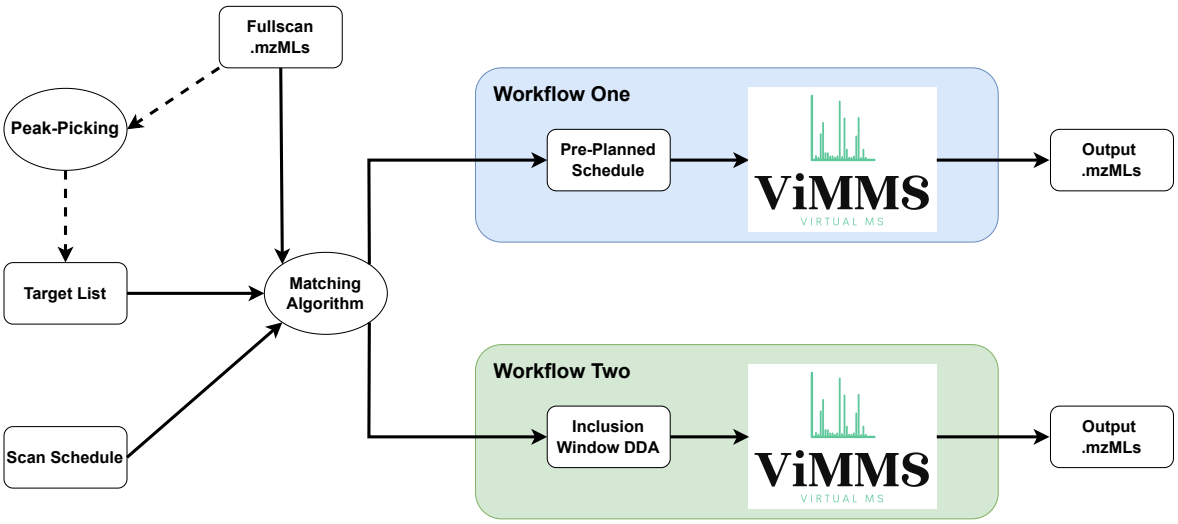

Figure 6.1: Diagram showing the two possible workflows and the flow of data. In either case, representative (fullscan) data, a target list and a scan-schedule are used to create a maximum matching. This maximum matching can then be converted into either a completely pre-planned schedule, or inclusion windows to inform a DDA method. These can then be run through the Virtual Metabolomics Mass Spectrometer (ViMMS) to produce output LC-MS/MS data. The target list can be produced by any means desired — to produce them in this work we use chromatographic peak-picking software on the representative data. The optional nature of this procedure is represented with dashed arrows.

optimise acquisition quality. Section 6.1.4 describes how we assign leftover MS2 scans into a full assignment for redundancy and therefore fault-tolerance. Finally, Section 6.1.5 describes how inclusion windows can be integrated into the TopNEXt framework, and therefore how maximum matching can be used to improve its DDA controllers.

6.1.1 Background

Formally, a graph G has vertices V and edges E. An (undirected) edge is of the form $\{u, v\}$ where $u, v \in V$. Each vertex represents an object of interest, and an (undirected) edge represents a symmetric relationship between them. In our case we want to capture two disjoint sets of objects, MS2 scans and chromatographic peaks, and model the relationship of whether a given scan is able to target a given peak.

As before, an MS2 scan is represented by a retention time at which it occurs, and a chromatographic peak has a bounding interval in $(rt, m/z)$ space i.e. a rectangle. MS2 scans follow an arbitrary schedule given as input, and chromatographic peaks may be provided by e.g. XCMS [74], MZMine [75] or an arbitrary method of the experimenter's choice. We therefore have two disjoint sets of vertices that are subsets of V i.e. $S \subset V$, $P \subset V$ and $S \cap P = \emptyset$ where S is the set of MS2 scans and P is the set of target peaks. An edge is created between a given $s \in S$ and $p \in P$ only if the RT of s intersects the peak interval

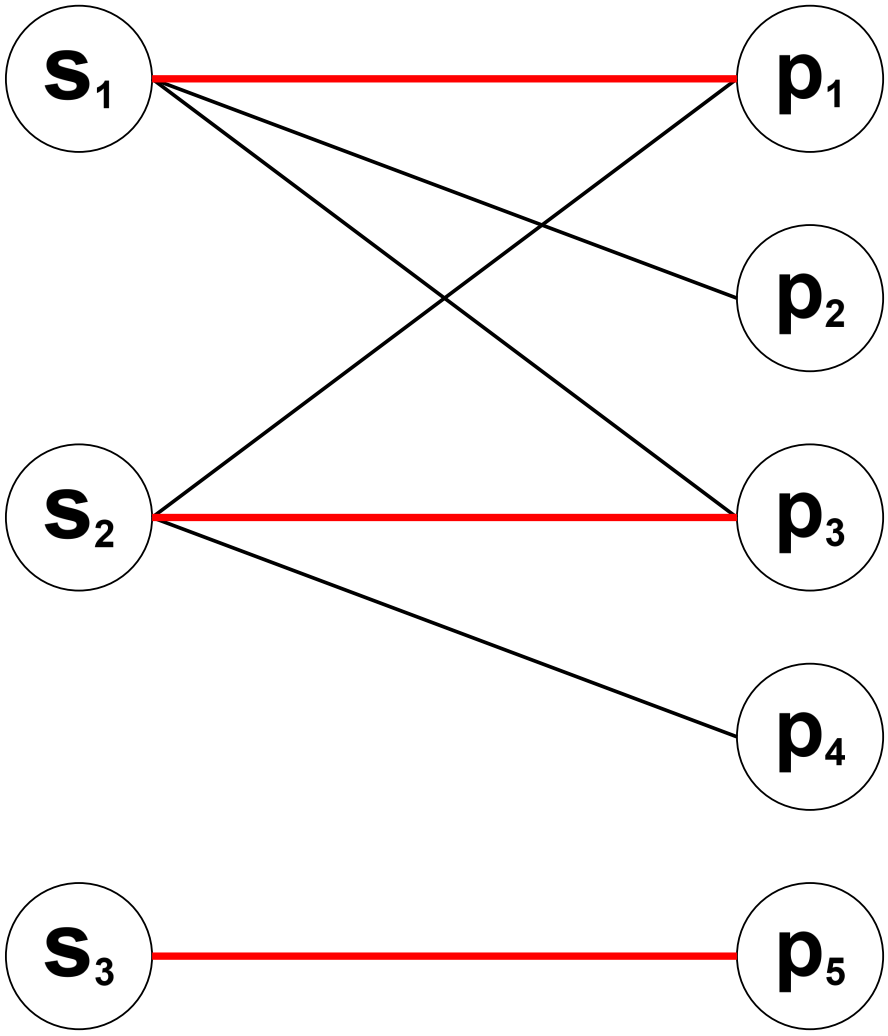

Figure 6.2: A toy example of a maximum bipartite matching between scans and peaks, where an edge indicates that a peak can be targeted by its connected scan. Edges included in the matching are marked in red and are slightly thicker. All the vertices in the left-hand (scan) side of the matching are assigned — it is "one-sided perfect" or "full". An obvious consequence of this is that it must also be a maximum matching.

and there exists a valid precursor (represented as a single point on the precursor MS1 scan) above a minimum intensity threshold inside the peak interval.

Because we have two disjoint sets of vertices S and P, and each edge connects a vertex $s \in S$ to a vertex $p \in P$, G is a bipartite graph. We now want to compute an assignment of scans to peaks such that each scan is assigned to no more than one peak (so that each ion species is isolated individually) and such that these one-to-one assignments cover the maximum number of distinct peaks. A matching on a bipartite graph is a subset of edges such that no two edges in the matching share a vertex i.e. $e_1 = \{s_1, p_1\} \in M,\ e_2 = \{s_2, p_2\} \in M \implies s_1 \neq s_2,\ p_1 \neq p_2$ for a matching $M \subset E$. A graph's maximum matching is a matching on that graph such that no larger matching exists on it. The Hopcroft-Karp algorithm [151] is able to solve the maximum bipartite matching problem in $O(|E| \cdot \sqrt{|V|})$ time, which is

superior to the $O(|V|.|E|)$ Ford-Fulkerson algorithm [23]. The implementation we use is provided by NetworkX [152]. Figure 6.2 illustrates a toy example of the maximum bipartite matching on a graph constructed this way.

6.1.2 Multi-Sample Matching

In Figure 6.2 we illustrated planning for a single LC-MS/MS run, but often we may be interested in planning for a series of multiple aligned LC-MS/MS runs (potentially of multiple sample types) to obtain coverage of them as a set. Were we to create a scan/peak graph for each run, these graphs would likely not be entirely independent. While the scans in each graph would be disjoint from any other graph (they represent different scan events happening across different LC-MS/MS runs) the aligned peaks would not be (if an analyte produces the same peak across two runs we may not want to acquire it twice).

A straightforward, greedy approach is to run the matching, delete any peaks you successfully targeted, and then run a new matching on the remainder. But this technique is not globally optimal. Consider p_3 in Figure 6.3A. Should s_2 be naively targeted at p_1 or p_4 instead, then p_3 cannot be acquired as it disappears in the second run in Figure 6.3B. A greedy method has no foreknowledge that p_3 is not going to be present in run two and thus may miss it. Assuming this disappearance will also occur in the fragmentation runs, then this should be avoidable. An example of a real-world scenario where an event like this might occur could be that the two graphs were generated from a case-control setup where p_3 represents a meaningful biomarker only present in one sample type.

We therefore instead combine the graphs into a single graph: all scans and edges are preserved as-is, and all *unique* peaks (as decided by peak alignment in e.g. XCMS) are also included. The combined graph in Figure 6.3C assigns all six peaks because (having observed the representative data) it has foreknowledge that p_3 must be acquired on the first LC-MS/MS run. Conversely, a greedy approach does not look for cases like this and thus will only obtain the optimal solution by luck some of the time. There are caveats to solving a matching on a combined graph, however. The number of runs must be specified in advance, and there is no priority to acquiring a peak sooner rather than later.

Consider p_5 in Figure 6.3. If s_6 was not able to target p_4 then the only target available to either s_3 and s_6 would have been p_5, and the matching solver would be indifferent to the resultant choice between s_3 and s_6. The solver might choose to target p_5 with s_6, but suppose that in the fragmentation runs the edge $\{s_6, p_5\}$ disappeared while $\{s_3, p_5\}$ remained. s_3 would have already occurred by the time s_6 was observed, so despite s_3 being unassigned there would be no opportunity to target p_5. A greedy approach might try p_5 on the first run, then the second, and so on until it succeeds. If a peak is deferred until a later scan one acquisition opportunity is lost, which may be problematic due to noise.

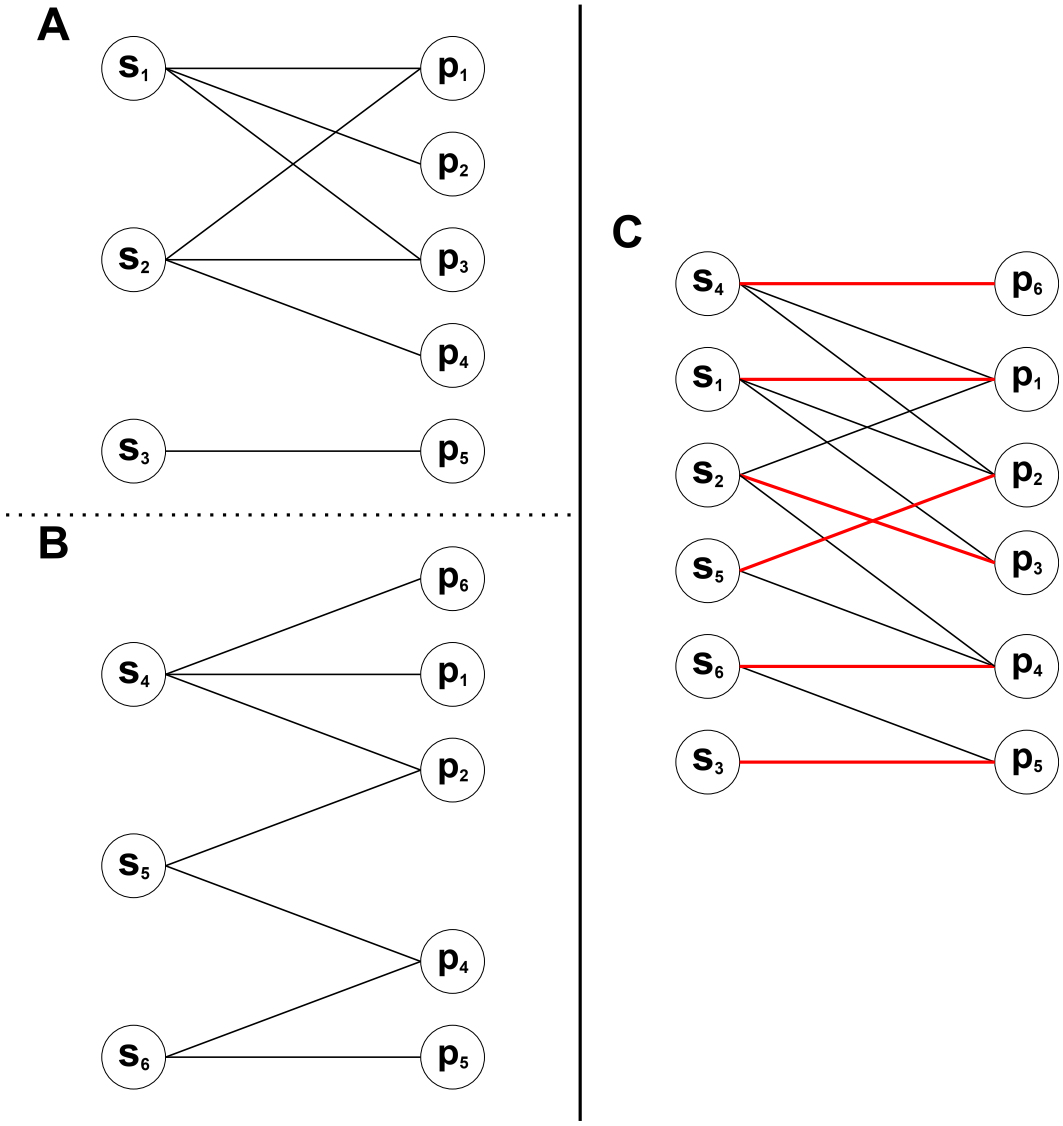

Figure 6.3: **A** and **B**: Individual graphs for a toy example of two runs of a mass spectrometer. **A** is from Figure 6.2. **B** is mostly similar, but p_6 appears in s_4, p_2 appears in s_5 instead of p_1, p_3 does not appear at all and p_4 appears in s_6. **C:** The combined version of the two previous graphs for a multi-sample matching. In order to combine the graphs, the scans are "stacked" and the peaks are "merged". All $3+3=6$ scans appear in the final graph, but each unique peak appears only once. All scans and peaks have been assigned, so this matching is "perfect". Note that in the combined graph scans have been reordered to reduce visual clutter only.

To address these issues, firstly, the size of the matching can be seen prior to running it, allowing the number of LC-MS/MS runs needed to be estimated in advance. Secondly, it is possible to combine the greedy and combined graph approaches by creating a combined graph for batches of runs — we leave the details of this implementation and selection of chunk size to future work. Thirdly, creating a full assignment as in Section 6.1.4 allocates scans redundantly to help avoid this issue.

6.1.3 Intensity Matching

As we have mentioned a number of times, it is generally preferable to target peaks at their apex precursor intensity to obtain the highest-quality fragmentation spectrum. However, a regular maximum bipartite matching will ensure as many peaks as possible have an MS2 scan assigned, but the algorithm is completely indifferent as to which scan is used provided the choice does not block other targets. Thus, to obtain higher-quality fragmentation spectra, we annotate each edge with the precursor intensity and solve a maximum *weighted* bipartite matching.

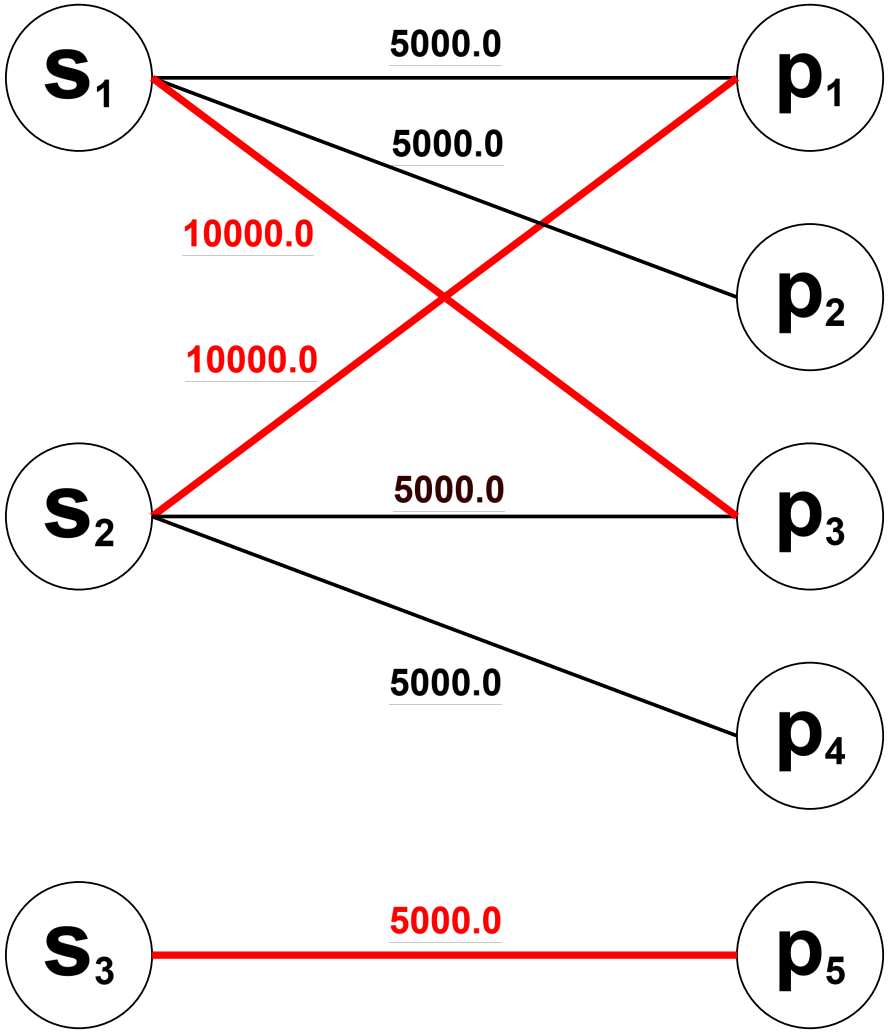

Figure 6.4: A maximum intensity matching on the toy graph from Figure 6.2 after precursor intensities have been annotated on the edges. In the two-step matching, we first perform the maximum coverage matching. Assuming we get the same matching as in Figure 6.2, p_2 and p_4 will not be included in it and thus will be removed to create the auxiliary graph. However, because there is a higher intensity edge from s_1 to p_3 and s_2 to p_1 these will be reassigned and the final matching will have a different assignment of edges to the matching in Figure 6.2. Any edge between a scan-peak pair is maintained.

However, it is common for algorithms and their implementations to assume that a "perfect" or "one-sided perfect" solution is available [153]. A "one-sided perfect" solution is one where all the vertices in one of the partitions are assigned: a "perfect" solution is one where all vertices are assigned. The algorithm implemented by NetworkX [154] requires the solution to be "full" (i.e. one-sided perfect). In practice this means if there is no solution where either all scans or all peaks can be included, the algorithm is not valid and cannot be used (the implementation will terminate without finding a solution if it finds this condition is not met). NetworkX does contain an implementation of a standard algorithm for a maximum weighted matching on general (i.e. not necessarily bipartite) graphs [155, 149] but we found that this algorithm was computationally infeasible for our scan-peak graphs.

Therefore to have a working algorithm with an acceptable runtime, we use a "two-step matching" approach. We firstly find an unweighted maximum matching, which we use to create a one-sided perfect auxiliary graph by deleting any peaks not included in the matching and any edges attached to them. We then secondly solve a weighted matching on the auxiliary graph using the NetworkX implementation of the Jonker-Volgenant algorithm [156]. Like the classic Hungarian algorithm, the Jonker-Volgenant algorithm has a complexity of $O(n^3)$, but is often much faster in practice, so is used for the standard implementation in NetworkX (which we chose for its convenience). Figure 6.4 shows how the targets might be "swapped" from the initial assignment in Figure 6.2. This has the side effect that even if it is otherwise possible to increase the total sum of acquisition intensities by reducing the number of unique peaks targeted, any matching we create will first prioritise the number of unique peaks targeted.

6.1.4 Full Assignment of MS2 Scans

As we use more LC-MS/MS runs (for example, to profile crowded regions) we will inevitably end up in a situation where not every MS2 scan can be given a unique target. For example, Figure 6.5 shows a simple case where there are more scans than peaks. By the definition of a bipartite matching, not all of these scans can be included, but they should still ideally be given a useful target. For example, they may redundantly target peaks in the matching to add robustness or aid identification with extra spectra.

We provide two simple heuristic rules to create a full assignment of scans (though the approach could be easily adapted to use another). The first rule, $nearest$, simply sets any unallocated MS2 scan to have the same target as the nearest (by scan index) allocated MS2 scan (even if the newly allocated MS2 scan did not have an edge to that peak in the matching and thus would not be considered targetable at that point). The second rule, $recursive$, computes a matching, creates an auxiliary graph by removing all peaks not in the matching and

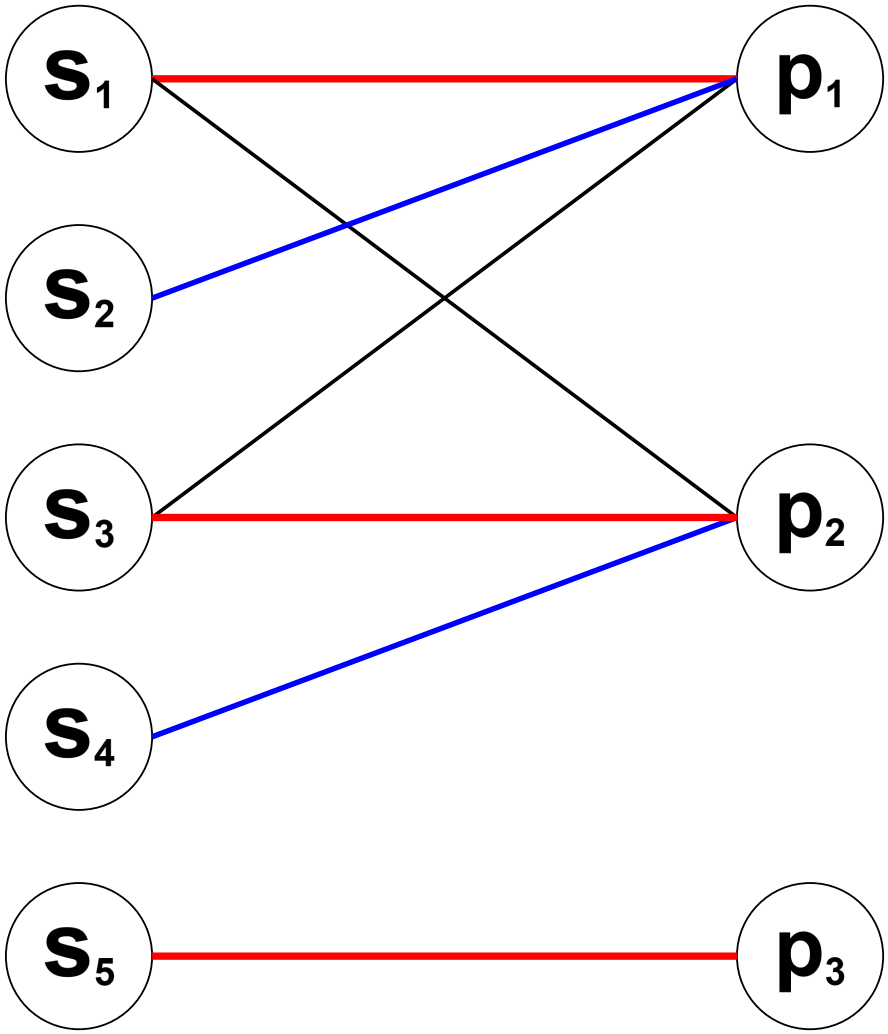

Figure 6.5: A toy example of a bipartite matching being turned into a full assignment. The graph is the same example given in Figure 6.2 but flipped so there are more scans than peaks. The matching, marked in red, is also the same but now not all scans are assigned to it. Therefore, we have marked a possible follow-up assignment in blue with lines which are thinner than the red lines but thicker than the black lines. This assignment could be created by, for example, running a second iteration of the matching with s_1, s_3, s_5 removed.

all scans in the matching, and then repeats solving on auxiliary graphs and removing scans until no scans remain in the auxiliary graph.

$nearest$ is simpler to compute and attempts to redundantly target peaks in a local time frame. $recursive$ is much more expensive to compute, taking up to 30 minutes to compute for the experiments in Section 6.3 and 6.4 (see Section 6.5 for timings). On the other hand, $nearest$ has a sub-second timing. In exchange for its more intensive processing, $recursive$ has a much better spread on its targets — it attempts to target each reachable peak once for each sub-iteration. This should make it more robust especially against forms of heavy noise such as peaks entirely dropping out (and we will show evidence for this in our experiments). While the computational cost had a reasonable upper bound for our data there may be larger

data for which a less intensive scheme like $nearest$ is required.

6.1.5 TopNEXt Inclusion Windows

Inclusion windows can be straightforwardly generated from a completed matching by creating a window of the desired size around each planned target. These can then be given to a standard inclusion workflow if desired. So that this kind of workflow can be used with advanced DDA methods, we have extended the TopNEXt framework from previous chapters. As a reminder, in TopNEXt, a DDA method is expressed as a scoring function of the form in Equation 6.1.

$$score(r, Ex, \lambda_{min}) = I_{ex}(r, Ex) \cdot I_{\lambda}(\lambda_r \geq \lambda_{min}) \cdot \lambda'_r \tag{6.1}$$

When deciding fragmentation targets, a TopNEXt method assigns each active Region of Interest (RoI), r, an initial score λ'_r usually based on current intensity. This score is then subject to multiple exclusion criteria. I_{λ} is an indicator function which ensures the current intensity of the RoI, λ_r, is above a user-defined minimum threshold, λ_{min}. I_{ex} is an indicator function which ensures the RoI is not currently excluded by DEWs or static exclusion windows, which are held in Ex. If either of these conditions are not met, the score, given by $score(r, Ex)$, becomes zero. Individual methods are implemented by defining λ'_r. For example, TopN and TopN Exclusion can be implemented by setting $\lambda'_r = log(\lambda_r)$, shown in Equation 6.2.

$$score(r, Ex, \lambda_{min}) = I_{ex}(r, Ex) \cdot I_{\lambda}(\lambda_r \geq \lambda_{min}) \cdot log(\lambda_r) \tag{6.2}$$

The nominal score for these functions is simply the logarithm of the precursor intensity and the difference between TopN and TopN Exclusion is handled by the exclusion windows stored in Ex.

To add inclusion windows to the TopNEXt framework, we included an additional term to the scoring function, seen in Equation 6.3.

$$score(r, Ex, \lambda_{min}) = I_{ex}(r, Ex) \cdot I_{\lambda}(\lambda_r \geq \lambda_{min}) \cdot (\lambda'_r + I_{in}(r, In) \cdot max_r(\lambda'_r)) \tag{6.3}$$

Similarly to I_{ex}, I_{in} tells us whether or not the RoI falls within any of the inclusion windows in In. If it does then the value of the largest score, $max_r(\lambda'_r)$, is added to the score. This ensures that any valid RoI with the latest precursor falling within an inclusion window will

override any RoIs where this condition is not met. However, prioritisation is retained between those RoIs which either both trigger an inclusion window or both do not. This will also not override exclusion windows and the minimum intensity threshold.

6.2 Parameter Settings

Our experiments in Section 6.3 will compare a number of baseline controllers to our new pre-scheduled controllers and baseline controllers with inclusion windows added. However, we must choose parameters to run these controllers with. For the controllers we tested previously in Chapters 4 and 5 (TopN, TopN Exclusion, Intensity Non-Overlap) we will simply continue using the previously-best parameters. These are shown again in Table 6.1. As we mentioned in Section 3.4.2, data acquisition was run in negative-only mode due to constraints on instrument availability, but the sample types themselves have not changed. The time taken to optimise parameters means it may not be realistic newly select them for every experiment. We will also have Intensity Non-Overlap use its SmartRoI variant, because this previously performed best. However, while we use the TopNEXt implementation of TopN Exclusion ("TopNEx") in order to use matching-generated inclusion windows, we use it with the regular DEW to give an idea of the performance gain when adding it directly to TopN Exclusion.

We will also include a new method in our comparisons: DsDA. A fuller description can be found in Section 2.3.3, but DsDA is a pre-scheduled method which uses XCMS to pick peaks between each run. Our main experiments will consequently use XCMS to construct matching and evaluation target lists to ensure a fair comparison with DsDA — and DsDA will use the same set of XCMS parameters for its peak-picking as will be used for both target lists (values can be found in Section 3.2). It should be noted that DsDA does not use XCMS' equivalent of the MZMine join aligner — it groups peaks if their m/z falls within a ppm tolerance and they overlap at all in RT. We have left this as-is in order to minimise the number of changes we made to the method and its implementation.

Other than the XCMS parameters, DsDA has two main parameters to choose values for: N and $maxdepth$. So the comparison to the TopNEXt methods would be fair, we chose values by the same optimisation procedure as with TopNEXt i.e. we grid searched values by running re-simulated experiments of the format later shown in Section 6.3.1 and choosing the parameter combination that maximised intensity coverage. The N parameter controls the number of MS2 scans generated per MS1 scan in the schedule DsDA fills in. A higher value of N gives DsDA more opportunities for targeting. But DsDA picks peaks on its previous runs — that means a lower value of N means richer information is given to $centwave$ and thus DsDA will better be able to pick peaks. A setting of $maxdepth = m$ causes DsDA to

completely exclude any previously-acquired peaks on every mth run. This parameter may therefore help if DsDA is prone to missing certain peaks. For N we searched the values 1, 5, 10, 20 and for $maxdepth$ we searched 1, 2, 3, 4 and NULL (NULL meaning to disable $maxdepth$). For the single-sample experiment we used $N = 10$ and $maxdepth = 3$ and for the round-robin experiment we used $N = 20$ and $maxdepth = NULL$.

Shared DDA Parameters	
Ionisation mode	Negative
N	20
Isolation width	1
Min MS1 intensity (λ_{min})	5000
mz_tol	10
min_roi_length	3
DEW Parameters	
rt_tol	60
SmartRoI Parameters	
rt_tol	15
$intensity\_increase\_factor$ (α)	3
$drop\_perc$ (β)	10^{-3}

Table 6.1: Parameters used for DDA fragmentation strategies in our experiments.

The matching algorithm used a TopN scan schedule with $N = 20$, the appropriate set of fullscans for each experiment, and the list of peak intervals produced by XCMS when processing and aligning each set of fullscans. Inclusion windows were created with an RT width of 10 seconds (approximately two full duty cycles) and an m/z width of 10 ppm around the target RT and m/z.

6.3 Main Results

The main experiments in this chapter will share a lot of similarities with those in Section 4.3. We will use the same overall datasets. We will also use the same overall structure for the experiments in terms of sample choice: a single beer repeated in one experiment type and cycling through a subset of beers in round-robin order in the other. However, because this time we will be asking not only how comprehensive our methods are, but also how *robust*, there will be some differences from our previous methodology.

Previously, for each sample type we included we peak-picked one fullscan per sample type to produce a target list to evaluate against. To measure pre-scheduled performance under natural variations between each run of the LC-MS/MS instrument (i.e. robustness) we peak-picked a different "held-out" fullscan of the same sample type for each fullscan the matching had access to. Additionally, because the distribution of peaks varies so heavily between

Fullscan Indices	Num. Peaks (XCMS)	Num. Peaks (MZMine Restrictive)
7	5784	1800
1-6	10385	3499
7-12	9671	3139
1-12	12626	4261
13-16	11804	4279
13-24	16916	6187
25-36	17741	6083
13-36	22293	7590

Table 6.2: Table showing different output peak numbers for different sets of fullscans being peak-picked and aligned (ranges are inclusive). Fullscans 1-12 are all runs using a single beer sample type. 13-36 use four different beer sample types in a repeating pattern. Note that some of these peaks will be below the required intensity threshold and will not be targeted or counted in the evaluation.

individual fullscans (whether due to algorithmic problems with the peak-picking software or genuine variations in the data) we peak-picked one fullscan of a matching sample type per simulated run we wished to perform. The variability in peaks between runs can be seen in how peak numbers grow as files are aligned in Table 6.2. One thing to note is that the total number of peaks is lower than an equivalent peak-picking approach run on the datasets generated for TopNEXt — this is most likely a consequence of the instrument being run in negative ionisation mode.

This approach is in contrast to our previous approach where we would re-use fullscans as long as the sample type was the same. Consequently, for each experiment we collected $2n$ fullscans, where n is number of LC-MS/MS runs. This was so that n fullscans could be used to simulate the entire experiment of n runs and create an evaluation target set, and the other n could be used for the matching methods to plan for that experiment of n runs. The disparity between the two should be an interesting proxy for the variation observed in the real-world. A lab experiment could also have been used for this purpose, but we could not obtain access to an instrument with a suitable API and thus instead collected seed data for simulations.

Using this data, we ran three different types of experiment. Section 6.3.1 follows the format of the TopNEXt experiments from Chapter 4 where fullscans are re-used. Section 6.3.2 uses a different fullscan per each run, but the same set is given to both the matching and the simulator. Finally, Section 6.3.3 describes the experiment where the simulator and matching each have access to a different set of fullscans for the same sample type.

Our experiments test similar sample type choices to previous experiments with TopNEXt, but with fewer total runs (since by nature pre-scheduled methods reach their performance limit in fewer runs). For each choice of how to use fullscans to inform simulation and target

list creation, we show both a case of six repeats of a single beer sample type, and three repeats of four sample types in round-robin order (1-2-3-4-1-2-3-4-1-2-3-4). We use XCMS [74] for peak-picking experiments to ensure fairness for DsDA [136], which uses XCMS internally. The parameters given to XCMS have a very low quality filter (similarly to the "permissive" parameter set described in Section 3.2 where the XCMS parameters can also be found) which allows us to see fragmentation strategy performance against a high density of peaks (a more challenging scenario).

Experiments will use three classes of method: "baselines" previously investigated in Chapters 4 and 5 plus DsDA, pure "matching" methods which use the matching to generate a fully pre-scheduled plan and "inclusion window" methods which augment a baseline with the matching algorithm by using the matching to generate inclusion windows for the baseline method. These are the two forms of workflow shown in Figure 6.1. Because the inclusion windows are implemented through TopNEXt, the inclusion window version of TopNEXt uses the TopNEXt equivalent of TopN Exclusion (see Chapter 4 for details) but the baseline version does not. Intensity Non-Overlap uses the SmartRoI variant because this performed best in Section 4.3 (details of combining Intensity Non-Overlap with SmartRoI are also in Chapter 4).

Finally, because our implementation of the matching workflow is given a set number of runs to plan for in advance and will not prioritise equivalent targets in earlier runs, when we ran an experiment of n runs we also ran all experiments of length $i \leq n$. We did this for both the case of the same beer repeated six times and the different four beers each repeated three times. For the same beer case this meant $(1 + 2 + 3 + 4 + 5 + 6 = 21)$ total runs to produce all experiments of length up to six (and similarly for the different beers case). We present results for both the run-by-run performance in the experiment of greatest length and the final result for each of these experiments of length up to the maximum. We will be focusing more on the latter case (as it is likely more reflective of how the method would be used in reality) but it is interesting to see how pre-scheduled matching methods allocate scans (particularly when considering the full assignment variants). Furthermore, because there is a possibility of using the inclusion window methods greedily (it is not known prior to running how the DDA component will perform) the run-by-run numbers may be relevant to this use-case.

6.3.1 Re-used Seed Data

We will first consider the case where a single fullscan is used for each sample type when generating simulations and target lists. This is most similar to the TopNEXt experiments in Chapters 4 and 5, but is not one-to-one comparable due to significantly different numbers of peaks resulting from the seed data being collected in negative ionisation mode and target lists being generated via XCMS.

Figure 6.6 shows this same fullscan setup for an experiment of the same beer sample type being repeated six times. To explain the legend, "INO" is Intensity Non-Overlap (SmartRoI), TopNEX is the TopNEXt implementation of TopN Exclusion (they are two slightly different variations of the same method), TS Matching is the two-step matching with R being the variant using $recursive$ and N being the variant using $nearest$, and a "+" (as in TopNEX + and INO +) denotes the use of matching-generated exclusion windows.

For the baseline methods studied in our work with TopNEXt the trend is roughly the same. TopN is a flat line well below the performance of those methods which consider information from multiple sample types. TopN Exclusion and Intensity Non-Overlap have comparable coverage, and Intensity Non-Overlap has a substantial (approx. 10% of total) advantage in intensity coverage. Coverage caps at around 80% and intensity coverage at around 60% after six runs. This is because of the low level of filtering with our XCMS settings. While we won't draw close comparisons due to the differing ionisation modes of the two datasets, it should be noted that this is approximately the same final threshold observed using the "permissive" peak-picking parameters in Section 5. The "permissive" MZMine parameters are closer to this XCMS setup in terms of quality threshold than the "restrictive" peak-picking parameters.

We can now also compare these methods to methods not used in the original TopNEXt study. Of these newcomers, DsDA is also very competitive, performing best by a small margin in coverage and intensity coverage out of any of the baselines. The pre-scheduled two-step matching method is predictably very strong in this best-case scenario of a noiseless environment where it knows exactly what is coming. The pre-scheduled two-step matching methods outperform all other methods massively, gaining effectively total coverage in an experiment of two runs and over 90% intensity coverage in an experiment of three runs (Figure 6.6B). Note that the two-step matching is roughly 20% ahead in coverage compared to any other method but is around 30% ahead in intensity coverage, meaning it is even better in terms of intensity coverage. This speaks to the available design space for future methods which attempt to optimise intensity coverage.

We also see the added value of the two-step matching procedure compared to the original matching technique, because while the unweighted matching has close to identical coverage to the two-step matching, it has much lower intensity coverage. It does not outperform the majority of the baselines in intensity coverage despite much higher overall coverage, showing it acquires spectra at very low average precursor intensity. After the second point on Figure 6.6, when it maximises coverage, the intensity coverage approximately flatlines because the unweighted matching is not trying to improve it. There are some minor fluctuations, but this can be attributed to the solver algorithm not making exactly the same choices when presented with a larger graph.

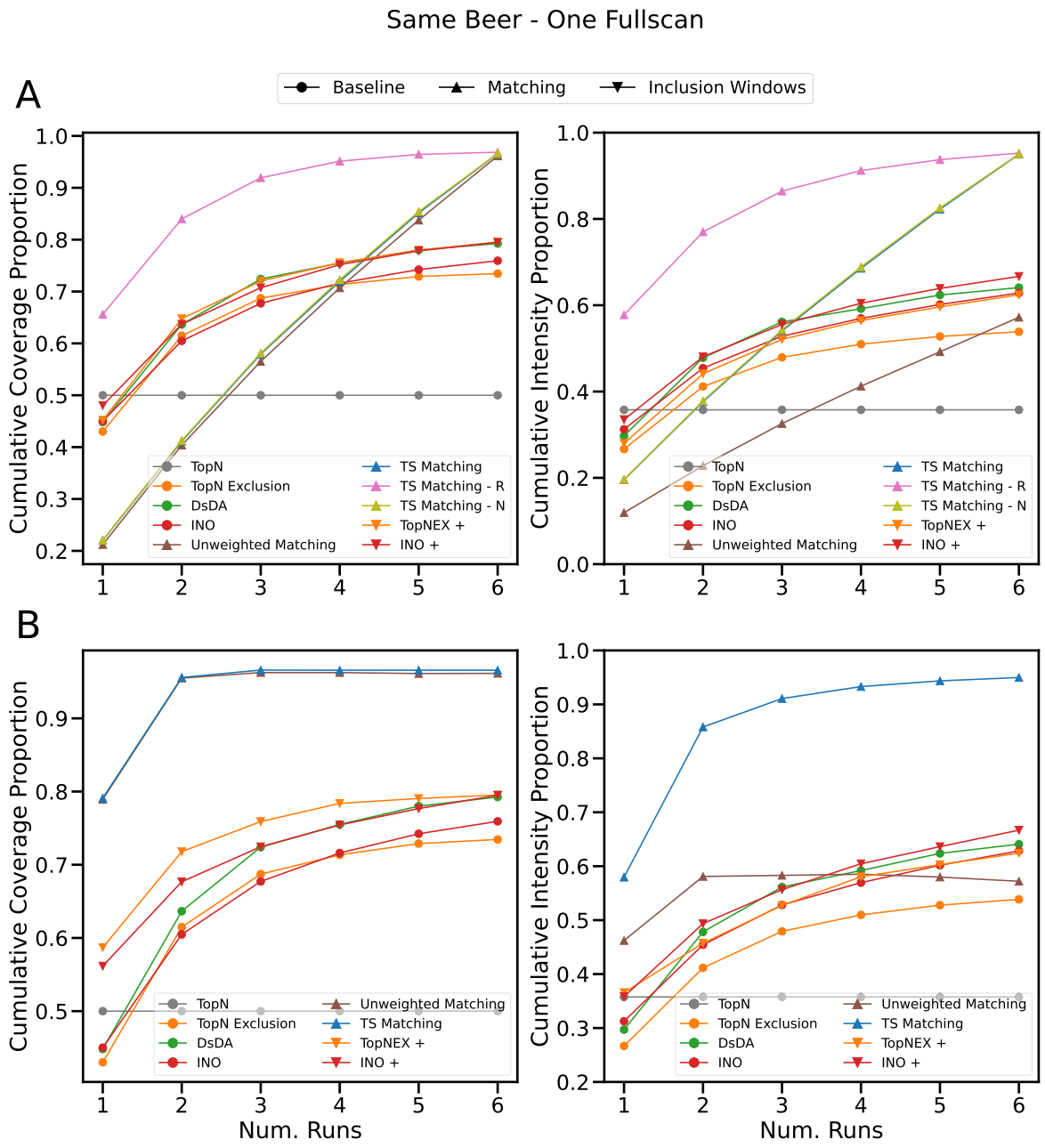

Figure 6.6: Simulated experiment using the same beer repeated for six runs. A single fullscan was used to generate simulations and target lists, and XCMS was used to generate target lists for both the matching algorithm and the evaluation. **A:** shows performance over different runs in a single experiment. **B:** shows performance over separate experiments of different run numbers.

We can also see that the recursive variant of the full assignment distributes its scans much more aggressively over the total number of runs in Figure 6.6A. Any target arbitrarily chosen for a later run by the matching algorithm may arbitrarily have a duplicate assigned to an earlier run by $recursive$, so the coverage line rises much faster on the run-by-run comparison. Note that because this experiment is noiseless all full assignment variants will have equal performance after the final run, so they have been combined into a single line for Figure

6.6B.

Finally, the inclusion method windows are a strict improvement on their respective baseline methods. This improvement is especially pronounced at lower levels of runs, where the inclusion does more to help guide the behaviour. With more runs, the iterative exclusion behaviour of the regular baselines allows them to close the gap somewhat. Interestingly, although inclusion windows are still a net benefit to Intensity Non-Overlap, the performance increase is minor as the number of runs increases. Conversely, without inclusion windows TopN Exclusion ends with the weakest coverage of the baselines and a severely lagging intensity coverage. But when inclusion windows are added its coverage is the best of the cluster of non-pre-scheduled methods and its intensity coverage is competitive. The likely implication is that inclusion windows are more of a complement to TopN Exclusion's inflexible hard exclusion behaviour, whereas while they add information to Intensity Non-Overlap they do not interface as well with its more complex behaviour.

Figure 6.7 shows the same setup where fullscans are allowed to be reused, but for the case where four beers are repeated three times each i.e. there are four fullscans used three times each by both the simulator and the matching algorithm. Again, similar results to the original tests of TopNEXt can be observed: TopN flatlines once it has seen all the sample types, and TopN Exclusion seems to perform comparatively poorly on different-samples cases, especially relative to Intensity Non-Overlap.

It should be noted that the final thresholds which these baseline methods reach (60% to 80%) are more similar to the experiments with permissive parameters in Section 5.3, but in that case the relative performance advantage of Intensity Non-Overlap narrowed. The performance advantage for Intensity Non-Overlap was seen in the *main* experiments in Section 4.3. The lack of consistency may be the result of using parameters more similar to the "permissive" parameters, but on a negative ionisation mode dataset with fewer peaks (again suggesting the data is not directly comparable).

We also see that the pre-scheduled matching can effectively obtain total coverage by only seeing each sample type once, and close to 90% intensity coverage in the same span. The same observations can be made regarding how aggressively the recursive variant spreads its scans, and that the unweighted matching has remarkably poor intensity coverage given its comparable coverage.

For the inclusion window methods, we also see again that their use slightly improves the performance of Intensity Non-Overlap. Additionally, again, TopN Exclusion is given a substantial improvement by the use of inclusion windows, but this time, it appears to be substantially stronger (around twice as much coverage gained). This gives increased credence to the idea that inclusion windows help compensate for some of the natural weaknesses of TopN Exclusion, as TopN Exclusion also struggled with this experiment structure when being tested

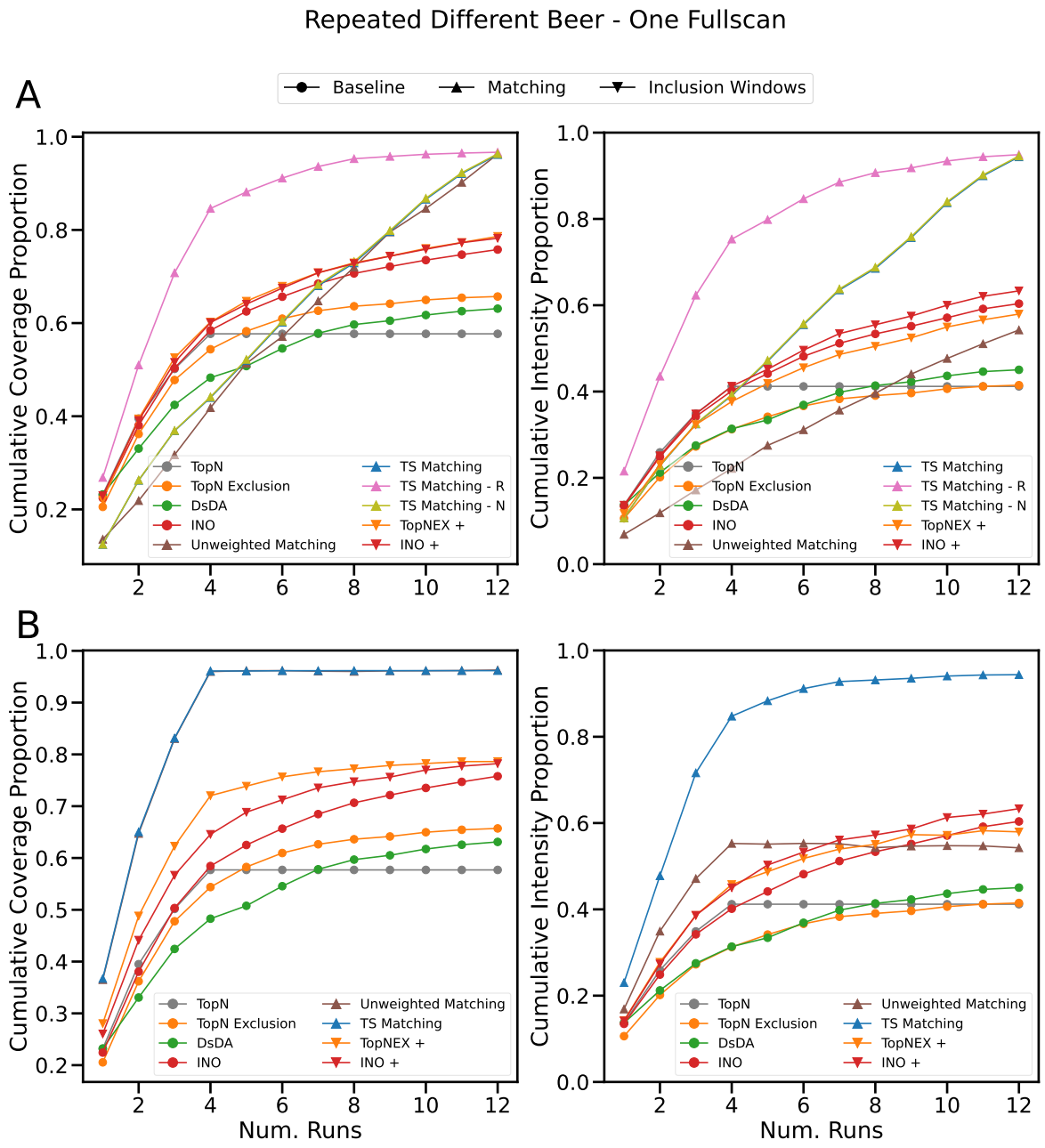

Figure 6.7: Simulated experiment using four beers repeated three times each in round-robin order (1-2-3-4-1-2...). A single fullscan was used to generate simulations and target lists, and XCMS was used to generate target lists for both the matching algorithm and the evaluation. **A:** shows performance over different runs in a single experiment. **B:** shows performance over separate experiments of different run numbers.

against TopNEXt.

One thing that has changed, however, is that DsDA is no longer effective, only barely outperforming TopN (and only once it has seen all the sample types twice). This makes a certain kind of sense: DsDA is by design trying to target commonalities in the sample set, and which common peaks are observed will grow as more runs are observed and shrink as the under-

lying sample types get more biologically diverse. It may therefore require many runs (the original work tested it with 20, 40 and 50 [136]) and it may be the case that above a certain level of sample diversity in a multi-sample experiment DsDA will fail to converge to collecting the non-shared parts. If it is the case that DsDA will hit a fundamental limit at a certain level of sample diversity, then in practice it will be necessary to perform data acquisition with DsDA for each sample type separately.

6.3.2 Per-run Seed Data

Now we will move into a different type of experiment from those presented in Chapters 4 and 5. Figure 6.8 shows the same beer repeated experiment, but with a different fullscan being used for each run by the simulator, evaluation and the matching algorithm. The first obvious change from the experiments we have seen previously is that TopN is no longer a flat line: the increased number of fullscans being used causes there to be a greater number of peaks only in a certain subset of scans, so there are always new peaks for TopN to acquire. Interestingly, this also makes TopN much more competitive relative to the other methods. TopN Exclusion and DsDA barely outperform TopN, though Intensity Non-Overlap does maintain a significant advantage.

The overall high performance of the pre-scheduled two-step matching has not changed, but because more peaks are being introduced per run, we do see that it takes longer to reach its apex, and that apex is closer to 90% than before. What is interesting is that the advantage for the baseline methods in having inclusion windows added seems significantly more pronounced (e.g. a 20% coverage increase for TopN Exclusion). It may be the case that having prior information about the distribution of these added peaks provides a significant advantage, but (as we will see) this advantage is not possible in a more realistic scenario, given that peaks randomly showing up in one run or another will happen, by definition, in a random order. If, however, it is the case that some of these peaks are below the normal limit of detection given only a single run, but the alignment of multiple runs allows their detection, then this improvement may partially translate into reality.

Now we will consider the repeated different beers in Figure 6.9. Surprisingly, TopN actually significantly outperforms both TopN Exclusion and DsDA. In DsDA's case, this is likely a matter of looking for common peaks to target in an environment where both sample types and individual peaks are constantly shifting. For TopN Exclusion, this seems like an exacerbated version of the drop in performance that multiple sample types cause. However, it is possible that part of this could be attributed to TopN Exclusion having exclusion windows that are too wide, or to XCMS inappropriately splitting peaks that should be merged together. If this were true it would speak to the well-known difficulties in LC-MS/MS for configuring tools

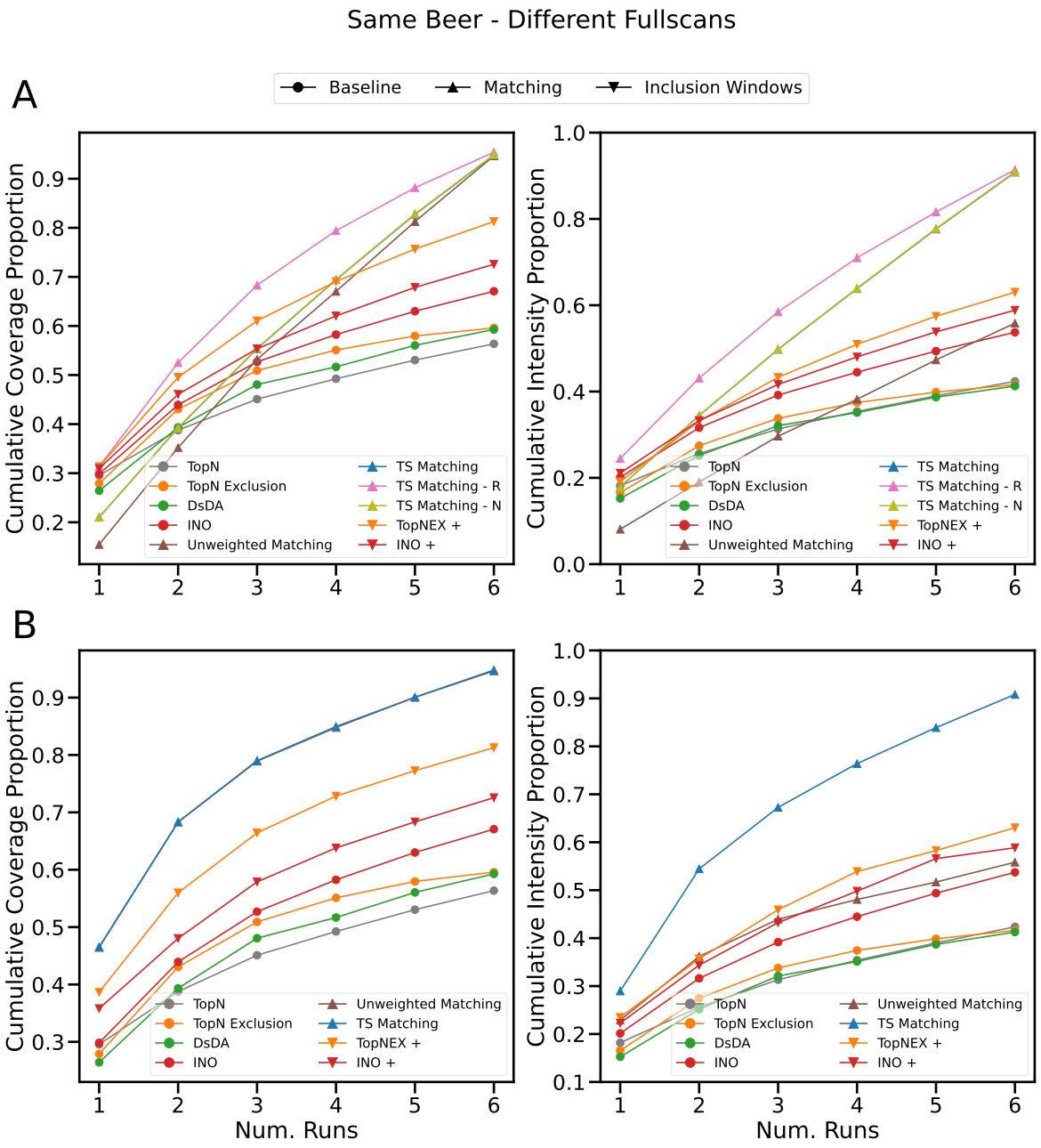

Figure 6.8: Simulated experiment using the same beer repeated for six runs. A different fullscan for each of the six runs was used to generate simulations and target lists, and XCMS was used to generate target lists for both the matching algorithm and the evaluation. **A:** shows performance over different runs in a single experiment. **B:** shows performance over separate experiments of different run numbers.

and in comparing results across different configurations. Regardless, Intensity Non-Overlap shows strong performance and is the best-performing baseline.

For the pre-scheduled two-step matching, we see the same effect from the same beer case where performance is very high, but takes a lot longer to accrue and caps at a slightly lower threshold. The trends with the inclusion window methods are also the same as before, other

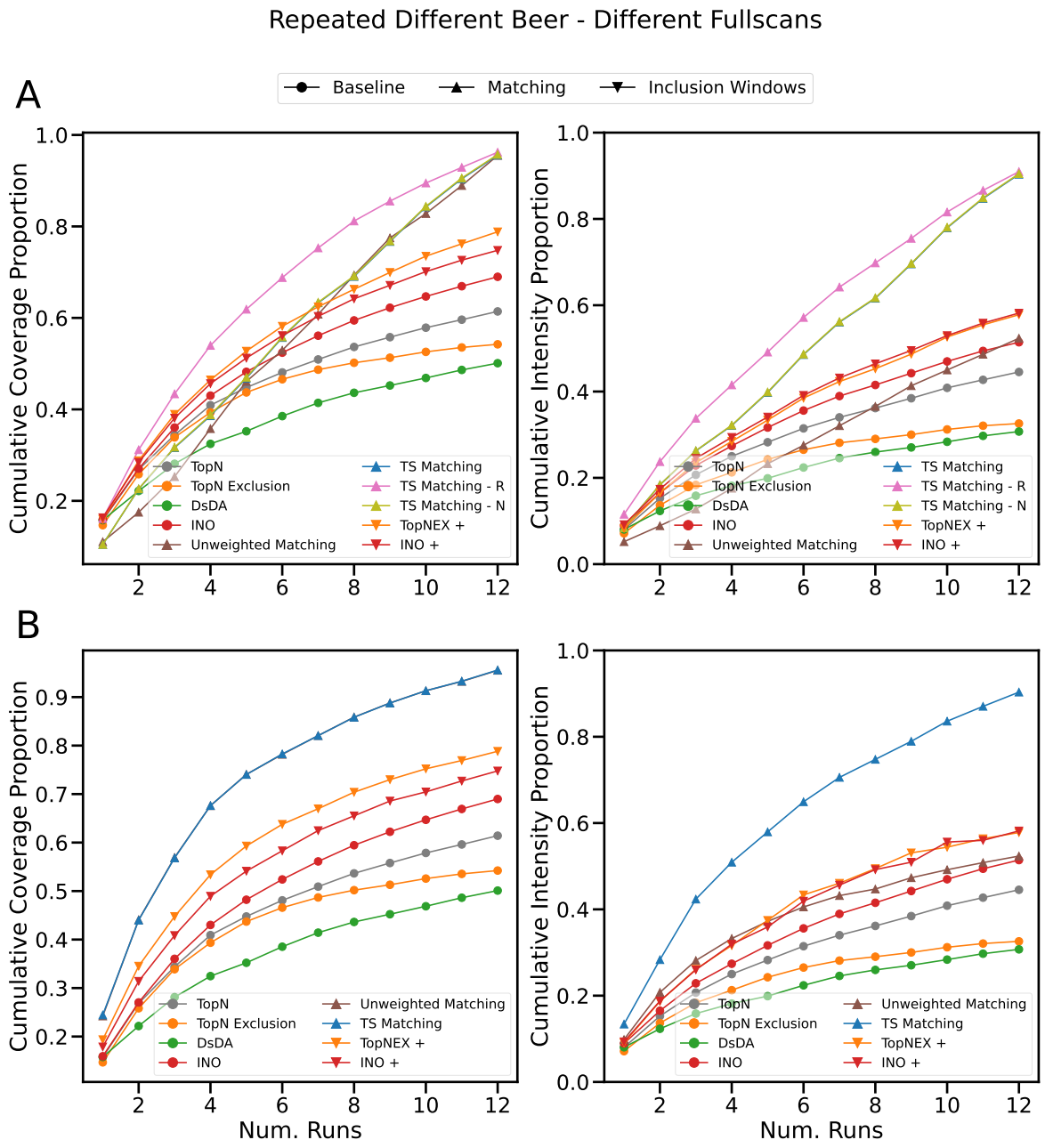

Figure 6.9: Simulated experiment using four beers repeated three times each in round-robin order (1-2-3-4-1-2...). A different fullscan for each of the twelve runs was used to generate simulations and target lists, and XCMS was used to generate target lists for both the matching algorithm and the evaluation. **A:** shows performance over different runs in a single experiment. **B:** shows performance over separate experiments of different run numbers.

than the fact that the difference between the inclusion window variant of TopN Exclusion and its baseline version has become even larger: again suggesting it has some corrective action.

6.3.3 Paired Seed Data

So far we have been looking at what is essentially the "best-case scenario" for our matching methods (both in terms of pre-scheduling and inclusion windows) but it is now time to challenge it with a difficult (and more realistic) scenario. In these experiments we will have two sets of fullscans, one to generate a target list for the matching algorithm, and the other to seed the simulator and create a target list for the evaluation. This causes the matching to see a dataset of the same underlying sample types in the same order, but with different peaks, and with peaks showing up in different runs in the run order (as would happen in reality). This is intended to mimic the real-world scenario where one could run an initial experiment, collecting fullscans to plan an acquisition, then run the acquisition afterwards. We have given these two stages the same number of fullscans to account for the fact that peak numbers generally increase as you peak-pick a larger set of fullscans (there is a possibility that this would be too many fullscans to collect in a realistic scenario, but we leave this to the reader's discretion).

Figure 6.10 shows the results of this setup on the experiment with the same beer repeated. The baselines were not re-run between this experiment and the previous one, as they do not take a target list as input, so are displayed at the same values as before. Predictably, the performance of the matching-based methods is no longer as good. They are still among the best performing methods, however. Pre-scheduled two-step matching with recursive assignment, and both inclusion window methods have similar performance to Intensity Non-Overlap by the sixth run (with Intensity Non-Overlap plus inclusion windows having the best performance by a slight margin) but accrue performance faster in the earlier runs.

Something to note is that because we have introduced a source of error for the matching into the process, Figure 6.10B now shows all three full assignment variants of the two-step matching. A and B both show the same content as in previous figures, but now the values between the three can differ, so they have been plotted separately. Notably, the way the recursive variant spreads targets across scans gives it a significant advantage here — $nearest$ barely outperforms the base variant, and they are close to the weaker baselines in coverage after enough runs have passed.

It may be worth observing, however, that while DsDA, TopN and TopN Exclusion catch up to non-recursive variants of the two-step matching in coverage, they still lag behind in intensity coverage (by approx. 5%). In fact, it was also previously observable (if less obvious) in Section 6.3.2 that these three baselines seem to perform slightly worse in terms of intensity coverage relative to coverage, when compared to the other methods. We have seen previously through TopNEXt that this is true for TopN and TopN Exclusion relative to a method like Intensity Non-Overlap, because they don't go back to optimise intensity coverage. Like Intensity Non-Overlap, the matching methods were designed with optimising intensity coverage in mind, so this shows that they are effective at this.

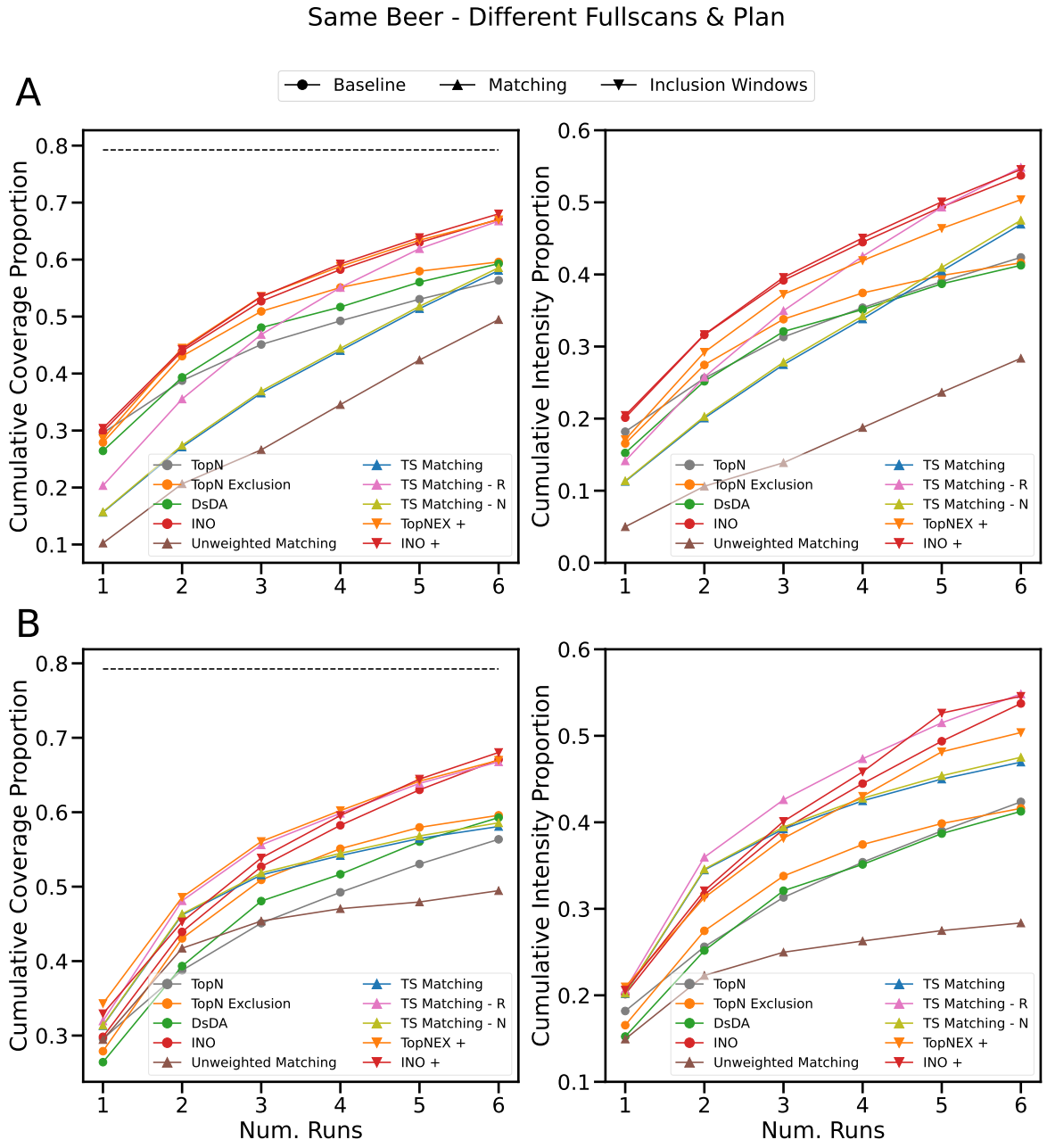

Figure 6.10: Simulated experiment using the same beer repeated for six runs. Two sets of six fullscans were used. One set was used to generate the target list for the matching, and the other was used for the simulations and the target list of the evaluation. XCMS was used to generate target lists for both the matching algorithm and the evaluation. The dotted line indicates the level of overlap between the target lists generated from the two sets of fullscans. **A:** shows performance over different runs in a single experiment. **B:** shows performance over separate experiments of different run numbers.

DsDA, however, was designed with optimising target acquisition in mind — though the DsDA publication measured this by plotting each time a peak was re-targeted, and found that DsDA "oversampled" low-intensity peaks more than TopN. It seems likely that intensity coverage gives us a slightly different view of this phenomenon by directly showing us

the maximum intensity each peak was targeted at. It may be the case that DsDA simply requires more runs to optimise intensity coverage (perhaps with an appropriate setting of the $maxdepth$ parameter) or it may be the case that while it may target peaks more often it does not do so at higher intensity or it may be the case that DsDA's scoring function somehow "saturates" because it is based on the relative rankings of the peaks it sees. These would be interesting questions to answer in future work.

Another thing to consider is that the unweighted matching has the worst performance of any method here, being strictly worse than any other method after the third run. One of the most interesting aspects is that while its intensity coverage has always been bad (after all, it pays no attention to targeting intensity) its coverage is also significantly worse than the two-step matching here — this was not true previously. The likely reason is that trying to optimise intensity coverage leads the two-step matching to target the more stable peak apex, where the unweighted matching may arbitrarily select an edge point that will not be in the peak if it moves between runs. Together with the observations about the recursive variant, this demonstrates the necessity of our updates to the matching technique for it to be practically usable.

One final thing to take note of is that the level of agreement between the target lists has been indicated by a dotted line on the plot. In addition to being peak-picked separately, both sets of fullscans were peak-picked together as one set and aligned. The dotted line shows how many of the peaks listed in the matching's target list were included in the evaluation's target list. In other words, if a matching method hit every target in its list (and only those targets) it would receive a score of just over 80%. Of course, the actual score for even the best performing pre-scheduled matching is much lower, showing that it is missing a good number of its targets due to them, for example, not being in the right run, or right location, or not appearing at all.

Finally we will consider the case where there are two different sets of fullscans for four beers repeated three times each, shown in Figure 6.11. Surprisingly, under these conditions, TopN is a very effective method, beating the majority of its competitors. This includes the unweighted matching, DsDA, TopN Exclusion, and the regular and $nearest$ variants of the two-step matching. It also generally remains the case that inclusion windows are a mild improvement for Intensity Non-Overlap, which finishes as the most effective method, but the pre-scheduled two-step matching using $recursive$ gains performance faster and is competitive by the end. TopN Exclusion with inclusion windows is also competitive with both but has generally worse intensity coverage and a slightly lower coverage point by the end. We also see that TopN Exclusion and DsDA do even more poorly in intensity coverage compared to Figure 6.10 which again suggests they struggle with this multi-sample case. TopN has caught up with the two-step matching methods, but notably the advantage it has (presumably because of the large differences in peaks between runs) is more in coverage

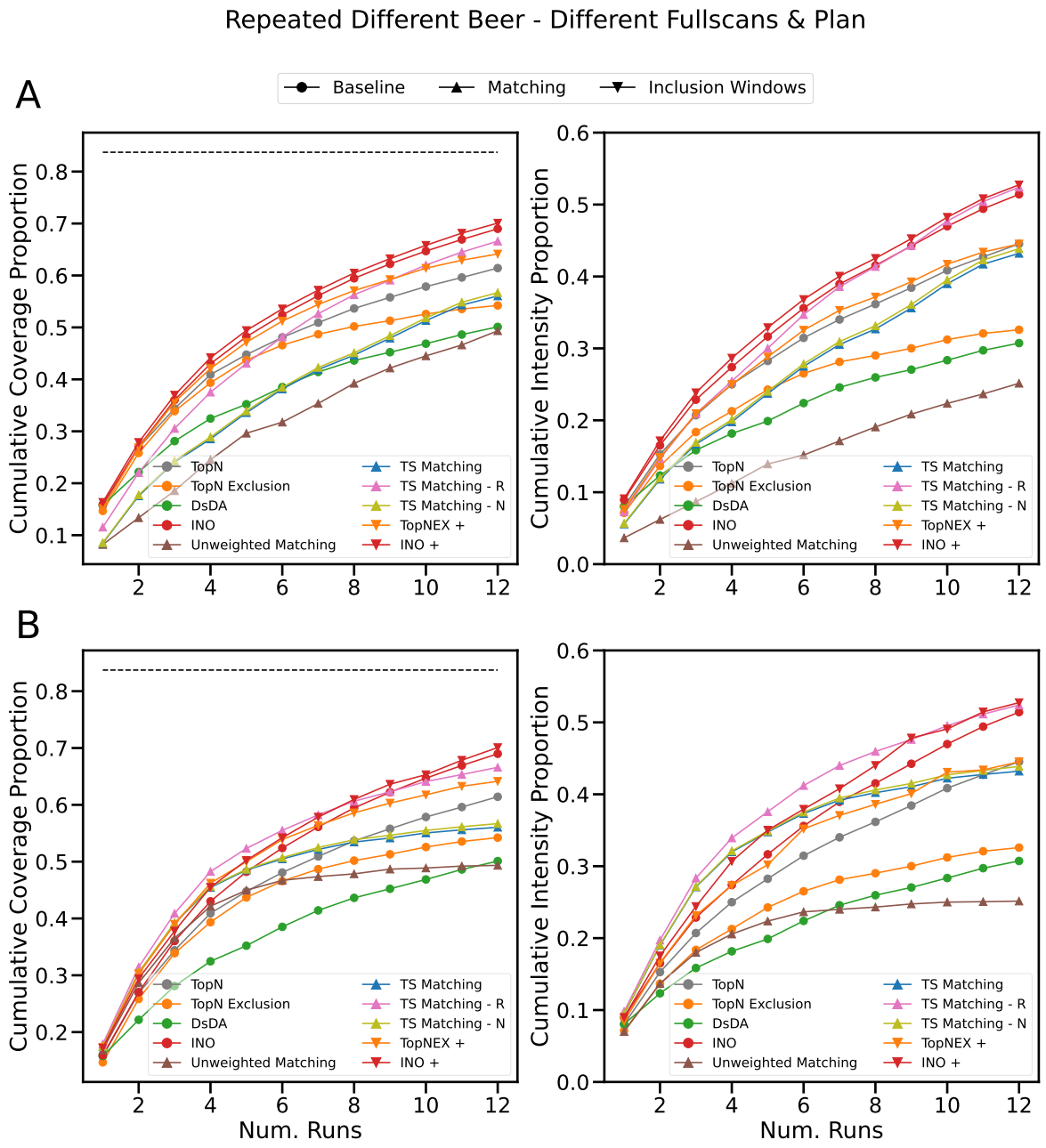

Figure 6.11: Simulated experiment using four beers repeated three times each in round-robin order (1-2-3-4-1-2...). Two sets of twelve fullscans were used. One set was used to generate the target list for the matching, and the other was used for the simulations and the target list of the evaluation. XCMS was used to generate target lists for both the matching algorithm and the evaluation. The dotted line indicates the level of overlap between the target lists generated from the two sets of fullscans. **A:** shows performance over different runs in a single experiment. **B:** shows performance over separate experiments of different run numbers.

than intensity coverage.

6.4 Results with MZMine (Restrictive)

The results in Section 6.3 showed XCMS results with a low quality-filter. We used XCMS both because it was fairer to DsDA (which uses XCMS internally) and the low quality-filter made the peak scheduling problem harder and allowed us to show that the matching algorithm is in theory capable of resolving a schedule for a very high number of peaks at

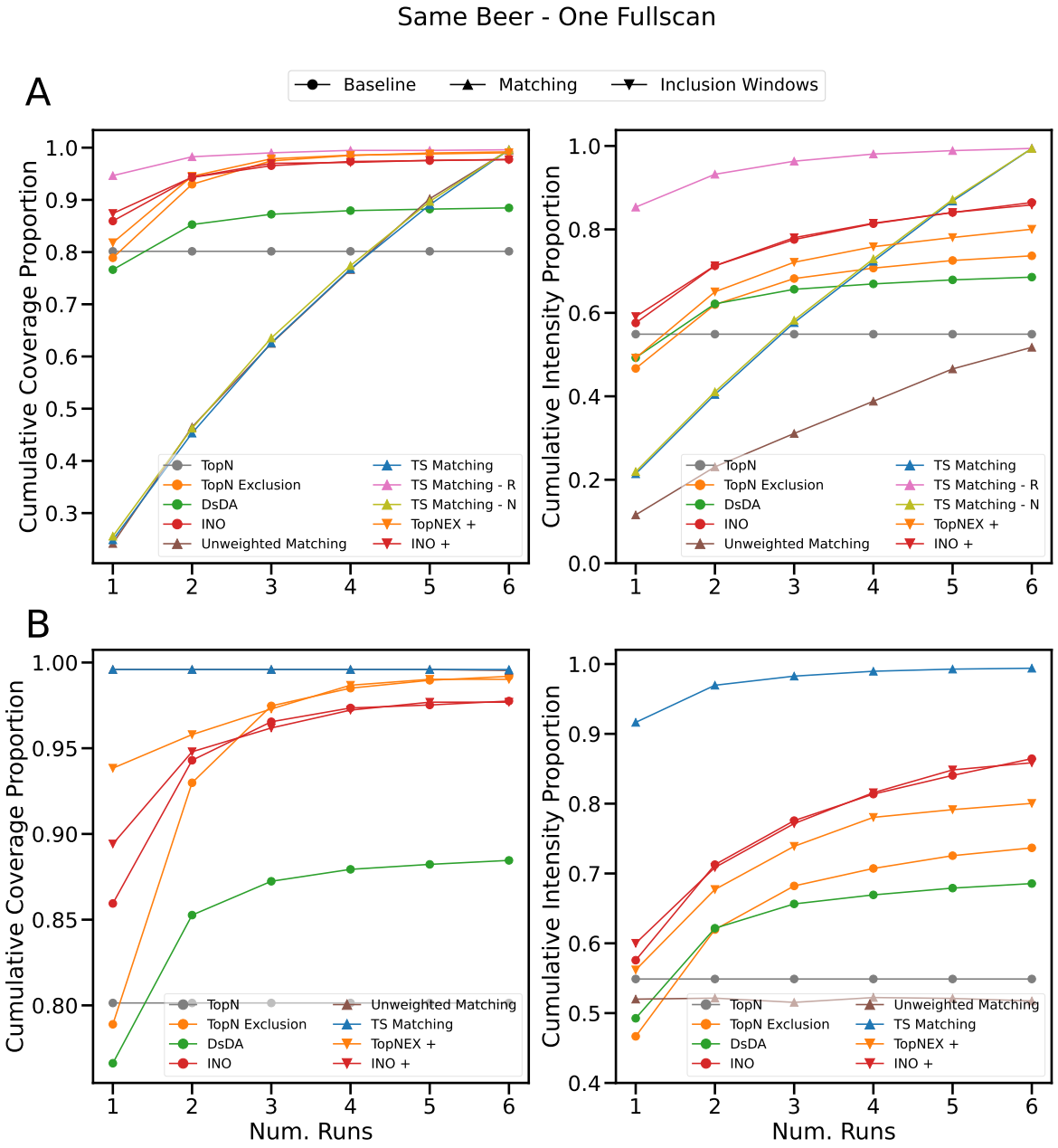

Figure 6.12: Simulated experiment using the same beer repeated for six runs. A single fullscan was used to generate simulations and target lists, and MZMine (restrictive parameter set) was used to generate target lists for both the matching algorithm and the evaluation. **A:** shows performance over different runs in a single experiment. **B:** shows performance over separate experiments of different run numbers.

once.

However, it is also interesting to ask what results are like for the restrictive MZMine parameter set which produces a much smaller, but higher-quality, set of peaks. To illustrate this, Table 6.2 shows us that there is a roughly 3:1 ratio between XCMS peaks and MZMine (Restrictive) peaks. This is a bit lower than the ratio between the two MZMine parameter sets on the TopNEXt experiments' dataset because the number of peaks in the more restrictive parameter set drops off less sharply when given the negative ionisation mode data we have used this time. Nonetheless, results from the two different peak-picking parameter sets are very different and produce very different results when evaluating a fragmentation strategy.

Figure 6.12 shows the results of this procedure for the same beer, single fullscan case (i.e. equivalent to Figure 6.6 but using different peak-picking to construct target lists). It is immediately apparent, like with the comparison between restrictive and permissive parameters between Chapters 4 and 5, that the scale on these plots is very different to those generated with our XCMS peak-picking setup. Essentially every method (minus TopN and DsDA) gets to 100% coverage and over 70% intensity coverage. The two-step matching is able to get to roughly 100% coverage and over 90% coverage in only a single run.

Broadly, we see some of the trends we have come to expect from this data: pre-scheduled matching is very good when it has perfect knowledge of what is coming, inclusion windows improve both Intensity Non-Overlap and TopN Exclusion, but this applies moreso for TopN Exclusion, and TopN has very poor performance when there is no variation in the underlying data (although it is worth remembering there will always be variation in reality). Compared to the XCMS-based experiments, both variants of Intensity Non-Overlap show somewhat poorer performance in coverage compared to TopN Exclusion with inclusion windows, but Intensity Non-Overlap maintains its intensity coverage advantage. One notable change is that the performance of the unweighted matching looks worse relative to Figure 6.6: but this is a function of the fact that it is generally easier for other methods to obtain coverage.

At a glance the most interesting trend is that DsDA appears to saturate below 90% coverage and 70% intensity coverage. This could be because its scoring function is prone to saturation or because of the mismatch between the XCMS and MZMine parameter settings (we did not adjust between these experiments) or some combination of both (i.e. the much higher number of XCMS peaks makes the saturation problem worse). It is therefore important not to read too deeply into DsDA's performance in this section, but it is notable that it (albeit like every other method) manages to target the majority of the evaluation target list in a scenario where it has a different target list due to using XCMS. These difficulties with comparisons using DsDA in this scenario also speak to the need for a more integrated platform for testing and running LC-MS/MS fragmentation strategies (i.e. continued development of ViMMS).

Examining Figure 6.13 we also observe the same basic trends but for the 4-3 repeated differ-

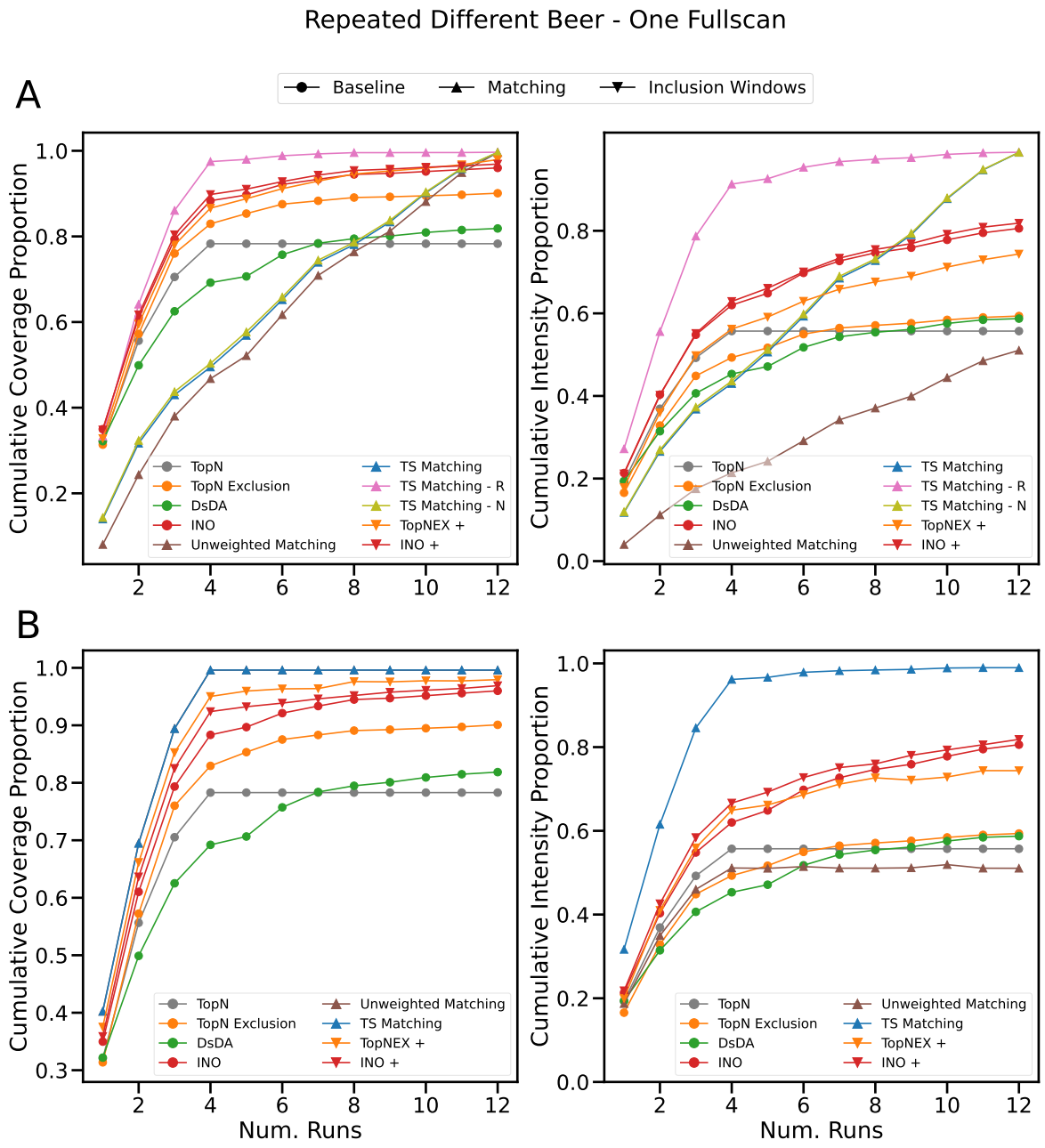

Figure 6.13: Simulated experiment using four beers repeated three times each in round-robin order (1-2-3-4-1-2...). A single fullscan was used to generate simulations and target lists, and MZMine (restrictive parameter set) was used to generate target lists for both the matching algorithm and the evaluation. **A:** shows performance over different runs in a single experiment. **B:** shows performance over separate experiments of different run numbers.

ent beers experiment (with re-used fullscans). The pre-scheduled matching gets effectively total coverage and almost total intensity coverage (in the case of the two-step matching) once it has seen all four sample types. Many methods again eventually converge to approx. 100% coverage, but likely due to the increased complexity of this experiment compared to the same beer case, the two-step matching also has much more intensity coverage than any other methods (approx. 100% vs approx. 80%) by the end of the experiment as well. We also

see again that TopN Exclusion with inclusion windows is the next most effective method in coverage, but Intensity Non-Overlap outperforms it in intensity coverage. DsDA also seems to saturate after slightly outperforming TopN — both of these trends are similar to Figure 6.7.

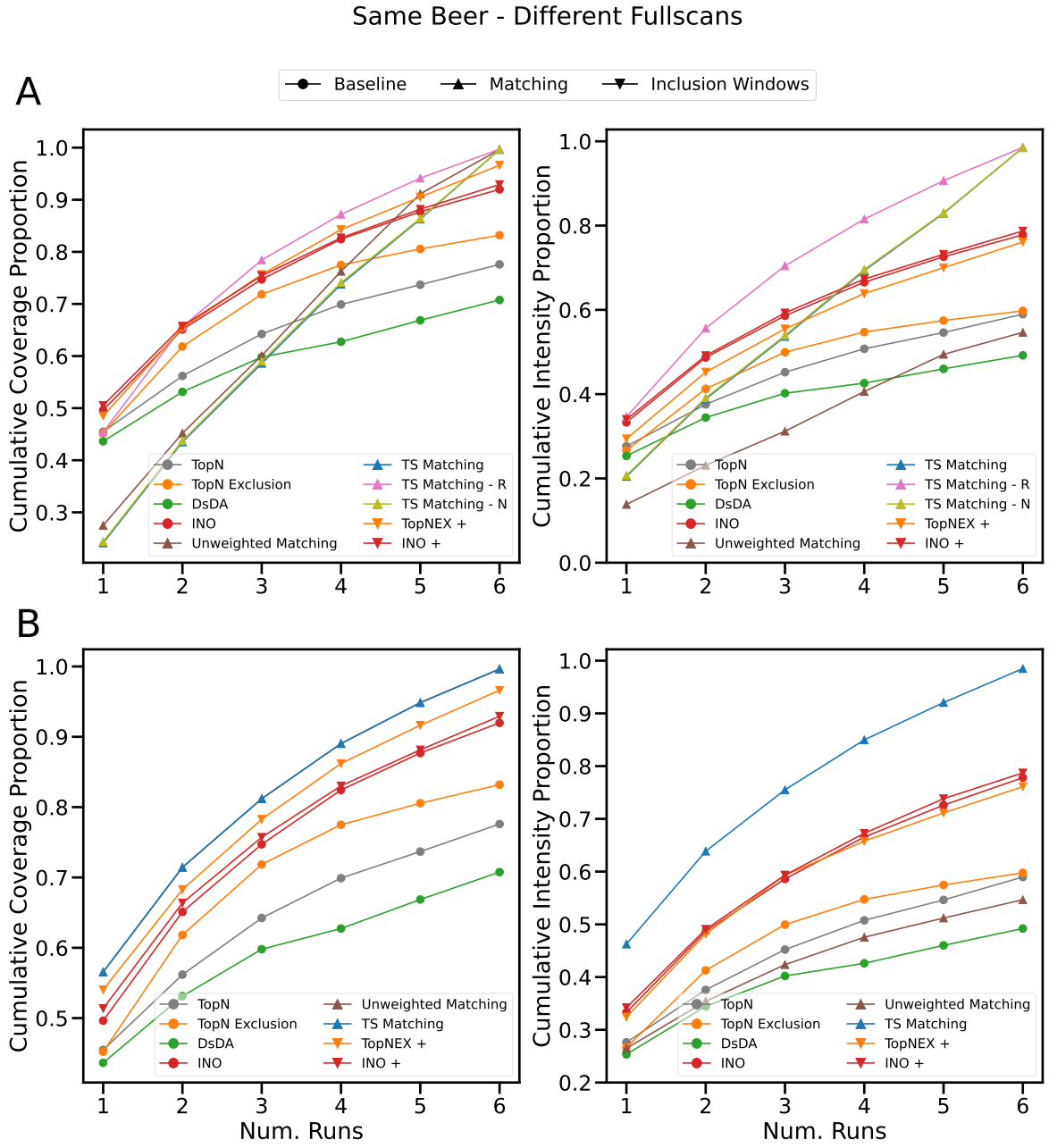

Figure 6.14: Simulated experiment using the same beer repeated for six runs. A different fullscan for each of the six runs was used to generate simulations and target lists, and MZMine (restrictive parameter set) was used to generate target lists for both the matching algorithm and the evaluation. **A:** shows performance over different runs in a single experiment. **B:** shows performance over separate experiments of different run numbers.

Figures 6.14 and 6.15 show the different fullscan scenario (using one set of fullscans shared between the matching algorithm and simulator) for the same beer and repeated different beer

(4-3) cases respectively. These essentially reiterate the same basic trends we have seen: like Figures 6.8 and 6.9 performance is more slowly gained over the total length of the experiment (due to more target peaks appearing with each run) but like the previous experiments using MZMine we also see that the numbers overall are higher. The pre-scheduled matching still has very strong performance given that it knows what is coming, but the inclusion window

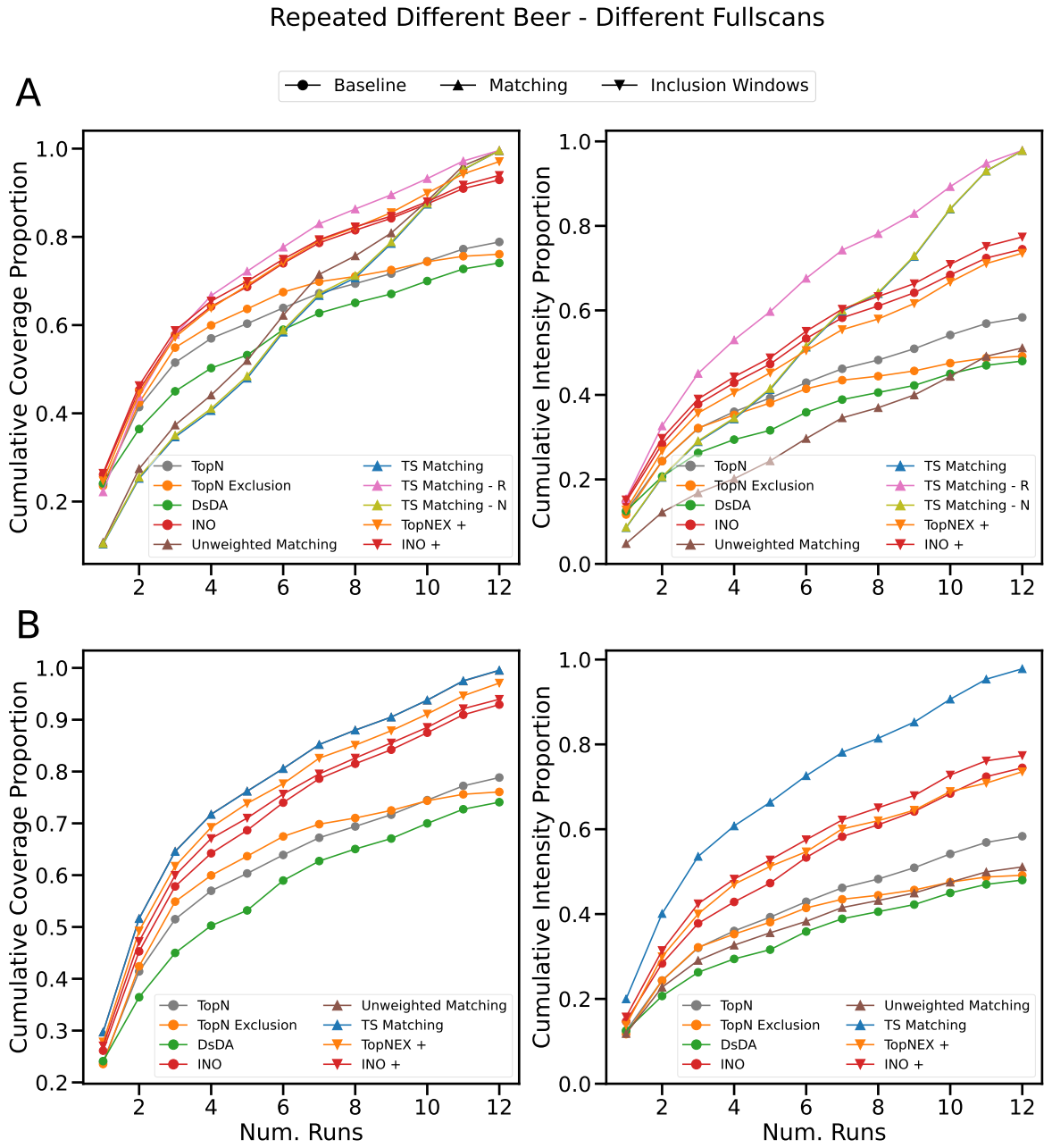

Figure 6.15: Simulated experiment using four beers repeated three times each in round-robin order (1-2-3-4-1-2...). A different fullscan for each of the twelve runs was used to generate simulations and target lists, and MZMine (restrictive parameter set) was used to generate target lists for both the matching algorithm and the evaluation. **A:** shows performance over different runs in a single experiment. **B:** shows performance over separate experiments of different run numbers.

methods and base Intensity Non-Overlap track it more closely due to the overall number of peaks being lower. We also see the performance of DsDA and TopN Exclusion falling relative to TopN in Figure 6.15 as we saw for the repeated different beer experiment before.

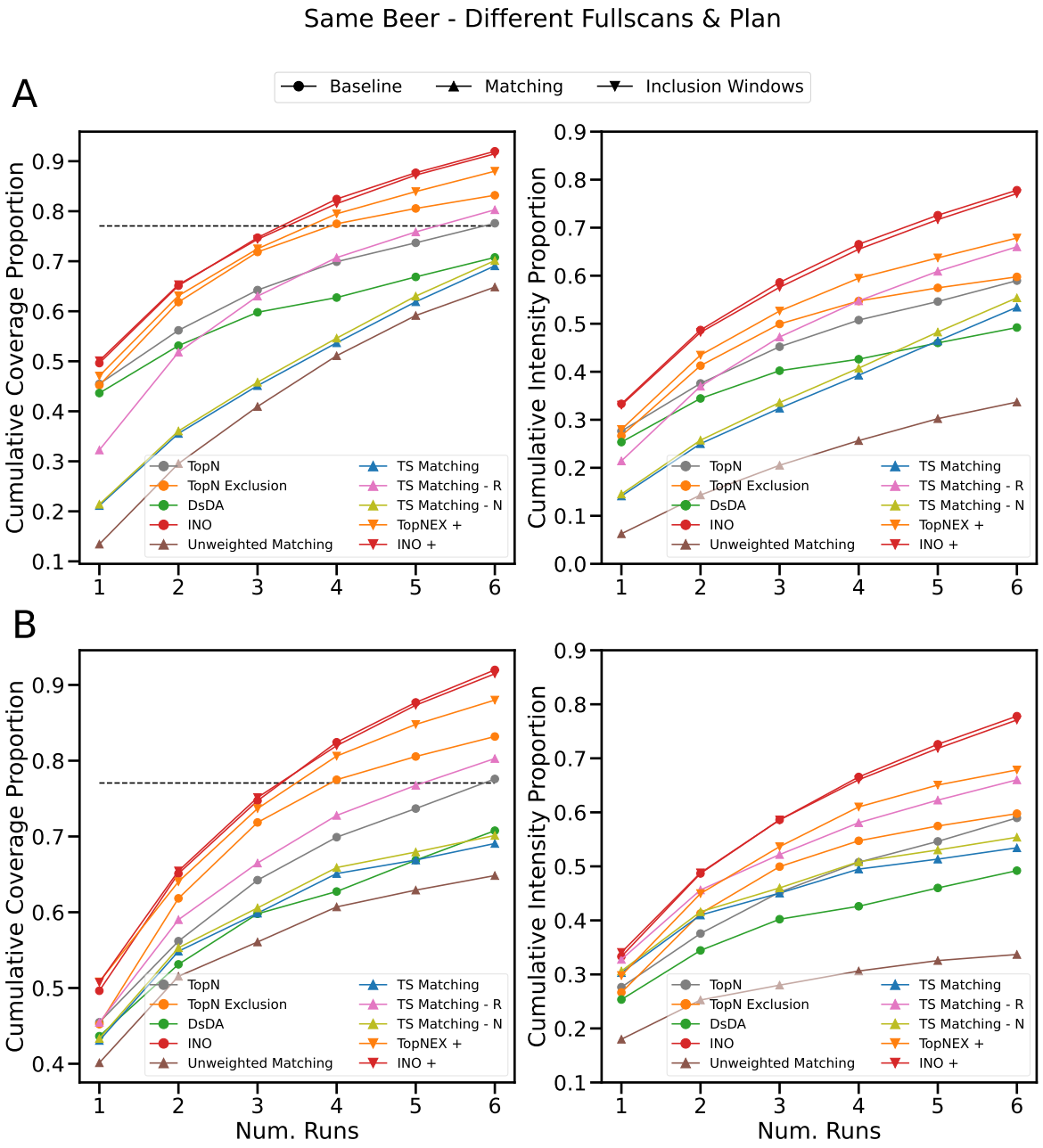

Figure 6.16: Simulated experiment using the same beer repeated for six runs. Two sets of six fullscans were used. One set was used to generate the target list for the matching, and the other was used for the simulations and the target list of the evaluation. MZMine (restrictive parameter set) was used to generate target lists for both the matching algorithm and the evaluation. The dotted line indicates the level of overlap between the target lists generated from the two sets of fullscans. **A:** shows performance over different runs in a single experiment. **B:** shows performance over separate experiments of different run numbers.

Finally, we will consider the case where a different set of fullscans is used to create the

target list for the matching algorithm versus the simulator and evaluation, but under the restrictive MZMine parameter set. Figure 6.16 shows this for the same beer case. Like Figure 6.10, the performance of the matching-based methods drops severely in this more

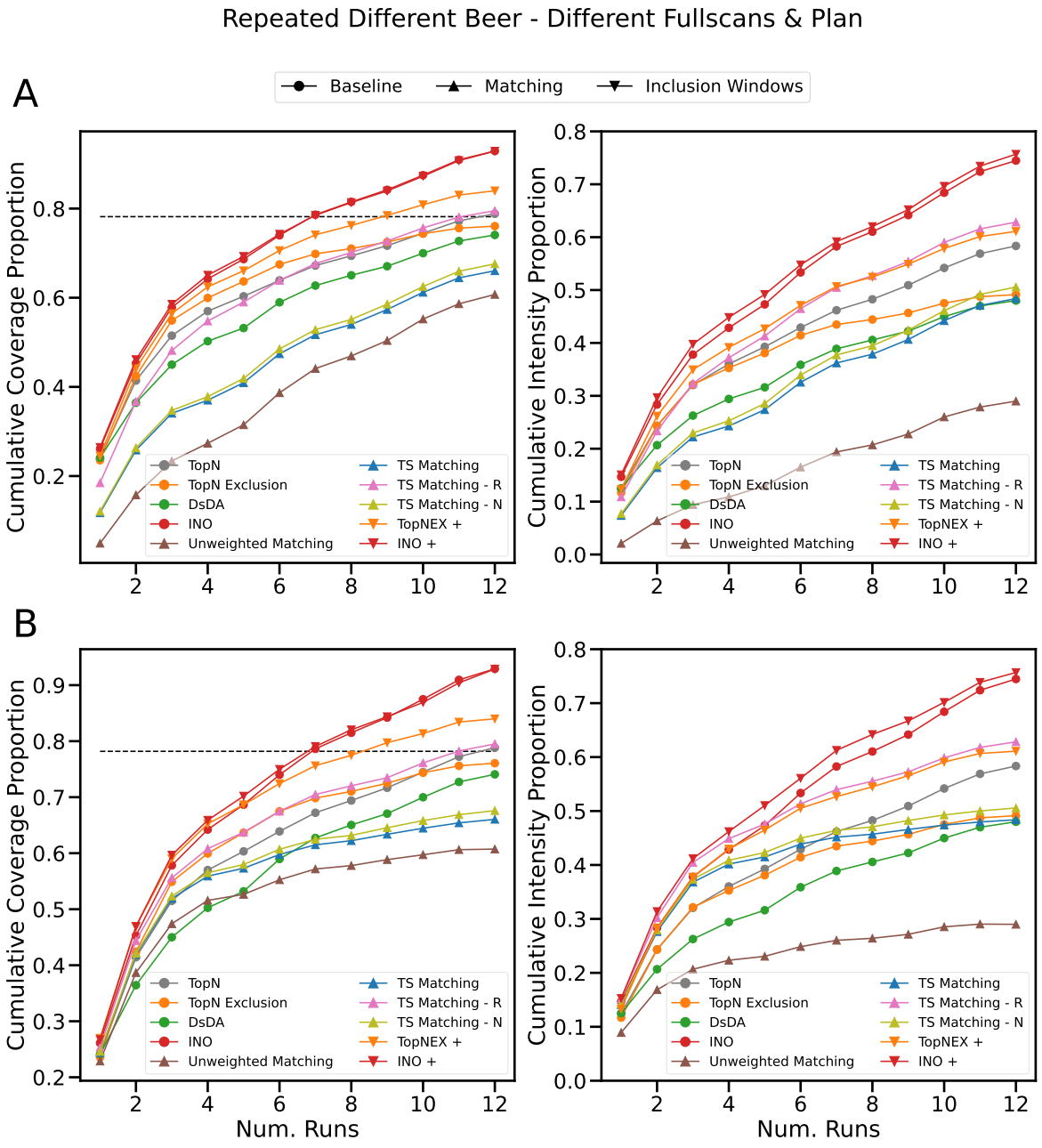

Figure 6.17: Simulated experiment using four beers repeated three times each in round-robin order (1-2-3-4-1-2...). Two sets of twelve fullscans were used. One set was used to generate the target list for the matching, and the other was used for the simulations and the target list of the evaluation. MZMine (restrictive parameter set) was used to generate target lists for both the matching algorithm and the evaluation. The dotted line indicates the level of overlap between the target lists generated from the two sets of fullscans. **A:** shows performance over different runs in a single experiment. **B:** shows performance over separate experiments of different run numbers.

realistic scenario where they do not know exactly what is coming. However, while using the XCMS parameters the recursive pre-scheduled matching was a reasonable competitor for best method, here it is behind both variants of TopN Exclusion and both variants of Intensity Non-Overlap in coverage.

Additionally, the other versions of the pre-scheduled matching perform only around as well as DsDA, well behind TopN (DsDA is also doing more poorly because of the smaller number of peaks). This is because without a large number of peaks to resolve a schedule for (and with only a relatively small number of higher-quality peaks in the target list) the advantages of the pre-scheduled matching are less prominent. 80% is a relatively high coverage score, but Intensity Non-Overlap reaches higher, at 90%. The matching method fares somewhat better in intensity coverage (beating base TopN Exclusion and competing with the exclusion window variant) but Intensity Non-Overlap is still the best performer.

This drop in matching-based performance also applies to the inclusion window methods. The two Intensity Non-Overlap variants are essentially identical in performance, and while adding inclusion windows is still a significant boost to TopN Exclusion, it no longer allows it to outperform Intensity Non-Overlap in coverage. One interesting thing to note, however, is that the recursive variant of the pre-scheduled two-step matching is above the dotted line, even though this means it is hitting things not included in the target list. This may suggest the width of the isolation window is large enough that it hits targets that MZMine has not managed to align together (possibly in dense areas of the data).

Figure 6.17 also reiterates a lot of the same basic patterns. Intensity Non-Overlap (either variant) remains the most effective method, pre-scheduled methods suffer relative to other methods because of the smaller peak number (though the recursive variant is still effective) and the gain for inclusion windows is also smaller relative to when the data was perfectly known in advance (though TopN Exclusion still benefits substantially). One notable change from Figure 6.16 is that the recursive two-step matching variant outperforms base TopN Exclusion in coverage (as we have seen, TopN Exclusion particularly struggles with this experiment design using multiple sample types). DsDA is also closer to reaching base TopN in performance, likely thanks to the increased number of runs.

Overall, we have seen that the lower number of peaks produced by the MZMine restrictive parameter-set has strong effects on the best choice of fragmentation strategy. While it moves the entire scale of the results up so that in theory only one or two runs per sample type are enough for a complete acquisition, the pre-scheduled matching struggles in an environment where reality differs from its target list. While its performance remains competitive, this underscores the tradeoff being made between DDA and pre-scheduled methods. Additionally, we have confirmed our earlier results from Chapters 4 and 5 by observing that Intensity Non-Overlap is a very effective method (in terms of both coverage and coverage) at lower peak

numbers. Unfortunately, it does not seem that (unlike TopN Exclusion) matching-generated inclusion windows improve it much, if at all, thanks to its more complex behaviour — although it will require more testing to see whether this was due to the base behaviour of Intensity Non-Overlap or due to its combination with SmartRoI for these experiments.

6.5 Timings

One of the factors that might affect the performance of a pre-scheduled method is how much time elapses between the collection of its representative data and the actual acquisition that has been planned. While ideally runs would be conducted directly one after another, processing overhead may increase the gap between these two steps. For this reason, we timed each part of the matching workflow while generating data for our experiments, and present the timings in Table 6.3. These timings were collected on an ordinary desktop PC with an Intel(R) Core(TM) i7-10700 2.90 GHz CPU and 32GB of RAM. Importantly, these timings were not collected in a controlled environment (i.e. other programs were running on the machine and some parts of the matching procedure which we were collecting separate timings for ran in parallel) nor were they averaged across replicates. Consequently, the timings should *only* be interpreted as a rough guide as to how much time the workflow will need, not as precise algorithmic timings.

In general there is no significant difference between XCMS and MZMine (with restrictive parameters) when it comes to processing time for the workflow, despite the very different number of peaks. The longest-running part of the workflow is the "Scan Creation" step, where scans from the representative data are interpolated to give expected intensity values for each scan in the schedule. In the majority of cases it is slower than all other steps combined. Scan creation is also notably slower when being used in the cases where we had a different fullscan per run because it simply had more files to process.

The scan creation subroutine uses ViMMS' implementation of RoI-building, is written in pure Python and is slow as a result of the very large amount of data in .mzMLs there is to process. However, the implementation of this step could be improved in future (for example, XCMS' implementation of RoI-building calls linked C libraries despite the software primarily being written in R). With that being said, this step is actually significantly less of a time bottleneck in our workflow than peak-picking — for example, MZMine had to be left to run over the course of several days to generate all of the peak-picked files we used.

While the two-step matching generally has a runtime twice or thrice as high as the unweighted matching, none of these runtimes exceed 3 minutes. To add context to this number, the scan creation times often exceed ten times this amount and it is not unusual for the length of a single LC-MS/MS metabolomics run to be half an hour. Given that we have seen that the

Same Beer				
	Re-used Fullscans		Unique Fullscans	
	XCMS	MZMine	XCMS	MZMine
Scan Creation	6.5 mins	6.3 mins	32 mins	31.8 mins
Unweighted Matching	17 secs	30 secs	21 secs	29 secs
Two-Step Matching	49 secs	1.3 mins	56 secs	1.3 mins
Recursive Assignment	5.4 mins	13.8 mins	4.9 mins	8.7 mins
Repeated Different Beer				
	Re-used Fullscans		Unique Fullscans	
	XCMS	MZMine	XCMS	MZMine
Scan Creation	24.8 mins	24.7 mins	69.1 mins	68 mins
Unweighted Matching	48 secs	1.1 mins	45 secs	1.1 mins
Two-Step Matching	2.1 mins	3 mins	2.5 mins	2.9 mins
Recursive Assignment	13.8 mins	26.2 mins	11.8 mins	20.1 mins

Table 6.3: A table of times elapsed for different stages of the matching process during the experiments in Sections 6.3 and 6.4 (some stages with negligible runtime have been omitted). Scan creation was run separately for the unweighted and two-step matching so times for those two cases are averaged.

unweighted matching is significantly less effective than the two-step matching, this seems like a small price to pay. Still, if this was too slow, it is possible to apply heuristics like only keeping the n edges with the highest weight for each vertex (potentially at the expense of results quality).

The full assignment step does, on the other hand, consume a significant amount of processing time when using $recursive$ mode, occasionally exceeding the length of scan creation time and reaching nearly half an hour in one case. While the processing time for $nearest$ is negligible, we also saw in our experiments that it barely registered an improvement in coverage and intensity coverage. It may be desirable in future to have a full assignment heuristic which redundantly spreads targets across runs like $recursive$ but which is less processing-intensive.

6.6 Conclusion

In this chapter we have shown a powerful maximum matching based technique for creating LC-MS/MS scan schedules for a given "target list" created from prior knowledge of the acquisition. We have used it both as a standalone "pre-scheduled" method and as an augment to existing Data-Dependent Acquisition methods by using it to generate inclusion windows. While maximum matching has been applied to the problem of LC-MS/MS acquisition con-

trol before [23], this is the first time it has been tested in terms of actually scheduling an acquisition, and we have also demonstrated the necessity of our improvements to the technique to make it practically applicable. We have also highlighted some of the trade-offs made between pure pre-scheduled methods and DDA methods.

Our experiments in Sections 6.3.1 and 6.3.2 showed impressive performance gains for the best fully pre-scheduled method, with it gaining over 20% on both coverage and intensity coverage in a simulated "best-case scenario" environment, theoretically enabling effectively completely comprehensive acquisitions. However, in Section 6.3.3, we saw that in a more realistic case its performance was similar to some of the other best-performing methods like Intensity Non-Overlap and TopN Exclusion augmented with inclusion windows generated by the matching. This illustrates the main tradeoff made between pre-scheduled methods and DDA in that while pre-scheduled methods can create very sophisticated schedules which theoretically lead to highly comprehensive data acquisition, they may have difficulty realising this potential in practice.

In the same experiments we also demonstrated the usefulness of our improvements to the existing matching workflow. In all cases having access to more than one LC-MS/MS run allowed a more comprehensive data acquisition so the benefits of having a multi-run technique (which we introduced in Section 6.1.2) obviously follow. We also showed how multiple sample types could be run as part of the same experiment. The "weighted matching" we introduced in Section 6.1.3 was justified both by the increased intensity coverage (the unweighted matching was often one of the worst performing methods at this metric) and by the fact it generally offered better coverage under the variations in the data we introduced in Section 6.3.3. Finally, the full assignments introduced in Section 6.1.4 was justified by the increase in robustness demonstrated in the same Section 6.3.3. The $recursive$ variant of the matching was the only method to show competitive performance with the other best methods.

For inclusion window generation, we decided to integrate this with our previous work on TopNEXt [27]. This was both because TopNEXt could be easily adapted to incorporate this functionality modularly and because it houses some of the current state-of-the-art in DDA acquisition methods (e.g. Intensity Non-Overlap, SmartRoI). While matching-generated inclusion windows did not seem to provide a significant benefit to Intensity Non-Overlap (which in general we found performed extremely well), they were a significant improvement on another established method, TopN Exclusion, often offering coverage increases of 10%. It should be noted that we generated our inclusion windows by assuming we knew the maximum number of runs in advance (as this is how our pre-scheduled method operates). In future, it may be better to generate inclusion windows greedily per-run to properly leverage the advantages of DDA.

However, this also illustrates one of the major trade-offs between DDA and pre-scheduled methods: DDA requires access to dynamic instrument control. If you cannot control the instrument in real-time then you cannot implement a DDA method by definition. Pre-scheduled methods, conversely, allow schedules to be translated beforehand into whichever format the instrument recognises. This may be one of the reasons why there is relatively little research on new DDA methods outside of frameworks which allow in-silico simulation like ViMMS [26]. While our matching-based method benefits from dynamic instrument control, because it allows it to run fully automatically and hybridisation with DDA requires dynamic control, at its core it is still a pre-scheduled method, so it can be used as-is. However, our experiments show the limitations of this totally offline approach, and we predict as this field develops dynamic algorithms will become more important. Our matching workflow may also integrate with this trend, for example by the use of a dynamic algorithm [157], which would make it more robust and allow it better performance in practice.

Another one of the strengths of our matching workflow is that it can be composed with a completely arbitrary target list of the user's choice. The matching algorithm merely resolves an optimal schedule across this list of targets. We used peak-picking to generate the target list, but it is equally possible to, for example, integrate the matching algorithm's scheduling with a CAMERA-style acquisition workflow [158, 159]. Alternatively, a hypothetical future piece of software could assign a "utility weight" to each edge, expressing the ultimate usefulness of that acquisition as a single numerical estimate. It would then be possible to plan a number of runs to obtain, say, 90% coverage or utility coverage in a completely closed-loop way. More complex problem formulations like stable marriage [160] or a constraint programming formulation [161] may also allow encoding more complex trade-offs between utility values.

The modularity of our method is also important because we have seen that the effectiveness of our workflow depends heavily on the peak-picking. The differences between the experiments in Sections 6.3.1, 6.3.2 and 6.3.3 are essentially a product of different peak-picking assumptions in the evaluation, and as we have seen, the differences are quite large. While conclusions from our evaluation can be supplemented by metabolite annotations from experimental data (and we hope to see such validations in future) this is also potentially a core bottleneck of the acquisition workflow we have presented (in terms of generating spurious targets, improperly aligning data, etc). The modularity of our approach means, however, that future improvements in peak-picking software can be easily integrated into the workflow.

Overall, we have shown a promising new technique for LC-MS/MS acquisition software control which competes with many state-of-the-art methods. This maximum matching based technique has many promising directions for future development, but can also be deployed as-is either using dynamic instrument control or as a completely offline workflow.

Chapter 7

Conclusion

The general theme of this thesis has been to improve the comprehensiveness of LC-MS/MS data acquisition through improving the strategy used by the instrument control software. Our means of doing this has been to improve the comprehensiveness of DDA and pre-scheduled methods by improving their scan scheduling capabilities. This undertaking has produced:

- TopNEXt, an extensible framework for DDA methods. Concretely, TopNEXt builds on existing work on real-time RoI-building [23] to provide two core features. Firstly, it allows the comparison of the current intensity of an RoI to intensities of exclusion windows from previous acquisitions. Secondly, it allows the comparison of the current area of an RoI to the areas of exclusion windows from previous acquisitions. Intensity Non-Overlap is a DDA strategy we introduced to use both these features, and we found it performs especially well.
- An acquisition strategy based on mapping LC-MS/MS data acquisition to an instance of the maximum bipartite matching problem. The existing technique for doing this [23] was extended to work for multiple LC-MS/MS runs and sample types, to optimise the intensity of individual acquisitions and to reallocate leftover scans to improve the robustness of the method. This workflow can be used as a pre-scheduled method which theoretically offers completely comprehensive coverage using a low number of runs, but which currently has performance comparable to state-of-the-art methods. It can also be used to improve existing DDA methods with inclusion windows.

Additionally, we have performed numerous comparative experiments with these methods and other, pre-existing, state-of-the-art methods. The typical means of evaluating such experiments (metabolite annotation) is difficult to apply in simulated experiments, and relies on a "known universe" of metabolite annotations and consequently may not effectively report the usefulness of a given strategy trying to discover heretofore unknown metabolites. We

therefore developed an evaluation strategy using peak-picking, as it is closer to the intrinsic properties of the LC-MS/MS data. The metrics of peak coverage (based on earlier work [23]) and intensity coverage (entirely new) offer a complementary insight into the effectiveness of a fragmentation strategy. All of this work has been contributed to the broader corpus of open-source scientific software via its inclusion in ViMMS.

7.1 Broader Insights

Over the course of investigating this problem, several larger themes have emerged. They have appeared throughout this thesis and often been discussed in some detail (especially towards the end of Chapter 6) but we shall nonetheless re-iterate some here to try and illustrate the bigger picture.

7.1.1 Peak-Picking

The use of peak-picking is a cornerstone of our methodology. The target lists we have constructed as our "ground truth" for evaluation have depended entirely on peak-picking. And while our matching workflow from Chapter 6 can use a target list from any source, all our experiments using the method relied on using peak-picking as input.

Moreso than other factors, we have seen that peak-picking affected the outcome of our experiments. For example, in Chapter 5, we re-examined the experiments in Chapter 4 by testing different sample type orders, different sample type choices, different fragmentation strategy parameters and a completely different dataset and the results did not significantly change. The one variable that did result in a potentially different conclusion was the choice of peak-picking parameters for the evaluation's target list. When the number of peaks identified significantly changed, some of the relative advantages of Intensity Non-Overlap in re-targeting previously targeted areas appeared to recede in favour of TopN Exclusion's more aggressive exclusion.

A similar phenomenon can be observed in Chapter 6, because the utility of the pre-scheduled method depends on its ability to resolve a schedule for a large number of target peaks. At lower peak numbers, simple DDA heuristics are good enough to target most peaks, and their online, real-time nature provides them an advantage in robustness. We also saw the number of peaks for a given sample type increase significantly as more fullscan observations of that sample type were added — this had significant consequences for the results. This is not only difficult to interpret from an evaluation perspective, but it also presents a practical limitation for a matching workflow based on peak-picking. Large variations per LC-MS/MS run may necessitate the acquisition of several fullscan files for planning, which is cumbersome for

the experimenter. Additionally, peak-picking may be orders of magnitude slower than any other part of the matching workflow, and a time lag may increase the chance of unexpected variations. Improvements in peak-picking algorithms (or perhaps custom algorithms which are more focused on a targeting context) may be necessary for deployment in practice.

Overall, there are many difficulties with peak-picking, and it can affect the results quite strongly. Future experimenters following our evaluation protocol should be cautious of the influence it can have, and our conclusions yet require further validation. This should not be interpreted as a disavowal of our peak-based evaluation, however. The objective of our study was to show that new methods could target more prospective targets than ever, and at more opportune times, thus reducing the need for us to be choosy about which peaks to target. An evaluation protocol closer to the intrinsic properties of the data is a useful tool for this purpose. As peak-picking algorithms improve, re-evaluation of our results is possible. And, as we discussed in Section 2.2.5, metabolite annotations come with many difficulties of their own. We therefore view peak-focused evaluations as a useful complement to metabolite annotations (and vice versa).

7.1.2 Intensity Coverage

While we are on the subject of tools for interrogating acquisition strategy performance, we also introduced the intensity coverage concept to complement the concept of (peak) coverage. Intensity coverage essentially averages the maximum intensity each peak was targeted at, taken as a proportion of its whole. Unlike previous attempts at measuring acquisition quality, which have included eyeballing a plot of intensity value vs times targeted [136] and showing how many peaks were acquired above a certain intensity threshold [23], intensity coverage expresses overall acquisition quality as a single numerical value.

Although intensity coverage is ostensibly a direct function of acquisition quality, we found it often split into two factors. Firstly, how broad peak coverage was (if you do not acquire a peak at all, your intensity coverage for it is 0) and secondly how high acquisition intensity was for those peaks you did acquire. While both of these measures going up brings intensity coverage up with them, they are often at odds with one another. If you choose to target every peak exactly once, and at the earliest point possible, you have more scans available to target new peaks, but you will likely have bad average acquisition intensity. Conversely, you could choose to focus all your fragmentation scans on one single peak and never miss the opportunity to target it at its maximum intensity, but you would be severely hampering your peak coverage.

We saw the impact of this in Chapter 5 when we used our “permissive” peak-picking parameter set, greatly increasing the number of peaks. While on a smaller number of peaks

generated by the "restrictive" parameter set Intensity Non-Overlap had been shown to be superior to TopN Exclusion in both peak coverage and intensity coverage, this was not the case with the permissive parameters. While Intensity Non-Overlap remained ahead in intensity coverage, it no longer had an advantage in coverage. However, this also meant that of those peaks which it did target, it was targeting them at a higher intensity on average (and in Chapter 5 we showed this).

Additionally, in Chapter 6 we introduced the idea of adding weights to the maximum matching, to allow acquisition at the highest possible intensity. The original unweighted matching had no preference over which scan to target a peak at, as long as it could acquire the maximum number of peaks. We saw that in theory such a method could simultaneously achieve astronomically high coverage yet have worse intensity coverage than methods well behind it in coverage. And in a more realistic scenario, this method was often targeting away from the centre of the peak, so coverage would also be lost to shifts in the data. The addition of weights to the matching was therefore a necessity in practical terms.

Generally, we saw that most of our new methods tended to have more of an advantage in intensity coverage than they did coverage. As a result we believe that improving the target intensities of individual acquisitions is relatively underexplored compared to improving acquisition coverage. Intensity coverage will hopefully be a useful guideline in the search for better acquisition methods.

7.1.3 Redundancy

Despite being one of the most widely-used methods, TopN is prone to extremely redundant behaviour. Other than hard limits imposed heuristically by DEWs, TopN does not remember its own behaviour and thus is prone to targeting the same high-intensity peaks repeatedly. In addition to this limiting opportunities for coverage, it can be argued that it is low-abundance analytes that most need extra targeting, rather than the most abundant analytes in the sample. Overcoming this weakness of TopN was one of the main motivators of this work.

However, as we alluded to in Section 7.1.2, this is not entirely disadvantageous. If you use every opportunity you have to target an individual peak, you will, by chance, target its apex. Additionally, in a multi-run experiment, even if you would otherwise have a peak be excluded, targeting it again will ensure you don't miss something important.

Of course, even if repeatedly targeting peaks has useful properties, TopN is still biased too heavily towards repeating its actions. We found that Intensity Non-Overlap was overall the best-performing DDA method we studied, and this may be understood in terms of redundancy. While TopN engages in very repetitive behaviour, TopN Exclusion very aggressively

excludes regions it has targeted before. Intensity Non-Overlap takes an approach between these two extremes.

As an example of this, we saw in Chapter 5 that Intensity Non-Overlap would target the same peaks at a greater rate than either TopN Exclusion or Non-Overlap. Intensity Non-Overlap was designed to do this to optimise intensity coverage (by re-targeting peaks at a higher intensity) and by having a less "hard" exclusion criterion than TopN Exclusion by using RoI area. But in addition to the obvious advantage in intensity coverage, we also saw the method had more robust coverage performance when using experiments with multiple sample types in Chapters 4, 5 and 6.

We also saw that TopN Exclusion suffered in performance terms relative to Intensity Non-Overlap when multiple fullscans were used to create target lists in Chapter 6. While this could be addressed by overriding its behaviour with inclusion windows, it does suggest that TopN Exclusion was excluding regions for potential future peaks to appear too aggressively. Intensity Non-Overlap, due to revisiting previously targeted peaks, did not have this problem.

Another thing we saw in Chapter 6 was that although pre-scheduling had a theoretically very high performance ceiling, it struggled heavily when applied to a scenario where reality was different from its plan. This was true for all variants, but the one method which retained performance competitive with methods like Intensity Non-Overlap was the two-step matching with recursive assignment. The recursive assignment step caused it to spread redundant targetings very widely across multiple runs, and this proved very useful to not losing performance in an unexpected scenario.

Overall, while we were initially motivated to engage in less redundant behaviour than TopN, the success of many of our methods may be understood by their taking a middle-ground position between complete redundancy and complete novelty. When designing fragmentation strategies in future, it may be worth remembering that sometimes doing the same thing again can be useful.

7.2 Future Work

The work in this thesis brings new perspective to fragmentation strategies and will hopefully encourage the study of computational problems in mass spectrometry instrument control. But while we have advanced the field, we have (unfortunately!) not completely solved the problem, and so there remains several obvious avenues for future research.

7.2.1 Better Peak-Picking

One of the more obvious limitations of our approach (discussed in Section 7.1.1) is its reliance on peak-picking. Of course, this choice was not arbitrary, and it has strengths complementary to other techniques, but it does create a need for better peak-picking systems. Peak-picking systems will ideally improve in terms of their accuracy in identifying peaks, their performance, and also in how complex it is to parameterise the method.

$asari$ [90] is one of the most exciting developments in this area, as it should greatly simplify the alignment process. This should result in less splitting of peaks across multiple runs. Advances in deep learning peak-recognition [87, 88, 89] may also prove to be faster and more accurate. To reduce parameter complexity, parameter-fitting approaches may be useful. It also may be the case that not all the processing used in peak-picking is necessary to improve *targeting*, so an approach specifically designed for this purpose may perform better.

7.2.2 Probabilistic Reasoning

One of the defining features of this work is that we have focused entirely on heuristics and exact algorithms. However, there is a lot of uncertainty in planning an LC-MS/MS acquisition because of how variable the data is. It may therefore be useful to build on this foundation with more probabilistic reasoning.

For example, a research project that did not make it to the final version of this thesis was to learn retention time drift in real-time using a Gaussian process regression model. Retention time correction is also used to guide acquisitions by systems like MaxQuantLive [68]. This information could be used by a DDA method to adjust its exclusion and inclusion windows to the true acquisition time. A pre-scheduled method like the matching workflow would also be able to recalculate its schedule based on more accurate retention times.

Another possible avenue might to be to predict whether or not a peak is "real" (i.e. based on an underlying analyte) or not in real time and add this information to a TopNEXt scoring function. This is a harder problem than offline peak-picking because the peak has only been partially observed, but a deep learning approach to peak-picking may be especially suited to tackling this problem. Even if from-scratch offline deep learning peak-picking is inferior to established techniques based on CWT, it may be possible for a deep learning system to learn to emulate a CWT system's behaviour with less information (i.e. a partially observed peak). Alternatively, it may be possible to use fast CWT algorithms [76] in real-time.

Another element that could be included in scoring functions could be predictions of how soon an acquisition needs to be in order to not miss the apex of the peak aka "urgency". This may be simpler than identifying whether or not a peak is real, and would allow considering

the trade-off between whether it is better to defer a peak and risk missing its apex or target it now and risk being too early.

7.2.3 More Specialised Knowledge

We have generally focused on this problem purely in the terms of a Computing Science allocation problem, and have incorporated little specialised knowledge. For example, while de-isotoping was included in our peak-picking procedure we did not use it to improve acquisition strategies directly. We mentioned in Chapter 6 that perhaps a CAMERA-style acquisition workflow [158, 159] could be used to improve target lists over raw peak-picking. Another potentially useful approach would be to use a chemical knowledge-base to make the method semi-targeted — it is not uncommon to see this approach in proteomics [83]. And for proteomics, accounting for isotoping and multiple charge-states is also necessary for applications of these methods.

One other interesting idea that we have passed over is the optimisation of acquisition *length*. We have generally assumed that all scans have a scan time distributed around a fixed mean time, but the underlying process is based on how long it takes the instrument to collect sufficient ions. Both the (semi-)targeted methods we have just mentioned [83] and MS2Planner [138] try to optimise acquisition length to obtain sufficient numbers of ions. Future acquisition methods could also take this into account.

7.2.4 Dynamic Instrument Control

In Chapter 6 we addressed at length the trade-off between DDA methods and pre-scheduled methods. Having the ability to change your actions in real-time in response to the observed data is a powerful advantage of DDA. However, it also requires that the DDA has access to the internals of the instrument control software (usually through a non-public API). As a consequence, it is easier to deploy pre-scheduled methods, but their performance suffers as a result of not being able to adapt. We therefore believe more openness in instrument control software will help move the field forwards.

We also identified a specific idea which would allow our pre-scheduled matching method to make the jump from pre-scheduling to DDA. Using a dynamic algorithm [157], sophisticated offline processing could still be done prior to the run, but small updates could be performed in real-time. This would combine some of the advantages of both DDA and pre-scheduling.

7.2.5 Experimental Validation

Due to resource limitations, many of the experiments in this thesis were performed using simulation via ViMMS — for example, all experiments in Chapter 6 are simulated (although based on real experimental data). In some cases this was done specifically to leverage the advantages of simulation: for example, the replication experiments in Chapter 5 used the high throughput of simulated experiments to perform a very large-scale experiment. However, in many other cases there was no instrument available. Even when an instrument was available, experiments in Chapters 4 and Chapter 5 reduced the number of methods tested in the lab. And Chapter 6 used negative ionisation mode because no instrument with working positive ionisation was available. While ViMMS has been repeatedly shown to produce realistic results [26, 137, 25, 27] it is still necessary for there to be additional experimental validation in the lab. This would also allow comparison by metabolite annotations, as a complement to our peak-based analysis.

Additionally, while our experiments have some of the broadest and most detailed comparisons in the literature (it is not unusual to simply compare a new method to TopN [138, 136, 123]) there are many comparisons we have had to omit. Notably, due to their different mechanics, we have not examined MS2Planner, any targeted method or any DIA method. We also, for example, did not explore how WeightedDEW or SmartRoI variants might interface with inclusion windows in Chapter 6. Future work may have to investigate the trade-offs of our methods with these methods, too. Of course, in addition to the intrinsic challenge of doing this, simulators like ViMMS may also need to be updated to allow comparisons of this scale. There is little doubt that our experiments could only have been this extensive thanks to easy access to simulation.

7.3 Final Remarks

Analysis by liquid chromatography tandem mass spectrometry works behind the scenes in a vast range of biological applications. Despite that, surprisingly little attention has been given to control software of the instrument. This thesis tries to bring a computing scientist's perspective to this underexplored problem, and hopefully breaks new ground in doing so.

In Chapter 3 we introduced a method of evaluating fragmentation strategies which we hope will complement metabolite annotations. In Chapters 4 and 5 we introduced state-of-the-art DDA methods, and in Chapter 6 we introduced a state-of-the-art pre-scheduled method and showed some of the trade-offs between these two approaches. Both of these approaches had some consistent throughlines, in that they focused on selecting the optimal acquisition time and used their redundant behaviour to improve method robustness. Despite this, much

work remains to be done in improving peak-picking algorithms, incorporating probabilistic and chemical information into fragmentation strategies, utilising dynamic instrument control and in validating existing fragmentation strategies.

But more than anything else, it is my personal hope that this thesis might serve as a useful guideline in designing the fragmentation strategies of the future.

Bibliography

[1] PA Media, "'Woman who can smell Parkinson's' helps scientists develop test," http://web.archive.org/web/20250215214203/https://www.theguardian.com/society/2022/sep/07/woman-who-can-smell-parkinsons-helps-scientists-develop-test, 2022, accessed: 2025-02-23.

[2] E. Sinclair, D. K. Trivedi, D. Sarkar, C. Walton-Doyle, J. Milne, T. Kunath, A. M. Rijs, R. M. A. de Bie, R. Goodacre, M. Silverdale, and P. Barran, "Metabolomics of sebum reveals lipid dysregulation in Parkinson's disease," *Nature Communications*, vol. 12, no. 1, Mar. 2021. [Online]. Available: http://dx.doi.org/10.1038/s41467-021-21669-4

[3] N. Z. Ballin and K. H. Laursen, "To target or not to target? Definitions and nomenclature for targeted versus non-targeted analytical food authentication," *Trends in Food Science & Technology*, vol. 86, p. 537–543, Apr. 2019. [Online]. Available: http://dx.doi.org/10.1016/j.tifs.2018.09.025

[4] X. Wang, Y. Li, L. Chen, and J. Zhou, "Analytical strategies for LC–MS-based untargeted and targeted metabolomics approaches reveal the entomological origins of honey," *Journal of Agricultural and Food Chemistry*, vol. 70, no. 4, p. 1358–1366, Jan. 2022. [Online]. Available: http://dx.doi.org/10.1021/acs.jafc.1c07153

[5] K. Man, C. Chan, H. Tang, N. Dong, F. Capozzi, K. Wong, K. W. H. Kwok, H. M. Chan, and D. K. Mok, "Mass spectrometry-based untargeted metabolomics approach for differentiation of beef of different geographic origins," *Food Chemistry*, vol. 338, p. 127847, Feb. 2021. [Online]. Available: http://dx.doi.org/10.1016/j.foodchem.2020.127847

[6] A. Bouslimani, L. M. Sanchez, N. Garg, and P. C. Dorrestein, "Mass spectrometry of natural products: current, emerging and future technologies," *Natural Product Reports*, vol. 31, no. 6, p. 718, 2014. [Online]. Available: https://doi.org/10.1039/c4np00044g

[7] J. Wolfender, J.-M. Nuzillard, J. J. J. van der Hooft, J.-H. Renault, and S. Bertrand, "Accelerating metabolite identification in natural product research:

Toward an ideal combination of liquid chromatography–high-resolution tandem mass spectrometry and NMR profiling, in silico databases, and chemometrics," *Analytical Chemistry*, vol. 91, no. 1, p. 704–742, Nov. 2018. [Online]. Available: http://dx.doi.org/10.1021/acs.analchem.8b05112

[8] G. Hjörleifsson Eldjárn, A. Ramsay, J. J. J. van der Hooft, K. R. Duncan, S. Soldatou, J. Rousu, R. Daly, J. Wandy, and S. Rogers, "Ranking microbial metabolomic and genomic links in the NPLinker framework using complementary scoring functions," *PLOS Computational Biology*, vol. 17, no. 5, p. e1008920, May 2021. [Online]. Available: http://dx.doi.org/10.1371/journal.pcbi.1008920

[9] M. Macel, N. M. Van Dam, and J. J. B. Keurentjes, "Metabolomics: the chemistry between ecology and genetics," *Molecular Ecology Resources*, vol. 10, no. 4, pp. 583–593, Mar. 2010. [Online]. Available: https://doi.org/10.1111/j.1755-0998.2010.02854.x

[10] O. A. H. Jones, M. L. Maguire, J. L. Griffin, D. A. Dias, D. J. Spurgeon, and C. Svendsen, "Metabolomics and its use in ecology: Metabolomics in ecology," *Austral Ecology*, vol. 38, no. 6, p. 713–720, Jan. 2013. [Online]. Available: http://dx.doi.org/10.1111/aec.12019

[11] M. Kortesniemi, S. Noerman, A. Kårlund, J. Raita, T. Meuronen, V. Koistinen, R. Landberg, and K. Hanhineva, "Nutritional metabolomics: Recent developments and future needs," *Current Opinion in Chemical Biology*, vol. 77, p. 102400, Dec. 2023. [Online]. Available: http://dx.doi.org/10.1016/j.cbpa.2023.102400

[12] A. Scalbert, L. Brennan, O. Fiehn, T. Hankemeier, B. S. Kristal, B. van Ommen, E. Pujos-Guillot, E. Verheij, D. Wishart, and S. Wopereis, "Mass-spectrometry-based metabolomics: limitations and recommendations for future progress with particular focus on nutrition research," *Metabolomics*, vol. 5, no. 4, pp. 435–458, Jun. 2009. [Online]. Available: https://doi.org/10.1007/s11306-009-0168-0

[13] J. Dawidowska, M. Krzyżanowska, M. J. Markuszewski, and M. Kaliszan, "The application of metabolomics in forensic science with focus on forensic toxicology and time-of-death estimation," *Metabolites*, vol. 11, no. 12, p. 801, Nov. 2021. [Online]. Available: http://dx.doi.org/10.3390/metabo11120801

[14] M. Szeremeta, K. Pietrowska, A. Niemcunowicz-Janica, A. Kretowski, and M. Ciborowski, "Applications of metabolomics in forensic toxicology and forensic medicine," *International Journal of Molecular Sciences*, vol. 22, no. 6, p. 3010, Mar. 2021. [Online]. Available: http://dx.doi.org/10.3390/ijms22063010

[15] S. Li, Y. Park, S. Duraisingham, F. H. Strobel, N. Khan, Q. A. Soltow, D. P. Jones, and B. Pulendran, "Predicting network activity from high throughput metabolomics," *PLoS Computational Biology*, vol. 9, no. 7, p. e1003123, Jul. 2013. [Online]. Available: http://dx.doi.org/10.1371/journal.pcbi.1003123

[16] 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. S7, May 2014. [Online]. Available: http://dx.doi.org/10.1186/1471-2105-15-S7-S9

[17] A. C. Schrimpe-Rutledge, S. G. Codreanu, S. D. Sherrod, and J. A. McLean, "Untargeted metabolomics strategies—challenges and emerging directions," *Journal of the American Society for Mass Spectrometry*, vol. 27, no. 12, p. 1897–1905, Sep. 2016. [Online]. Available: http://dx.doi.org/10.1007/s13361-016-1469-y

[18] L. Cui, H. Lu, and Y. H. Lee, "Challenges and emergent solutions for LC-MS/MS based untargeted metabolomics in diseases," *Mass Spectrometry Reviews*, vol. 37, no. 6, p. 772–792, Feb. 2018. [Online]. Available: http://dx.doi.org/10.1002/mas.21562

[19] B. Buchanan, E. Feigenbaum, and J. Lederberg, "Heuristic DENDRAL: A program for generating explanatory hypotheses in organic chemistry," *Machine Intelligence*, vol. 4, pp. 209–254, 1969.

[20] T. M. Ebbels, J. J. van der Hooft, H. Chatelaine, C. Broeckling, N. Zamboni, S. Hassoun, and E. A. Mathé, "Recent advances in mass spectrometry-based computational metabolomics," *Current Opinion in Chemical Biology*, vol. 74, p. 102288, Jun. 2023. [Online]. Available: http://dx.doi.org/10.1016/j.cbpa.2023.102288

[21] J. Guo, H. Yu, S. Xing, and T. Huan, "Addressing big data challenges in mass spectrometry-based metabolomics," *Chemical Communications*, vol. 58, no. 72, p. 9979–9990, 2022. [Online]. Available: http://dx.doi.org/10.1039/D2CC03598G

[22] L. M. Petrick and N. Shomron, "AI/ML-driven advances in untargeted metabolomics and exposomics for biomedical applications," *Cell Reports Physical Science*, vol. 3, no. 7, p. 100978, Jul. 2022. [Online]. Available: http://dx.doi.org/10.1016/j.xcrp.2022.100978

[23] V. Davies, J. Wandy, S. Weidt, J. J. J. van der Hooft, A. Miller, R. Daly, and S. Rogers, "Rapid development of improved data-dependent acquisition strategies," *Analytical Chemistry*, vol. 93, no. 14, pp. 5676–5683, 2021, pMID: 33784814. [Online]. Available: https://doi.org/10.1021/acs.analchem.0c03895

[24] J. Guo and T. Huan, "Comparison of full-scan, data-dependent, and data-independent acquisition modes in liquid chromatography–mass spectrometry based untargeted metabolomics," *Analytical Chemistry*, vol. 92, no. 12, pp. 8072–8080, May 2020. [Online]. Available: https://doi.org/10.1021/acs.analchem.9b05135

[25] J. Wandy, R. McBride, S. Rogers, N. Terzis, S. Weidt, J. J. J. van der Hooft, K. Bryson, R. Daly, and V. Davies, "Simulated-to-real benchmarking of acquisition methods in untargeted metabolomics," *Frontiers in Molecular Biosciences*, vol. 10, Mar. 2023. [Online]. Available: https://doi.org/10.3389/fmolb.2023.1130781

[26] J. Wandy, V. Davies, R. McBride, S. Weidt, S. Rogers, and R. Daly, "ViMMS 2.0: A framework to develop, test and optimise fragmentation strategies in LC-MS metabolomics," *Journal of Open Source Software*, vol. 7, no. 71, p. 3990, Mar. 2022. [Online]. Available: https://doi.org/10.21105/joss.03990

[27] R. McBride, J. Wandy, S. Weidt, S. Rogers, V. Davies, R. Daly, and K. Bryson, "TopNEXt: Automatic DDA exclusion framework for multi-sample mass spectrometry experiments," *Bioinformatics*, vol. 39, no. 7, Jun. 2023. [Online]. Available: http://dx.doi.org/10.1093/bioinformatics/btad406

[28] Y. Chen, E. Li, and L. Xu, "Guide to metabolomics analysis: A bioinformatics workflow," *Metabolites*, vol. 12, no. 4, p. 357, Apr. 2022. [Online]. Available: http://dx.doi.org/10.3390/metabo12040357

[29] D. J. Creek, W. B. Dunn, O. Fiehn, J. L. Griffin, R. D. Hall, Z. Lei, R. Mistrik, S. Neumann, E. L. Schymanski, L. W. Sumner, R. Trengove, and J. Wolfender, "Metabolite identification: are you sure? And how do your peers gauge your confidence?" *Metabolomics*, vol. 10, no. 3, p. 350–353, Apr. 2014. [Online]. Available: http://dx.doi.org/10.1007/s11306-014-0656-8

[30] L. W. Sumner, A. Amberg, D. Barrett, M. H. Beale, R. Beger, C. A. Daykin, T. Fan, O. Fiehn, R. Goodacre, J. L. Griffin, T. Hankemeier, N. Hardy, J. Harnly, R. Higashi, J. Kopka, A. N. Lane, J. C. Lindon, P. Marriott, A. W. Nicholls, M. D. Reily, J. J. Thaden, and M. R. Viant, "Proposed minimum reporting standards for chemical analysis: Chemical analysis working group (CAWG) metabolomics standards initiative (MSI)," *Metabolomics*, vol. 3, no. 3, p. 211–221, Sep. 2007. [Online]. Available: http://dx.doi.org/10.1007/s11306-007-0082-2

[31] P. A. Hoskisson and R. F. Seipke, "Cryptic or silent? the known unknowns, unknown knowns, and unknown unknowns of secondary metabolism," *mBio*, vol. 11, no. 5, Oct. 2020. [Online]. Available: http://dx.doi.org/10.1128/mBio.02642-20

[32] G. Hjorleifsson Eldjarn, "Ranking microbial metabolomic and genomic links using complementary scoring functions," Ph.D. dissertation, University of Glasgow, 2021.

[33] V. de Lorenzo, "From the selfish gene to selfish metabolism: Revisiting the central dogma," *BioEssays*, vol. 36, no. 3, p. 226–235, Jan. 2014. [Online]. Available: http://dx.doi.org/10.1002/bies.201300153

[34] A. Elofsson, "Progress at protein structure prediction, as seen in CASP15," *Current Opinion in Structural Biology*, vol. 80, p. 102594, Jun. 2023. [Online]. Available: http://dx.doi.org/10.1016/j.sbi.2023.102594

[35] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, and D. Hassabis, "Highly accurate protein structure prediction with AlphaFold," *Nature*, vol. 596, no. 7873, p. 583–589, Jul. 2021. [Online]. Available: http://dx.doi.org/10.1038/s41586-021-03819-2

[36] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter, *Molecular Biology of the Cell*, 4th ed. Garland Science, 2002.

[37] S. Ramazi and J. Zahiri, "Post-translational modifications in proteins: resources, tools and prediction methods," *Database*, vol. 2021, Jan. 2021. [Online]. Available: http://dx.doi.org/10.1093/database/baab012

[38] M. A. Skinnider, C. W. Johnston, M. Gunabalasingam, N. J. Merwin, A. M. Kieliszek, R. J. MacLellan, H. Li, M. R. M. Ranieri, A. L. H. Webster, M. P. T. Cao, A. Pfeifle, N. Spencer, Q. H. To, D. P. Wallace, C. A. Dejong, and N. A. Magarvey, "Comprehensive prediction of secondary metabolite structure and biological activity from microbial genome sequences," *Nature Communications*, vol. 11, no. 1, Nov. 2020. [Online]. Available: http://dx.doi.org/10.1038/s41467-020-19986-1

[39] K. Blin, S. Shaw, K. Steinke, R. Villebro, N. Ziemert, S. Y. Lee, M. H. Medema, and T. Weber, "antiSMASH 5.0: updates to the secondary metabolite genome mining pipeline," *Nucleic Acids Research*, vol. 47, no. W1, p. W81–W87, Apr. 2019. [Online]. Available: http://dx.doi.org/10.1093/nar/gkz310

[40] O. Fiehn, "Metabolomics – the link between genotypes and phenotypes," *Plant Molecular Biology*, vol. 48, no. 1/2, p. 155–171, 2002. [Online]. Available: http://dx.doi.org/10.1023/A:1013713905833

[41] M. Tzafetas, A. Mitra, M. Paraskevaidi, Z. Bodai, I. Kalliala, S. Bowden, K. Lathouras, F. Rosini, M. Szasz, A. Savage, E. Manoli, J. Balog, J. McKenzie, D. Lyons, P. Bennett, D. MacIntyre, S. Ghaem-Maghami, Z. Takats, and M. Kyrgiou, "The intelligent knife (iknife) and its intraoperative diagnostic advantage for the treatment of cervical disease," *Proceedings of the National Academy of Sciences*, vol. 117, no. 13, p. 7338–7346, Mar. 2020. [Online]. Available: http://dx.doi.org/10.1073/pnas.1916960117

[42] E. Humer, C. Pieh, and T. Probst, "Metabolomic biomarkers in anxiety disorders," *International Journal of Molecular Sciences*, vol. 21, no. 13, p. 4784, Jul. 2020. [Online]. Available: http://dx.doi.org/10.3390/ijms21134784

[43] Y. Zhu, S. C. Jha, K. H. Shutta, T. Huang, R. Balasubramanian, C. B. Clish, S. E. Hankinson, and L. D. Kubzansky, "Psychological distress and metabolomic markers: A systematic review of posttraumatic stress disorder, anxiety, and subclinical distress," *Neuroscience & Biobehavioral Reviews*, vol. 143, p. 104954, Dec. 2022. [Online]. Available: http://dx.doi.org/10.1016/j.neubiorev.2022.104954

[44] M. Y. Hirai, M. Yano, D. B. Goodenowe, S. Kanaya, T. Kimura, M. Awazuhara, M. Arita, T. Fujiwara, and K. Saito, "Integration of transcriptomics and metabolomics for understanding of global responses to nutritional stresses in arabidopsis thaliana," *Proceedings of the National Academy of Sciences*, vol. 101, no. 27, p. 10205–10210, Jun. 2004. [Online]. Available: http://dx.doi.org/10.1073/pnas.0403218101

[45] M. P. Papageorgiou, D. Theodoridou, M. Nussbaumer, M. Syrrou, and M. D. Filiou, "Deciphering the metabolome under stress: Insights from rodent models," *Current Neuropharmacology*, vol. 22, no. 5, p. 884–903, May 2024. [Online]. Available: http://dx.doi.org/10.2174/1570159X21666230713094843

[46] D. Schranner, G. Kastenmüller, M. Schönfelder, W. Römisch-Margl, and H. Wackerhage, "Metabolite concentration changes in humans after a bout of exercise: a systematic review of exercise metabolomics studies," *Sports Medicine - Open*, vol. 6, no. 1, Feb. 2020. [Online]. Available: http://dx.doi.org/10.1186/s40798-020-0238-4

[47] D. B. Kell and S. G. Oliver, "The metabolome 18 years on: a concept comes of age," *Metabolomics*, vol. 12, no. 9, Sep. 2016. [Online]. Available: http://dx.doi.org/10.1007/s11306-016-1108-4

[48] F. Wang, J. Liigand, S. Tian, D. Arndt, R. Greiner, and D. S. Wishart, "CFM-ID 4.0: More accurate ESI-MS/MS spectral prediction and compound identification," *Analytical Chemistry*, vol. 93, no. 34, p. 11692–11700, Aug. 2021. [Online]. Available: http://dx.doi.org/10.1021/acs.analchem.1c01465

[49] T. Kind, K. Liu, D. Y. Lee, B. DeFelice, J. K. Meissen, and O. Fiehn, "LipidBlast in silico tandem mass spectrometry database for lipid identification," *Nature Methods*, vol. 10, no. 8, p. 755–758, Jun. 2013. [Online]. Available: http://dx.doi.org/10.1038/nmeth.2551

[50] C. P. Wild, "Complementing the genome with an "exposome": The outstanding challenge of environmental exposure measurement in molecular epidemiology," *Cancer Epidemiology, Biomarkers & Prevention*, vol. 14, no. 8, p. 1847–1850, Aug. 2005. [Online]. Available: http://dx.doi.org/10.1158/1055-9965.EPI-05-0456

[51] M. H. Medema, Y. Paalvast, D. D. Nguyen, A. Melnik, P. C. Dorrestein, E. Takano, and R. Breitling, "Pep2Path: Automated mass spectrometry-guided genome mining of peptidic natural products," *PLoS Computational Biology*, vol. 10, no. 9, p. e1003822, Sep. 2014. [Online]. Available: http://dx.doi.org/10.1371/journal.pcbi.1003822

[52] A. Amoresano and P. Pucci, *Mass spectrometry in metabolomics*. Elsevier, 2022, p. 109–147. [Online]. Available: http://dx.doi.org/10.1016/B978-0-323-85062-9.00004-0

[53] F. W. McLafferty, "A century of progress in molecular mass spectrometry," *Annual Review of Analytical Chemistry*, vol. 4, no. 1, p. 1–22, Jul. 2011. [Online]. Available: http://dx.doi.org/10.1146/annurev-anchem-061010-114018

[54] C. Dass, *Fundamentals of Contemporary Mass Spectrometry*. Wiley, Aug. 2006. [Online]. Available: http://dx.doi.org/10.1002/9780470118498.ch5

[55] D. Martins-de Souza, "Proteomics, metabolomics, and protein interactomics in the characterization of the molecular features of major depressive disorder," *Dialogues in Clinical Neuroscience*, vol. 16, no. 1, p. 63–73, Mar. 2014. [Online]. Available: http://dx.doi.org/10.31887/DCNS.2014.16.1/dmartins

[56] F. Pullen, "The fascinating history of the development of LC-MS; a personal perspective," https://gala.gre.ac.uk/id/eprint/6875/, 2010, accessed: 2024-09-08.

[57] J. Gale, M. ElNaggar, M. Grayson, D. Sparkman, A. Yergey, and K. Tomer, "Liquid chromatography and mass spectrometry: From impossible union to a match made in heaven," https://www.asms.org/docs/default-source/history-posters/tech_asms-lcms-poster_2018_final-(1).pdf?sfvrsn=ef3476c3_0, 2018, accessed: 2024-09-15.

[58] Nobel Foundation, "2002 Nobel Prize press release," https://www.nobelprize.org/prizes/chemistry/2002/press-release/, 2002, accessed: 2024-09-05.

[59] J. B. Fenn, M. Mann, C. K. Meng, S. F. Wong, and C. M. Whitehouse, "Electrospray ionization for mass spectrometry of large biomolecules," *Science*, vol. 246, no. 4926, p. 64–71, Oct. 1989. [Online]. Available: http://dx.doi.org/10.1126/science.2675315

[60] M. A. Grayson, "Origins of tandem mass spectrometry," https://www.asms.org/docs/history-posters/tandem-ms-poster-2012.pdf, 2012, accessed: 2024-09-09.

[61] B. A. Thomson, "Atmospheric pressure ionization and liquid chromatography/mass spectrometry—together at last," *Journal of the American Society for Mass Spectrometry*, vol. 9, no. 3, p. 187–193, Mar. 1998. [Online]. Available: http://dx.doi.org/10.1016/S1044-0305(97)00285-7

[62] M. J. Rardin, "Rapid assessment of contaminants and interferences in mass spectrometry data using Skyline," *Journal of the American Society for Mass Spectrometry*, vol. 29, no. 6, p. 1327–1330, Apr. 2018. [Online]. Available: http://dx.doi.org/10.1007/s13361-018-1940-z

[63] T. Cajka, J. Hricko, L. Rudl Kulhava, M. Paucova, M. Novakova, O. Fiehn, and O. Kuda, "Exploring the impact of organic solvent quality and unusual adduct formation during LC-MS-based lipidomic profiling," *Metabolites*, vol. 13, no. 9, p. 966, Aug. 2023. [Online]. Available: http://dx.doi.org/10.3390/metabo13090966

[64] Z. Lai, H. Tsugawa, G. Wohlgemuth, S. Mehta, M. Mueller, Y. Zheng, A. Ogiwara, J. Meissen, M. Showalter, K. Takeuchi, T. Kind, P. Beal, M. Arita, and O. Fiehn, "Identifying metabolites by integrating metabolome databases with mass spectrometry cheminformatics," *Nature Methods*, vol. 15, no. 1, p. 53–56, Nov. 2017. [Online]. Available: http://dx.doi.org/10.1038/nmeth.4512

[65] N. Hoffmann, M. Keck, H. Neuweger, M. Wilhelm, P. Högy, K. Niehaus, and J. Stoye, "Combining peak- and chromatogram-based retention time alignment algorithms for multiple chromatography-mass spectrometry datasets," *BMC Bioinformatics*, vol. 13, no. 1, Aug. 2012. [Online]. Available: http://dx.doi.org/10.1186/1471-2105-13-214

[66] R. Wei, J. Wang, M. Su, E. Jia, S. Chen, T. Chen, and Y. Ni, "Missing value imputation approach for mass spectrometry-based metabolomics data," *Scientific Reports*, vol. 8, no. 1, Jan. 2018. [Online]. Available: http://dx.doi.org/10.1038/s41598-017-19120-0

[67] S. Heiles, "Advanced tandem mass spectrometry in metabolomics and lipidomics—methods and applications," *Analytical and Bioanalytical Chemistry*, vol. 413, no. 24, p. 5927–5948, Jun. 2021. [Online]. Available: http://dx.doi.org/10.1007/s00216-021-03425-1

[68] C. Wichmann, F. Meier, S. Virreira Winter, A.-D. Brunner, J. Cox, and M. Mann, "MaxQuant.Live enables global targeting of more than 25, 000 peptides," *Molecular & Cellular Proteomics*, vol. 18, no. 5, p. 982a–9994, May 2019. [Online]. Available: http://dx.doi.org/10.1074/mcp.TIR118.001131

[69] M. W. Senko, S. C. Beu, and F. W. McLaffertycor, "Determination of monoisotopic masses and ion populations for large biomolecules from resolved isotopic distributions," *Journal of the American Society for Mass Spectrometry*, vol. 6, no. 4, p. 229–233, Apr. 1995. [Online]. Available: http://dx.doi.org/10.1016/1044-0305(95)00017-8

[70] M. R. Blumer, C. H. Chang, E. Brayfindley, J. R. Nunez, S. M. Colby, R. S. Renslow, and T. O. Metz, "Mass spectrometry adduct calculator," *Journal of Chemical Information and Modeling*, vol. 61, no. 12, p. 5721–5725, Nov. 2021. [Online]. Available: http://dx.doi.org/10.1021/acs.jcim.1c00579

[71] S. Alseekh, A. Aharoni, Y. Brotman, K. Contrepois, J. D'Auria, J. Ewald, J. C. Ewald, P. D. Fraser, P. Giavalisco, R. D. Hall, M. Heinemann, H. Link, J. Luo, S. Neumann, J. Nielsen, L. Perez de Souza, K. Saito, U. Sauer, F. C. Schroeder, S. Schuster, G. Siuzdak, A. Skirycz, L. W. Sumner, M. P. Snyder, H. Tang, T. Tohge, Y. Wang, W. Wen, S. Wu, G. Xu, N. Zamboni, and A. R. Fernie, "Mass spectrometry-based metabolomics: a guide for annotation, quantification and best reporting practices," *Nature Methods*, vol. 18, no. 7, p. 747–756, Jul. 2021. [Online]. Available: http://dx.doi.org/10.1038/s41592-021-01197-1

[72] S. Böcker, M. C. Letzel, Z. Lipták, and A. Pervukhin, "SIRIUS: decomposing isotope patterns for metabolite identification," *Bioinformatics*, vol. 25, no. 2, p. 218–224, Nov. 2008. [Online]. Available: http://dx.doi.org/10.1093/bioinformatics/btn603

[73] S. Böcker, "Algorithmic mass spectrometry," https://bio.informatik.uni-jena.de/textbook-algoms/, p. 210–212, 2022, accessed: 2024-09-14.

[74] C. A. Smith, E. J. Want, G. O'Maille, R. Abagyan, and G. Siuzdak, "XCMS: processing mass spectrometry data for metabolite profiling using nonlinear peak alignment, matching, and identification," *Analytical Chemistry*, vol. 78, no. 3, pp. 779–787, Feb. 2006. [Online]. Available: https://doi.org/10.1021/ac051437y

[75] 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, p. 395, Jul. 2010.

[76] L. P. A. Arts and E. L. van den Broek, "The fast continuous wavelet transformation (fCWT) for real-time, high-quality, noise-resistant time–frequency analysis," *Nature Computational Science*, vol. 2, no. 1, p. 47–58, Jan. 2022. [Online]. Available: http://dx.doi.org/10.1038/s43588-021-00183-z

[77] R. Tautenhahn, C. Böttcher, and S. Neumann, "Highly sensitive feature detection for high resolution LC/MS," *BMC Bioinformatics*, vol. 9, 2008. [Online]. Available: https://doi.org/10.1186/1471-2105-9-504

[78] O. D. Myers, S. J. Sumner, S. Li, S. Barnes, and X. Du, "One step forward for reducing false positive and false negative compound identifications from mass spectrometry metabolomics data: New algorithms for constructing extracted ion chromatograms and detecting chromatographic peaks," *Analytical Chemistry*, vol. 89, no. 17, pp. 8696–8703, 2017, pMID: 28752754. [Online]. Available: https://doi.org/10.1021/acs.analchem.7b00947

[79] X. Du, A. Smirnov, T. Pluskal, W. Jia, and S. Sumner, *Metabolomics Data Preprocessing Using ADAP and MZmine 2*. Springer US, 2020, p. 25–48. [Online]. Available: http://dx.doi.org/10.1007/978-1-0716-0239-3_3

[80] J. Wandy, "Unsupervised bayesian explorations of mass spectrometry data," Ph.D. dissertation, University of Glasgow, 2017.

[81] G. Skoraczyński, A. Gambin, and B. Miasojedow, "Alignstein: Optimal transport for improved LC-MS retention time alignment," *GigaScience*, vol. 11, 2022. [Online]. Available: http://dx.doi.org/10.1093/gigascience/giac101

[82] Y. Liu, Y. Yang, W. Chen, F. Shen, L. Xie, Y. Zhang, Y. Zhai, F. He, Y. Zhu, and C. Chang, "DeepRTAlign: toward accurate retention time alignment for large cohort mass spectrometry data analysis," *Nature Communications*, vol. 14, no. 1, Dec. 2023. [Online]. Available: http://dx.doi.org/10.1038/s41467-023-43909-5

[83] M. van Bentum and M. Selbach, "An introduction to advanced targeted acquisition methods," *Molecular & Cellular Proteomics*, vol. 20, p. 100165, 2021. [Online]. Available: http://dx.doi.org/10.1016/j.mcpro.2021.100165

[84] C. Smith, R. Tautenhahn, S. Neumann, P. Benton, C. Conley, J. Rainer, M. Witting, W. Kumler, P. Louail, P. Vangeenderhuysen, and C. Brunius, "XCMS reference manual," https://bioconductor.org/packages/devel/bioc/manuals/xcms/man/xcms.pdf, 2024, accessed: 2024-08-03.

[85] E. Müller, C. E. Huber, W. Brack, M. Krauss, and T. Schulze, "Symbolic aggregate approximation improves gap filling in high-resolution mass spectrometry

data processing," *Analytical Chemistry*, vol. 92, no. 15, p. 10425–10432, Jul. 2020. [Online]. Available: http://dx.doi.org/10.1021/acs.analchem.0c00899

[86] H. Tsugawa, T. Cajka, T. Kind, Y. Ma, B. Higgins, K. Ikeda, M. Kanazawa, J. VanderGheynst, O. Fiehn, and M. Arita, "MS-DIAL: data-independent MS/MS deconvolution for comprehensive metabolome analysis," *Nature methods*, vol. 12, no. 6, pp. 523–526, 2015.

[87] A. D. Melnikov, Y. P. Tsentalovich, and V. V. Yanshole, "Deep learning for the precise peak detection in high-resolution LC–MS data," *Analytical Chemistry*, vol. 92, no. 1, p. 588–592, Dec. 2019. [Online]. Available: http://dx.doi.org/10.1021/acs.analchem.9b04811

[88] Y. Gloaguen, J. A. Kirwan, and D. Beule, "Deep learning-assisted peak curation for large-scale LC-MS metabolomics," *Analytical Chemistry*, vol. 94, no. 12, p. 4930–4937, Mar. 2022. [Online]. Available: http://dx.doi.org/10.1021/acs.analchem.1c02220

[89] E. Stancliffe and G. J. Patti, "PeakDetective: A semisupervised deep learning-based approach for peak curation in untargeted metabolomics," *Analytical Chemistry*, vol. 95, no. 25, p. 9397–9403, Jun. 2023. [Online]. Available: http://dx.doi.org/10.1021/acs.analchem.3c00764

[90] S. Li, A. Siddiqa, M. Thapa, Y. Chi, and S. Zheng, "Trackable and scalable LC-MS metabolomics data processing using asari," *Nature Communications*, vol. 14, no. 1, Jul. 2023. [Online]. Available: http://dx.doi.org/10.1038/s41467-023-39889-1

[91] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach Global Edition*, 3rd ed. Pearson, 2016.

[92] Y. Li, T. Kind, J. Folz, A. Vaniya, S. S. Mehta, and O. Fiehn, "Spectral entropy outperforms MS/MS dot product similarity for small-molecule compound identification," *Nature Methods*, vol. 18, no. 12, p. 1524–1531, Dec. 2021. [Online]. Available: http://dx.doi.org/10.1038/s41592-021-01331-z

[93] F. Huber, L. Ridder, S. Verhoeven, J. H. Spaaks, F. Diblen, S. Rogers, and J. J. J. van der Hooft, "Spec2Vec: Improved mass spectral similarity scoring through learning of structural relationships," *PLOS Computational Biology*, vol. 17, no. 2, p. e1008724, Feb. 2021. [Online]. Available: http://dx.doi.org/10.1371/journal.pcbi.1008724

[94] D. S. Wishart, A. Guo, E. Oler, F. Wang, A. Anjum, H. Peters, R. Dizon, Z. Sayeeda, S. Tian, B. L. Lee, M. Berjanskii, R. Mah, M. Yamamoto, J. Jovel, C. Torres-Calzada, M. Hiebert-Giesbrecht, V. W. Lui, D. Varshavi, D. Varshavi, D. Allen, D. Arndt,

N. Khetarpal, A. Sivakumaran, K. Harford, S. Sanford, K. Yee, X. Cao, Z. Budinski, J. Liigand, L. Zhang, J. Zheng, R. Mandal, N. Karu, M. Dambrova, H. B. Schiöth, R. Greiner, and V. Gautam, "HMDB 5.0: the human metabolome database for 2022," *Nucleic Acids Research*, vol. 50, no. D1, p. D622–D631, 2022. [Online]. Available: http://dx.doi.org/10.1093/nar/gkab1062

[95] H. Horai, M. Arita, S. Kanaya, Y. Nihei, T. Ikeda, K. Suwa, Y. Ojima, K. Tanaka, S. Tanaka, K. Aoshima, Y. Oda, Y. Kakazu, M. Kusano, T. Tohge, F. Matsuda, Y. Sawada, M. Y. Hirai, H. Nakanishi, K. Ikeda, N. Akimoto, T. Maoka, H. Takahashi, T. Ara, N. Sakurai, H. Suzuki, D. Shibata, S. Neumann, T. Iida, K. Tanaka, K. Funatsu, F. Matsuura, T. Soga, R. Taguchi, K. Saito, and T. Nishioka, "MassBank: a public repository for sharing mass spectral data for life sciences," *Journal of Mass Spectrometry*, vol. 45, no. 7, p. 703–714, Jul. 2010. [Online]. Available: http://dx.doi.org/10.1002/jms.1777

[96] K. W. Phinney, G. Ballihaut, M. Bedner, B. S. Benford, J. E. Camara, S. J. Christopher, W. C. Davis, N. G. Dodder, G. Eppe, B. E. Lang, S. E. Long, M. S. Lowenthal, E. A. McGaw, K. E. Murphy, B. C. Nelson, J. L. Prendergast, J. L. Reiner, C. A. Rimmer, L. C. Sander, M. M. Schantz, K. E. Sharpless, L. T. Sniegoski, S. S. Tai, J. B. Thomas, T. W. Vetter, M. J. Welch, S. A. Wise, L. J. Wood, W. F. Guthrie, C. R. Hagwood, S. D. Leigh, J. H. Yen, N.-F. Zhang, M. Chaudhary-Webb, H. Chen, Z. Fazili, D. J. LaVoie, L. F. McCoy, S. S. Momin, N. Paladugula, E. C. Pendergrast, C. M. Pfeiffer, C. D. Powers, D. Rabinowitz, M. E. Rybak, R. L. Schleicher, B. M. H. Toombs, M. Xu, M. Zhang, and A. L. Castle, "Development of a standard reference material for metabolomics research," *Analytical Chemistry*, vol. 85, no. 24, p. 11732–11738, Dec. 2013. [Online]. Available: http://dx.doi.org/10.1021/ac402689t

[97] A. T. Aron, E. C. Gentry, K. L. McPhail, L. Nothias, M. Nothias-Esposito, A. Bouslimani, D. Petras, J. M. Gauglitz, N. Sikora, F. Vargas, J. J. J. van der Hooft, M. Ernst, K. B. Kang, C. M. Aceves, A. M. Caraballo-Rodríguez, I. Koester, K. C. Weldon, S. Bertrand, C. Roullier, K. Sun, R. M. Tehan, C. A. Boya P., M. H. Christian, M. Gutiérrez, A. M. Ulloa, J. A. Tejeda Mora, R. Mojica-Flores, J. Lakey-Beitia, V. Vásquez-Chaves, Y. Zhang, A. I. Calderón, N. Tayler, R. A. Keyzers, F. Tugizimana, N. Ndlovu, A. A. Aksenov, A. K. Jarmusch, R. Schmid, A. W. Truman, N. Bandeira, M. Wang, and P. C. Dorrestein, "Reproducible molecular networking of untargeted mass spectrometry data using GNPS," *Nature Protocols*, vol. 15, no. 6, p. 1954–1991, May 2020. [Online]. Available: http://dx.doi.org/10.1038/s41596-020-0317-5

[98] C. Guijas, J. R. Montenegro-Burke, X. Domingo-Almenara, A. Palermo, B. Warth, G. Hermann, G. Koellensperger, T. Huan, W. Uritboonthai, A. E. Aisporna, D. W. Wolan, M. E. Spilker, H. P. Benton, and G. Siuzdak, "METLIN: A technology platform for identifying knowns and unknowns," *Analytical Chemistry*, vol. 90, no. 5, p. 3156–3164, Jan. 2018. [Online]. Available: http://dx.doi.org/10.1021/acs.analchem.7b04424

[99] D. N. Perkins, D. J. C. Pappin, D. M. Creasy, and J. S. Cottrell, "Probability-based protein identification by searching sequence databases using mass spectrometry data," *Electrophoresis*, vol. 20, no. 18, p. 3551–3567, Dec. 1999. [Online]. Available: http://dx.doi.org/10.1002/(SICI)1522-2683(19991201)20:18⟨3551::AID-ELPS3551⟩3.0.CO;2-2

[100] J. K. Eng, T. A. Jahan, and M. R. Hoopmann, "Comet: An open-source MS/MS sequence database search tool," *PROTEOMICS*, vol. 13, no. 1, p. 22–24, Dec. 2012. [Online]. Available: http://dx.doi.org/10.1002/pmic.201200439

[101] R. Daly, S. Rogers, J. Wandy, A. Jankevics, K. E. V. Burgess, and R. Breitling, "MetAssign: probabilistic annotation of metabolites from LC–MS data using a bayesian clustering approach," *Bioinformatics*, vol. 30, no. 19, p. 2764–2771, Jun. 2014. [Online]. Available: http://dx.doi.org/10.1093/bioinformatics/btu370

[102] N. F. de Jonge, K. Mildau, D. Meijer, J. J. R. Louwen, C. Bueschl, F. Huber, and J. J. J. van der Hooft, "Good practices and recommendations for using and benchmarking computational metabolomics metabolite annotation tools," *Metabolomics*, vol. 18, no. 12, Dec. 2022. [Online]. Available: http://dx.doi.org/10.1007/s11306-022-01963-y

[103] K. Dührkop, H. Shen, M. Meusel, J. Rousu, and S. Böcker, "Searching molecular structure databases with tandem mass spectra using CSI:FingerID," *Proceedings of the National Academy of Sciences*, vol. 112, no. 41, p. 12580–12585, Sep. 2015. [Online]. Available: http://dx.doi.org/10.1073/pnas.1509788112

[104] M. A. Hoffmann, L. Nothias, M. Ludwig, M. Fleischauer, E. C. Gentry, M. Witting, P. C. Dorrestein, K. Dührkop, and S. Böcker, "High-confidence structural annotation of metabolites absent from spectral libraries," *Nature Biotechnology*, vol. 40, no. 3, p. 411–421, Oct. 2021. [Online]. Available: http://dx.doi.org/10.1038/s41587-021-01045-9

[105] K. Scheubert, F. Hufsky, D. Petras, M. Wang, L. Nothias, K. Dührkop, N. Bandeira, P. C. Dorrestein, and S. Böcker, "Significance estimation for large scale metabolomics

annotations by spectral matching," *Nature Communications*, vol. 8, no. 1, Nov. 2017. [Online]. Available: http://dx.doi.org/10.1038/s41467-017-01318-5

[106] B. C. DeFelice, S. S. Mehta, S. Samra, T. Čajka, B. Wancewicz, J. F. Fahrmann, and O. Fiehn, "Mass spectral feature list optimizer (MS-FLO): A tool to minimize false positive peak reports in untargeted liquid chromatography–mass spectroscopy (LC-MS) data processing," *Analytical Chemistry*, vol. 89, no. 6, p. 3250–3255, Mar. 2017. [Online]. Available: http://dx.doi.org/10.1021/acs.analchem.6b04372

[107] J. Watrous, P. Roach, T. Alexandrov, B. S. Heath, J. Y. Yang, R. D. Kersten, M. van der Voort, K. Pogliano, H. Gross, J. M. Raaijmakers, B. S. Moore, J. Laskin, N. Bandeira, and P. C. Dorrestein, "Mass spectral molecular networking of living microbial colonies," *Proceedings of the National Academy of Sciences*, vol. 109, no. 26, May 2012. [Online]. Available: http://dx.doi.org/10.1073/pnas.1203689109

[108] L. Nothias, D. Petras, R. Schmid, K. Dührkop, J. Rainer, A. Sarvepalli, I. Protsyuk, M. Ernst, H. Tsugawa, M. Fleischauer, F. Aicheler, A. A. Aksenov, O. Alka, P.-M. Allard, A. Barsch, X. Cachet, A. M. Caraballo-Rodriguez, R. R. Da Silva, T. Dang, N. Garg, J. M. Gauglitz, A. Gurevich, G. Isaac, A. K. Jarmusch, Z. Kameník, K. B. Kang, N. Kessler, I. Koester, A. Korf, A. Le Gouellec, M. Ludwig, C. Martin H., L.-I. McCall, J. McSayles, S. W. Meyer, H. Mohimani, M. Morsy, O. Moyne, S. Neumann, H. Neuweger, N. H. Nguyen, M. Nothias-Esposito, J. Paolini, V. V. Phelan, T. Pluskal, R. A. Quinn, S. Rogers, B. Shrestha, A. Tripathi, J. J. J. van der Hooft, F. Vargas, K. C. Weldon, M. Witting, H. Yang, Z. Zhang, F. Zubeil, O. Kohlbacher, S. Böcker, T. Alexandrov, N. Bandeira, M. Wang, and P. C. Dorrestein, "Feature-based molecular networking in the GNPS analysis environment," *Nature Methods*, vol. 17, no. 9, p. 905–908, Aug. 2020. [Online]. Available: http://dx.doi.org/10.1038/s41592-020-0933-6

[109] J. J. J. van der Hooft, J. Wandy, M. P. Barrett, K. E. V. Burgess, and S. Rogers, "Topic modeling for untargeted substructure exploration in metabolomics," *Proceedings of the National Academy of Sciences*, vol. 113, no. 48, p. 13738–13743, Nov. 2016. [Online]. Available: http://dx.doi.org/10.1073/pnas.1608041113

[110] A. D. Shrivastava, N. Swainston, S. Samanta, I. Roberts, M. Wright Muelas, and D. B. Kell, "MassGenie: A transformer-based deep learning method for identifying small molecules from their mass spectra," *Biomolecules*, vol. 11, no. 12, p. 1793, Nov. 2021. [Online]. Available: http://dx.doi.org/10.3390/biom11121793

[111] C. Ruttkies, E. L. Schymanski, S. Wolf, J. Hollender, and S. Neumann, "Metfrag relaunched: incorporating strategies beyond in silico fragmentation,"

Journal of Cheminformatics, vol. 8, no. 1, Jan. 2016. [Online]. Available: http://dx.doi.org/10.1186/s13321-016-0115-9

[112] K. Dührkop, M. Fleischauer, M. Ludwig, A. A. Aksenov, A. V. Melnik, M. Meusel, P. C. Dorrestein, J. Rousu, and S. Böcker, "SIRIUS 4: a rapid tool for turning tandem mass spectra into metabolite structure information," *Nature Methods*, vol. 16, no. 4, p. 299–302, Mar. 2019. [Online]. Available: http://dx.doi.org/10.1038/s41592-019-0344-8

[113] Y. Djoumbou-Feunang, J. Fiamoncini, A. Gil-de-la Fuente, R. Greiner, C. Manach, and D. S. Wishart, "Biotransformer: a comprehensive computational tool for small molecule metabolism prediction and metabolite identification," *Journal of Cheminformatics*, vol. 11, no. 1, Jan. 2019. [Online]. Available: http://dx.doi.org/10.1186/s13321-018-0324-5

[114] L. Perez De Souza, S. Alseekh, Y. Brotman, and A. R. Fernie, "Network-based strategies in metabolomics data analysis and interpretation: from molecular networking to biological interpretation," *Expert Review of Proteomics*, vol. 17, no. 4, p. 243–255, Apr. 2020. [Online]. Available: http://dx.doi.org/10.1080/14789450.2020.1766975

[115] F. R. Pinu, D. J. Beale, A. M. Paten, K. Kouremenos, S. Swarup, H. J. Schirra, and D. Wishart, "Systems biology and multi-omics integration: Viewpoints from the metabolomics research community," *Metabolites*, vol. 9, no. 4, p. 76, Apr. 2019. [Online]. Available: http://dx.doi.org/10.3390/metabo9040076

[116] M. Babu and M. Snyder, "Multi-omics profiling for health," *Molecular & Cellular Proteomics*, vol. 22, no. 6, p. 100561, Jun. 2023. [Online]. Available: http://dx.doi.org/10.1016/j.mcpro.2023.100561

[117] Y. J. Heo, C. Hwa, G. Lee, J. Park, and J. An, "Integrative multi-omics approaches in cancer research: From biological networks to clinical subtypes," *Molecules and Cells*, vol. 44, no. 7, p. 433–443, Jul. 2021. [Online]. Available: http://dx.doi.org/10.14348/molcells.2021.0042

[118] O. Ramos-Lopez, J. A. Martinez, and F. I. Milagro, "Holistic integration of omics tools for precision nutrition in health and disease," *Nutrients*, vol. 14, no. 19, p. 4074, Sep. 2022. [Online]. Available: http://dx.doi.org/10.3390/nu14194074

[119] M. Palmblad, "History of computers and computing in mass spectrometry - the early years," https://www.asms.org/docs/default-source/history-posters/05_technology_history-of-computing-in-mass-spectrometry-i.pdf, 2024, accessed: 2024-09-15.

[120] R. A. Hites and K. Biemann, "A computer-compatible digital data acquisition system for fast-scanning, single-focusing mass spectrometers," *Analytical Chemistry*, vol. 39, no. 8, p. 965–970, Jul. 1967. [Online]. Available: http://dx.doi.org/10.1021/ac60252a043

[121] J. R. Yates, "Pivotal role of computers and software in mass spectrometry – SEQUEST and 20 years of tandem MS database searching," *Journal of the American Society for Mass Spectrometry*, vol. 26, no. 11, p. 1804–1813, Aug. 2015. [Online]. Available: http://dx.doi.org/10.1007/s13361-015-1220-0

[122] S. C. Bendall, C. Hughes, J. L. Campbell, M. H. Stewart, P. Pittock, S. Liu, E. Bonneil, P. Thibault, M. Bhatia, and G. A. Lajoie, "An enhanced mass spectrometry approach reveals human embryonic stem cell growth factors in culture," *Molecular & Cellular Proteomics*, vol. 8, no. 3, pp. 421–432, Mar. 2009. [Online]. Available: https://doi.org/10.1074/mcp.m800190-mcp200

[123] J. P. Koelmel, N. M. Kroeger, E. L. Gill, C. Z. Ulmer, J. A. Bowden, R. E. Patterson, R. A. Yost, and T. J. Garrett, "Expanding lipidome coverage using LC-MS/MS data-dependent acquisition with automated exclusion list generation," *Journal of the American Society for Mass Spectrometry*, vol. 28, no. 5, p. 908–917, Mar. 2017. [Online]. Available: http://dx.doi.org/10.1007/s13361-017-1608-0

[124] S. Kreimer, M. E. Belov, W. F. Danielson, L. I. Levitsky, M. V. Gorshkov, B. L. Karger, and A. R. Ivanov, "Advanced precursor ion selection algorithms for increased depth of bottom-up proteomic profiling," *Journal of Proteome Research*, vol. 15, no. 10, p. 3563–3573, Sep. 2016. [Online]. Available: http://dx.doi.org/10.1021/acs.jproteome.6b00312

[125] T. Geiger, J. Cox, and M. Mann, "Proteomics on an orbitrap benchtop mass spectrometer using all-ion fragmentation," *Molecular & Cellular Proteomics*, vol. 9, no. 10, p. 2252–2261, Oct. 2010. [Online]. Available: http://dx.doi.org/10.1074/mcp.M110.001537

[126] L. C. Gillet, P. Navarro, S. Tate, H. Röst, N. Selevsek, L. Reiter, R. Bonner, and R. Aebersold, "Targeted data extraction of the MS/MS spectra generated by data-independent acquisition: A new concept for consistent and accurate proteome analysis," *Molecular & Cellular Proteomics*, vol. 11, no. 6, p. O111.016717, Jun. 2012. [Online]. Available: http://dx.doi.org/10.1074/mcp.O111.016717

[127] C. Tsou, D. Avtonomov, B. Larsen, M. Tucholska, H. Choi, A. Gingras, and A. I. Nesvizhskii, "DIA-Umpire: comprehensive computational framework for

data-independent acquisition proteomics," *Nature Methods*, vol. 12, no. 3, p. 258–264, Jan. 2015. [Online]. Available: http://dx.doi.org/10.1038/nmeth.3255

[128] Y. Yin, R. Wang, Y. Cai, Z. Wang, and Z. Zhu, "DecoMetDIA: Deconvolution of multiplexed MS/MS spectra for metabolite identification in SWATH-MS-based untargeted metabolomics," *Analytical Chemistry*, vol. 91, no. 18, p. 11897–11904, Aug. 2019. [Online]. Available: http://dx.doi.org/10.1021/acs.analchem.9b02655

[129] I. Tada, R. Chaleckis, H. Tsugawa, I. Meister, P. Zhang, N. Lazarinis, B. Dahlén, C. E. Wheelock, and M. Arita, "Correlation-based deconvolution (CorrDec) to generate high-quality MS2 spectra from data-independent acquisition in multisample studies," *Analytical Chemistry*, vol. 92, no. 16, p. 11310–11317, Jul. 2020. [Online]. Available: http://dx.doi.org/10.1021/acs.analchem.0c01980

[130] C. Ludwig, L. Gillet, G. Rosenberger, S. Amon, B. C. Collins, and R. Aebersold, "Data-independent acquisition-based SWATH-MS for quantitative proteomics: a tutorial," *Molecular Systems Biology*, vol. 14, no. 8, Aug. 2018. [Online]. Available: http://dx.doi.org/10.15252/msb.20178126

[131] P. Willems, U. Fels, A. Staes, K. Gevaert, and P. Van Damme, "Use of hybrid data-dependent and -independent acquisition spectral libraries empowers dual-proteome profiling," *Journal of Proteome Research*, vol. 20, no. 2, p. 1165–1177, Jan. 2021. [Online]. Available: http://dx.doi.org/10.1021/acs.jproteome.0c00350

[132] J. Guo, S. Shen, S. Xing, and T. Huan, "DaDIA: Hybridizing data-dependent and data-independent acquisition modes for generating high-quality metabolomic data," *Analytical Chemistry*, vol. 93, no. 4, p. 2669–2677, Jan. 2021. [Online]. Available: http://dx.doi.org/10.1021/acs.analchem.0c05022

[133] A. Martínez-Val, K. Fort, C. Koenig, L. Van der Hoeven, G. Franciosa, T. Moehring, Y. Ishihama, Y.-j. Chen, A. Makarov, Y. Xuan, and J. V. Olsen, "Hybrid-DIA: intelligent data acquisition integrates targeted and discovery proteomics to analyze phospho-signaling in single spheroids," *Nature Communications*, vol. 14, no. 1, Jun. 2023. [Online]. Available: http://dx.doi.org/10.1038/s41467-023-39347-y

[134] E. E. Hubbard, L. R. Heil, G. E. Merrihew, J. P. Chhatwal, M. R. Farlow, C. A. McLean, B. Ghetti, K. L. Newell, M. P. Frosch, R. J. Bateman, E. B. Larson, C. D. Keene, R. J. Perrin, T. J. Montine, M. J. MacCoss, and R. R. Julian, "Does data-independent acquisition data contain hidden gems? a case study related to Alzheimer's disease," *Journal of Proteome Research*, vol. 21, no. 1, p. 118–131, Nov. 2021. [Online]. Available: http://dx.doi.org/10.1021/acs.jproteome.1c00558

[135] M. Walzer, D. García-Seisdedos, A. Prakash, P. Brack, P. Crowther, R. L. Graham, N. George, S. Mohammed, P. Moreno, I. Papatheodorou, S. J. Hubbard, and J. A. Vizcaíno, "Implementing the reuse of public DIA proteomics datasets: from the PRIDE database to Expression Atlas," *Scientific Data*, vol. 9, no. 1, Jun. 2022. [Online]. Available: http://dx.doi.org/10.1038/s41597-022-01380-9

[136] C. D. Broeckling, E. Hoyes, K. Richardson, J. M. Brown, and J. E. Prenni, "Comprehensive Tandem-Mass-Spectrometry coverage of complex samples enabled by Data-Set-Dependent acquisition," *Analytical Chemistry.*, vol. 90, no. 13, pp. 8020–8027, Jul. 2018.

[137] J. Wandy, V. Davies, J. J. J. van der Hooft, S. Weidt, R. Daly, and S. Rogers, "In silico optimization of mass spectrometry fragmentation strategies in metabolomics," *Metabolites*, vol. 9, no. 10, p. 219, Oct. 2019. [Online]. Available: http://dx.doi.org/10.3390/metabo9100219

[138] Z. Zuo, L. Cao, L.-F. Nothia, and H. Mohimani, "MS2Planner: improved fragmentation spectra coverage in untargeted mass spectrometry by iterative optimized data acquisition," *Bioinformatics*, vol. 37, no. Supplement_1, p. i231–i236, Jul. 2021. [Online]. Available: http://dx.doi.org/10.1093/bioinformatics/btab279

[139] E. W. Deutsch, *Mass Spectrometer Output File Format mzML*. Humana Press, Dec. 2009, p. 319–331. [Online]. Available: http://dx.doi.org/10.1007/978-1-60761-444-9_22

[140] J. Klein and J. Zaia, "psims - a declarative writer for mzML and mzIdentML for python," *Molecular & Cellular Proteomics*, vol. 18, no. 3, p. 571–575, Mar. 2019. [Online]. Available: http://dx.doi.org/10.1074/mcp.rp118.001070

[141] M. Kösters, J. Leufken, S. Schulze, K. Sugimoto, J. Klein, R. P. Zahedi, M. Hippler, S. A. Leidel, and C. Fufezan, "pymzML v2.0: introducing a highly compressed and seekable gzip format," *Bioinformatics*, vol. 34, no. 14, p. 2513–2514, Jan. 2018. [Online]. Available: http://dx.doi.org/10.1093/bioinformatics/bty046

[142] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars, *Computational Geometry: Algorithms and Applications*. Springer Berlin Heidelberg, 2008. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-77974-2

[143] I. E. Sutherland and G. W. Hodgman, "Reentrant polygon clipping," *Communications of the ACM*, vol. 17, no. 1, p. 32–42, Jan. 1974. [Online]. Available: http://dx.doi.org/10.1145/360767.360802

[144] K. Weiler and P. Atherton, "Hidden surface removal using polygon area sorting," in *Proceedings of the 4th annual conference on Computer graphics and interactive techniques*, ser. SIGGRAPH '77. ACM, Jul. 1977. [Online]. Available: http://dx.doi.org/10.1145/563858.563896

[145] B. R. Vatti, "A generic solution to polygon clipping," *Communications of the ACM*, vol. 35, no. 7, p. 56–63, Jul. 1992. [Online]. Available: http://dx.doi.org/10.1145/129902.129906

[146] G. Greiner and K. Hormann, "Efficient clipping of arbitrary polygons," *ACM Transactions on Graphics*, vol. 17, no. 2, p. 71–83, Apr. 1998. [Online]. Available: http://dx.doi.org/10.1145/274363.274364

[147] J. Culberson and R. Reckhow, "Covering polygons is hard," in *[Proceedings 1988] 29th Annual Symposium on Foundations of Computer Science*. IEEE, 1988. [Online]. Available: http://dx.doi.org/10.1109/SFCS.1988.21976

[148] J. J. J. van der Hooft, J. Wandy, F. Young, S. Padmanabhan, K. Gerasimidis, K. E. V. Burgess, M. P. Barrett, and S. Rogers, "Unsupervised discovery and comparison of structural families across multiple samples in untargeted metabolomics," *Analytical Chemistry*, vol. 89, no. 14, pp. 7569–7577, Jul. 2017. [Online]. Available: https://doi.org/10.1021/acs.analchem.7b01391

[149] Z. Galil, "Efficient algorithms for finding maximum matching in graphs," *ACM Computing Surveys*, vol. 18, no. 1, p. 23–38, Mar. 1986. [Online]. Available: http://dx.doi.org/10.1145/6462.6502

[150] A. Mehta, "Online matching and ad allocation," *Foundations and Trends in Theoretical Computer Science*, vol. 8 (4), pp. 265–368, 2013. [Online]. Available: http://dx.doi.org/10.1561/0400000057

[151] J. E. Hopcroft and R. M. Karp, "An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs," *SIAM Journal on Computing*, vol. 2, no. 4, p. 225–231, Dec. 1973. [Online]. Available: http://dx.doi.org/10.1137/0202019

[152] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using NetworkX," in *Proceedings of the 7th Python in Science Conference*, G. Varoquaux, T. Vaught, and J. Millman, Eds., Pasadena, CA USA, 2008, pp. 11 – 15.

[153] L. Ramshaw and R. E. Tarjan, "A weight-scaling algorithm for min-cost imperfect matchings in bipartite graphs," in *2012 IEEE 53rd Annual Symposium on*

Foundations of Computer Science. IEEE, Oct. 2012, p. 581–590. [Online]. Available: http://dx.doi.org/10.1109/FOCS.2012.9

[154] R. Karp, "An algorithm to solve the mxn assignment problem in expected time O (mn log n)," EECS Department, University of California, Berkeley, Tech. Rep. UCB/ERL M78/67, Sep 1978. [Online]. Available: http://www2.eecs.berkeley.edu/Pubs/TechRpts/1978/29160.html

[155] J. Edmonds, "Maximum matching and a polyhedron with 0, 1-vertices," *Journal of Research of the National Bureau of Standards Section B Mathematics and Mathematical Physics*, vol. 69B, no. 1 and 2, p. 125, Jan. 1965. [Online]. Available: http://dx.doi.org/10.6028/jres.069b.013

[156] R. Jonker and A. Volgenant, "A shortest augmenting path algorithm for dense and sparse linear assignment problems," *Computing*, vol. 38, no. 4, p. 325–340, Dec. 1987. [Online]. Available: http://dx.doi.org/10.1007/BF02278710

[157] A. Bernstein and C. Stein, *Fully Dynamic Matching in Bipartite Graphs*. Springer Berlin Heidelberg, 2015, p. 167–179. [Online]. Available: http://dx.doi.org/10.1007/978-3-662-47672-7_14

[158] 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, p. 283–289, Dec. 2011. [Online]. Available: http://dx.doi.org/10.1021/ac202450g

[159] M. Yu, G. Dolios, and L. Petrick, "Reproducible untargeted metabolomics workflow for exhaustive MS2 data acquisition of MS1 features," *Journal of Cheminformatics*, vol. 14, no. 1, Feb. 2022. [Online]. Available: http://dx.doi.org/10.1186/s13321-022-00586-8

[160] D. Gale and L. S. Shapley, "College admissions and the stability of marriage," *The American Mathematical Monthly*, vol. 69, no. 1, pp. 9–15, 1962. [Online]. Available: https://doi.org/10.1080/00029890.1962.11989827

[161] M. Wallace, "Practical applications of constraint programming," *Constraints*, vol. 1, no. 1–2, p. 139–168, Sep. 1996. [Online]. Available: http://dx.doi.org/10.1007/BF00143881