Efficient Workflows for Analyzing High-Performance Liquid Chromatography Mass Spectrometry-Based Proteomics Data

Dissertation

der Mathematisch-Naturwissenschaftlichen Fakultät

der Eberhard Karls Universität Tübingen

zur Erlangung des Grades eines

Doktors der Naturwissenschaften

(Dr. rer. nat.)

vorgelegt von

Johannes Veit (geb. Junker), M.Sc.

aus Filderstadt

Tübingen

2019

Gedruckt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Eberhard Karls Universität Tübingen.

Tag der mündlichen Qualifikation:	11.07.2019
Dekan:	Prof. Dr. Wolfgang Rosenstiel
1. Berichterstatter:	Prof. Dr. Oliver Kohlbacher
2. Berichterstatter:	Prof. Dr. Knut Reinert

“There is no time like the present for postponing what you ought to be doing.”

ANONYMOUS

Abstract

Modern high-throughput technologies in proteomics and related fields produce evergrowing amounts of complex experimental data. The wide range of techniques for quantification and identification of peptides and proteins and the wealth of available instrument types give rise to a wide range of computational challenges. As a consequence, computational data analysis has become a crucial bottleneck of the overall workflow in today's proteomics studies.

In this thesis, we present novel algorithms and tools for efficient automated data analysis of high-throughput LC-MS proteomics data, evaluate their performance in various benchmark settings, and demonstrate the successful application of our methods in a proteomics study in the field of forensics science. All tools developed in the context of this thesis are implemented in OpenMS, an open-source framework for computational mass spectrometry.

We introduce TOPPAS, a dedicated workflow engine for the analysis of LC-MS proteomics data using OpenMS. TOPPAS facilitates rapid construction of complex analysis workflows and offers parallel data processing on multi-core systems. The entire data processing workflow can be designed, tested, fine-tuned, executed, and documented in a single interface, thus providing researchers with a convenient way to organize and communicate their data analyses. The successor of TOPPAS, an OpenMS plugin for the popular workflow platform KNIME, takes this approach even one step further: In addition to the data processing tools provided by OpenMS, KNIME offers a wealth of available workflow nodes for downstream data manipulation, statistical analysis, and visualization. To enable analyses that require massive compute power, we provide the KNIME2gUSE extension for KNIME, which allows to export KNIME workflows to the Grid and Cloud User Support Environment (gUSE), which executes them on powerful grid and cloud resources. Finally, we present a free plugin for the popular commercial Proteome Discoverer platform (Thermo Scientific) making OpenMS algorithms available to an even larger group of non-bioinformatics experts: LFQProfiler for label-free quantification and RNP^{xl} for protein-RNA cross-linking data analysis.

Motivated by common issues of existing approaches for label-free quantification in the context of high sample complexity, we have developed OptiQuant, a novel method for label-free quantification using mixed integer programming for globally optimal feature detection in label-free proteomics experiments. The OptiQuant workflow includes FeatureLinkerUnlabeledKD,

a novel algorithm for retention time alignment and linking of corresponding signals across label-free LC-MS maps, which has become the state-of-the-art feature linking tool in OpenMS.

Last, but not least, we demonstrate the successful application of TOPPAS workflows for label-free quantification of proteomics data, statistical data analysis, and machine learning to assist in the forensic reconstruction of shooting incidents. Our proof-of-principle study demonstrates that proteomics can be used to match bullets to perforated vital organs based on the protein expression profiles found in traces of organic material remaining on the bullets. In cases involving multiple shooters, this information can help answer the crucial question: who fired the lethal bullet?

Zusammenfassung

Moderne Hochdurchsatz-Technologien in der Proteomik und in verwandten Gebieten produzieren immer größere Mengen an komplexen experimentellen Daten. Die große Auswahl an Methoden zur Quantifizierung und Identifikation von Peptiden und Proteinen, sowie die Fülle von verfügbaren Instrumententypen, führen zu einer großen Bandbreite von Herausforderungen in der Datenauswertung. Somit ist die rechnergestützte Datenanalyse ein kritischer Engpass heutiger Proteomik-Studien.

In dieser Arbeit präsentieren wir innovative Algorithmen und Anwendungen zur effizienten automatisierten Datenanalyse von Hochdurchsatz-LC-MS-Proteomik-Daten, vergleichen ihre Leistungsfähigkeit mit Hilfe verschiedener Benchmarks, und demonstrieren die erfolgreiche Anwendung unserer Methoden im Rahmen einer Proteomikstudie im Gebiet der Rechtsmedizin. Alle Anwendungen, die im Rahmen dieser Arbeit entwickelt wurden, sind als Teil von OpenMS implementiert, einem quelloffenen Softwarepaket für rechnergestützte Massenspektrometrie.

Wir präsentieren TOPPAS, eine dedizierte Workflow-Lösung zur Analyse von LC-MS-Proteomik-Daten mit Hilfe von OpenMS. TOPPAS ermöglicht den raschen Entwurf von komplexen Datenanalyse-Workflows und bietet Parallelprozessierung auf Multi-Core-CPUs. Der gesamte Ablauf der Datenanalyse kann an ein und derselben Stelle entworfen, getestet, optimiert, ausgeführt und dokumentiert werden, was es Wissenschaftlern ermöglicht, ihre Ergebnisse auf komfortable Weise zu organisieren und zu kommunizieren. Der TOPPAS-Nachfolger, ein OpenMS-Plugin für die Workflow-Plattform KNIME, geht noch einen Schritt weiter: Zusätzlich zu den Anwendungen, die von OpenMS selbst bereitgestellt werden, bietet KNIME eine große Auswahl an verfügbaren Workflow-Nodes zur anschließenden Datenmanipulation, statistischen Analyse, und Visualisierung. Um Analysen zu ermöglichen, die extreme Anforderungen an die Rechenkapazität stellen, stellen wir die KNIME2gUSE-Erweiterung für KNIME bereit, die es erlaubt, KNIME-Workflows in die Grid and Cloud User Support Environment (gUSE) zu konvertieren, wo sie dann auf leistungsfähigen Grid- und Cloud-Ressourcen ausgeführt werden. Schließlich präsentieren wir ein quelloffenes Plugin für die beliebte kommerzielle Anwendung Proteome Discoverer (Thermo Scientific), das OpenMS-Algorithmen einer noch größeren Gruppe von Nicht-Bioinformatikern zugänglich macht: LFQProfiler zur label-freien Quantifizierung und RNP^{xl} für Protein-RNA-Crosslinking-Analysen.

Angeregt durch häufig auftretende Probleme von existierenden Ansätzen zur label-freien Quantifizierung bei hoher Probenkomplexität haben wir OptiQuant entwickelt, eine innovative Methode zur label-freien Quantifizierung, die mit Hilfe von Mixed Integer Programming eine global-optimale Lösung für das Feature-Detektionsproblem berechnet. Der OptiQuant-Workflow beinhaltet FeatureLinkerUnlabeledKD, einen neuen Algorithmus zur Retentionszeitkorrektur und zum Zusammenführen einander entsprechender Signale über mehrere LC-MS-Proben hinweg, der mittlerweile die effizienteste Lösung für diese Probleme in OpenMS darstellt.

Zu guter Letzt beschreiben wir die erfolgreiche Anwendung von TOPPAS-Workflows zur label-freien Quantifizierung, statistischer Datenauswertung und maschinellem Lernen, um die rechtsmedizinische Rekonstruktion von Schießereien zu verbessern. Unsere Machbarkeitsstudie zeigt, dass die Proteomik es prinzipiell ermöglicht, Projektile und Schusskanäle einander zuzuweisen, anhand der Proteinexpressionsprofile in Spuren von organischem Material, das an den Projektilen haftet. Sind mehrere Schützen involviert, kann diese Information sehr hilfreich sein um die die entscheidende Frage zu beantworten: Wer gab den tödlichen Schuss ab?

Acknowledgments

First of all, I owe a tremendous debt of gratitude to my advisor Prof. Oliver Kohlbacher for giving me the opportunity to pursue research under his guidance, for offering his vast knowledge and expertise whenever it was needed, for his open-mindedness and supportive attitude not only in professional matters, and for expertly adjusting the mix of pressure and indulgence that has finally led to the conclusion of this work.

I would like to thank my fellow OpenMS developers from all over the world for being such a great team to be part of, and for all the blood, sweat, and tears that had to be invested at times. Many thanks to Timo Sachsenberg for building hundreds of PD node binary installers (apparently not getting tired), for his valuable feedback in general, and for proofreading parts of this manuscript in particular. I thank Fabian Aicheler for introducing me to the adventures of PD node development, as well as Hannes Röst, Chris Bielow, Stephan Aiche, Mathias Walzer, and Hendrik Weisser for fruitful cooperations and discussions. I want to thank all my other colleagues in Tübingen for creating the productive and enjoyable atmosphere I have had the privilege to work in. Thank you, Sven Nahnsen, for introducing me to the world of scientific conferences (and hospitality suites) at ASMS 2012 in Vancouver, and thank you, Sascha Dammeier, for an extraordinary cooperation on the interface of proteomics and forensics. My research would not have been possible without the incredible job Luis de la Garza did maintaining the coffee machine. A special thank you goes to Tonatiuh for triumphing over Tlaloc more often than not, granting us all those memorable BBQ sessions in the garden.

Last, but certainly not least, I am eternally grateful to my family and friends for their constant support throughout all these years. I want to thank my in-laws Jule and Seb for helping out with child care, for emotional support during writing, and for their persistence in asking for the current status. Thank you very much, Cece, for your company and back rubs during the final phase of writing. Many thanks to my parents Gertrud and Wolfgang and to my sister Tabea for everything they have given me until this day. I don't even know how to begin to thank my wife Lena for her (nearly) endless patience, enduring support, and love. Hugs and kisses to my son Timo for the special painting he put on the wall next to my desk. It did help.

Contents

Chapter 1

Introduction

In 2001, the first drafts of the sequence of the human genome were published as a result of a huge effort made by an international research project called the Human Genome Project and a private company named Celera Genomics[1,2]. Genome sequencing had been around earlier, and the human genome was not the first to be sequenced. Full genome sequencing of smaller genomes (4000 - 7000 base pairs) has been described as early as 1979[3]. The *Haemophilus influenzae* genome (1.8 million base pairs) was sequenced in 1995[4], *Drosophila melanogaster* (140 million base pairs) in 2000[5]. And yet, when the sequencing efforts culminated in the publication of the 3 billion base pair sequence of the human genome, a pivotal milestone in biology was reached, heralding the start of a new era of biological research using approaches that were fundamentally different from the ones existing before.

The availability of vast and ever-growing amounts of sequencing data led to a paradigm shift from classical reductionist, hypothesis-driven research to technology- and data-driven systems-level approaches. While classical reductionism tries to elucidate the function of a system by understanding all its parts, one after another, systems biology aims at investigating the entire system at once, as a whole. Here, the fundamental idea is that a system is more than the sum of its parts. It has certain properties that emerge only from the complex interplay between its individual components, and these properties cannot be understood by analyzing its parts in isolation. The holy grail of systems biology is to eventually understand (and hence be able to simulate) an entire biological system in all its complexity.

The rise of systems biology led to the emergence of a plethora of so-called *omics* fields, such as genomics, transcriptomics, proteomics, metabolomics, or interactomics. All omics technologies aim at investigating their respective *omes*, i.e., the entirety of the respective entities. In the case of transcriptomics, for instance, the aim is to study the entire transcriptome, which denotes the entirety of all messenger ribonucleic acid (mRNA) transcripts in a sample. Interactomics aims at elucidating the entirety of all interactions between certain kinds of

molecules. As of June 2018, the community-maintained list of omes and omics on omics.org[6] contained 364 entries (not all of which, admittedly, are equally relevant).[7]

This thesis deals mainly with data analysis in the field of proteomics. In contrast to the genome, which is static in an individual (cancer and epigenetics aside), the proteome is highly dynamic. One can think of the genome as an instruction manual describing all possible biological processes (i.e., which genes are present in an organism and *could* eventually be translated to proteins). The transcriptome describes what is actually supposed to happen under certain circumstances, at a certain point in time (i.e., which genes are supposed to be translated to proteins at this particular point and how many copies should be produced). Differential gene expression in different cells, tissues, organs, or at different points in time leads to different abundances of RNA on the transcriptome level. This, in turn, translates to differential abundances in the proteome, which comprises the *actual* set of proteins in a sample. On the proteome level, in contrast to the transcriptome, post-translational modifications like phosphorylation or acetylation constitute another important layer of complexity, as the modification state is often crucial for the function of a protein. Moreover, some proteins only become active when forming complexes with other proteins or various other kinds of molecules. Proteins are the key players in the cell as they are involved in virtually every biological function. Hence, proteomics provides a very detailed picture of the cellular processes taking place under specific circumstances. Differential proteomics plays a key role in understanding the molecular mechanisms of cellular signaling in general and in particular those of cancer and other diseases. Here, the proteome is analyzed and compared between different conditions (e.g., tumor tissue versus healthy tissue) or sampled multiple times in a time-series experiment where changes in the proteome are monitored over time.

The most important technology in modern high-throughput proteome research is mass spectrometry (MS), often coupled to high-performance liquid chromatography (HPLC, LC). The typical experimental workflow starts with a sample containing purified proteins which are digested to peptides using a protease, usually trypsin. These peptides are then injected into an HPLC system which performs a chromatographic separation to reduce complexity. he HPLC system is coupled on-line to a mass spectrometer which records the masses of all molecules entering it. From the peptide masses and their signal intensities, the original set of proteins in the sample can be reconstructed and quantified.

In today's "post-genomic era", proteomics is an important piece of the puzzle. The questions that can be asked are diverse and a variety of experimental techniques exist that can help answering them. While some definitions of the term "proteomics" include anything that deals with the study of proteins (including reductionist approaches like experimental structure elucidation), there is some consensus among high-profile proteomics scientists that this term should only be used for large-scale studies using high-throughput technologies that examine the proteome as a whole rather than concentrating on individual proteins. With the rise of

personalized medicine, large-scale clinical studies employing proteomics experiments with hundreds or even thousands of samples are becoming more and more common. As an example, the "Snyderome" project[8], a huge integrative multi-omics project initialized by Michael Snyder and co-workers in Stanford, has so far produced over half a petabyte of omics data of just one person (as of November 2015). This study has since been extended to 100 people and monitors, among other omes, the human blood proteome and the proteomes of the various microorganisms inhabiting the subjects' bodies over time. The goal is to further expand this study to a million people and to use the tools of big data to gain a better understanding of the interplay of various omes in health and disease, for identifying novel biomarkers for various conditions, and for finding new drug targets[9].

An obvious requirement for such large-scale studies is efficient automated data processing, as the sheer amount of raw data makes manual analysis impossible. In principle, there are two different approaches to automating data analyses: monolithic applications and modular workflow systems. The former have the advantage that they are often relatively easy to use. They are typically tailored towards one or few specific applications and require only little interaction with the user. Most of the commercial data analysis platforms for proteomics data fall into this category. The big disadvantage of monolithic solutions, however, is missing flexibility. Study designs, the combination of the various employed instruments, and other experimental conditions can vary immensely between different studies. More often than not, these differences should also be reflected in the data analysis part. This is where modular workflow systems come into play. They provide a means to composing custom data analysis workflows by offering many small building blocks, each of them solving one specific task, which can be chained together in a flexible manner. This approach ensures reusability of large parts of the data analysis workflows while enabling the user to flexibly adapt and fine-tune the analysis. Last but not least, using a single workflow from raw data to final results, including statistical downstream analyses and the creation of plots, is a great way of documenting the entire computational data analysis of a study and thus an important step towards reproducible science.

Besides automation, another requirement for the analysis of such vast amounts of data is the availability of efficient algorithms. The two most fundamental tasks in HPLC-MS-based computational proteomics are identification and quantification of peptides and proteins. The computational complexity of peptide identification usually scales linearly with the number of samples. This is because each spectrum can be identified independently of the other spectra and the number of samples. For quantification, however, this is not necessarily true. Here, one fundamental problem is to establish correspondence between identical analytes from different samples, so that their quantities can be compared. This problem arises mostly because of the poor reproducibility of the HPLC part: slightly different experimental conditions (such as temperature, amount of injected analyte, age of the HPLC column, etc.) lead to shifts in the

retention time of corresponding analytes across different HPLC runs. Shifts of up to several minutes are not uncommon. Hence, individual samples cannot be analyzed independently of each other and data analysis becomes more and more challenging with growing numbers of samples.

There are two major strategies to tackle this issue: labeled quantification and label-free quantification (LFQ). The various kinds of labeled quantification techniques all share the basic idea that multiple samples can be labeled and then mixed together, so that they can be measured in a single MS run. Labels are chosen such that they do not change the chromatographic properties of the peptides but introduce a well-defined mass shift. Thus, corresponding peptides originating from different samples elute at the same time. The problem of retention time shifts is eliminated. Because the mass difference introduced by these labels is known, linking corresponding analytes and assigning them to their respective samples of origin is relatively straightforward. However, all labeled approaches have one obvious drawback: the number of samples that can be quantified and compared is limited by the number of available labels. LFQ, on the other hand, is also suitable for large-scale quantitative proteomics studies as the number of samples is basically unlimited. LFQ methods can be further divided into spectral counting and intensity-based approaches. While spectral counting circumvents the additional computational effort of aligning corresponding analytes to each other, it is a rather unprecise quantification method. Intensity-based LFQ outperforms spectral counting in terms of quantification accuracy and sensitivity[10] and should thus be the method of choice. As mentioned above, these methods come at the price of additional computational problems that need to be tackled. Considering the huge amounts of data produced by modern mass spectrometers, this computational part nowadays represents the bottleneck of many large-scale proteomics studies.

The main contributions of this thesis are threefold: We have developed software for automated data processing of high-throughput LC-MS data using modular workflows, contributed new concepts and algorithms for sensitive and accurate label-free quantification of proteomics data, and successfully applied our tools in the context of an LC-MS proteomics study in the field of forensic science. All software developed in the context of this thesis is closely related to and has in large part been integrated into the OpenMS[11,12] project. OpenMS is a versatile open-source software framework for analyzing HPLC-MS proteomics and metabolomics data.

We present TOPPAS[13], the OpenMS Proteomics Pipeline Assistant, which has become the dedicated workflow engine of the OpenMS[11] software suite. TOPPAS is specifically designed for the analysis of LC-MS proteomics data using OpenMS/TOPP[12]. It enables fast construction of complex analysis workflows using all available TOPP tools, and also provides a mechanism to integrate external programs like the popular ProteinProphet[14] or ProteoWizard's raw file conversion tool msconvert[15]. From the development and use of TOPPAS, we have learned valuable lessons about the importance of various workflow language concepts.

The TOPPAS workflow language is rather minimalistic, and yet is able to implement almost any conceivable TOPP workflow. These ideas found their way into the implementation of the Generic KNIME Nodes (GKN)[16] plugin for the popular general-purpose workflow and data-analysis platform KNIME[17]. GKN allows to wrap arbitrary command line tools into KNIME nodes, as long as their command line interface can be described by a so-called Common Tool Descriptor (CTD) file. It enables file-based data flow (as opposed to KNIME's default, which is table-based in-memory data flow) and provides special nodes implementing a subset of the workflow language features of TOPPAS. The successor of TOPPAS, an OpenMS plugin for KNIME, was built on top of GKN. KNIME-OpenMS thus uses a workflow language very similar to that of TOPPAS. To enable analyses that require more computational resources than a single compute node can offer, we provide the KNIME2gUSE extension for KNIME, which allows to convert entire KNIME workflows to the Grid and Cloud User Support Environment (gUSE) workflow language[18]. With gUSE, workflows can then be executed on powerful grid and cloud resources.

Moreover, we present two new plugins for the Thermo Proteome Discoverer platform that make OpenMS algorithms available to an even larger group of non-bioinformatics experts: LFQProfiler for label-free quantification and RNP^{xl} for protein-RNA cross-linking data analysis. The tight integration with the built-in data processing and visualization tools of Proteome Discoverer make these more user-friendly than their equivalents in full-fledged workflow solutions like TOPPAS or KNIME, at the expense of modularity and flexibility.

Signal detection and quantification of raw data in datasets with high sample complexity is a key challenge in today's label-free proteomics studies, since overlapping signals can skew quantification or prevent feature detection altogether. Based on established concepts of a previously described LFQ workflow using OpenMS[19] and on a recently described signal detection approach originally designed for metabolomics data[20], we have developed OptiQuant, a novel method for label-free quantification using a mixed integer programming approach for globally optimal feature detection across all runs of a label-free experiment at once. The OptiQuant approach includes FeatureLinkerUnlabeledKD, a novel algorithm for retention time alignment and linking of corresponding signals across label-free LC-MS maps, which has shown to be substantially faster and thus replaced its predecessor, FeatureLinkerUnlabeledQT, as the state-of-the-art feature linking tool in OpenMS. We evaluate the overall quantification performance of the OptiQuant workflow in a series of benchmark comparisons with other state-of-the-art tools for LFQ.

Last, but not least, we demonstrate the successful application of TOPPAS workflows for label-free quantification of proteomics data, statistical data analysis, and machine learning to assist in the forensic reconstruction of shooting incidents. Matching bullets to victims by means of DNA analysis has become a routine task in modern forensics. However, it is still rather difficult to determine which projectile caused the lethal injury in cases involving multiple shooters

and bullet channels. In a proof-of-principle study, we demonstrate that proteomics can be used to match bullets to perforated vital organs based on the protein expression profiles found in traces of organic material remaining on the bullets. This study lays important groundwork for future efforts to establish our method as a routinely used technology in forensic science.

Chapter 2

Background

2.1 Mass Spectrometry-Based Proteomics

Mass spectrometry is a widely used analytical technique for measuring the mass and abundance of charged analytes. It has applications in various omics fields, most notably in proteomics and metabolomics. Here, the principal goal is to identify and quantify analytes of interest. In targeted omics experiments, the set of analytes of interest is determined *a priori*, whereas in untargeted omics, the entire ome (proteome, metabolome, lipidome, ...) is captured, without prior assumptions as to what will be detected. For proteomics, the analytes are either intact proteins or – more commonly – peptides resulting from the enzymatic digestion of proteins. If proteins are digested first, the approach is also referred to as shotgun (or bottom-up) proteomics, whereas experiments on intact proteins are called top-down proteomics experiments. Analyzing intact proteins has the obvious advantage that the identity and abundance of a protein of interest does not have to be reconstructed from the identity and abundance of its peptides. However, it comes with significant challenges in protein identification, since the mass and isotopic fine structure of an entire protein cannot be resolved accurately enough for unambiguous identification in the general case. In the following, we will provide an overview of the technology and discuss the fundamental data analysis challenges in mass spectrometry-based shotgun proteomics.

2.1.1 Sample Preparation

Sample preparation for LC-MS shotgun proteomics experiments typically involves a protein purification step and the enzymatic digestion of proteins to peptides. A variety of sample preparation tools and strategies exist, and careful and consistent sample preparation is a crucial first step in proteomics experiments. This is because inconsistencies in sample preparation often translate to significant technical variation in the measurement results. Here, we restrict ourselves to a brief explanation of the protein digestion step, since this is the fundamental

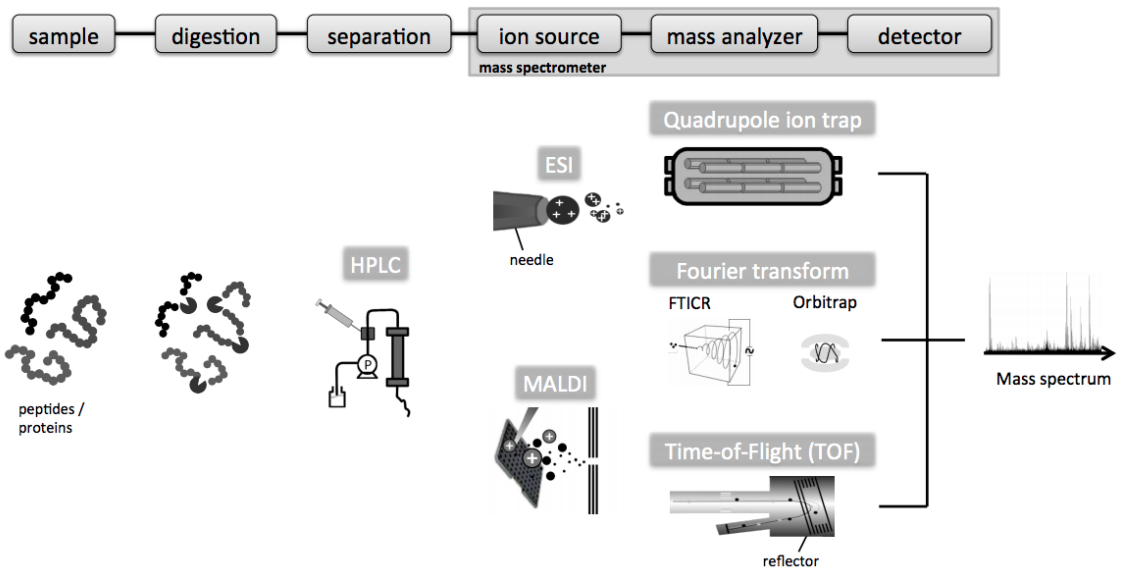

Figure 2.1: Basic components of a typical LC-MS proteomics experiment. A protein sample is digested to peptides, which then undergo a separation step before they enter the mass spectrometer (MS). The MS consists of three fundamental parts: the ion source, the mass analyzer, and the detector. Here, we show a selection of available types for the different components. Adapted from Bielow[21].

step producing the actual analytes. The digestion procedure has an important impact on the properties of these peptides.

Proteins are digested to peptides using a protease, most commonly trypsin. Alternatives are available, but trypsin is usually preferred because it has a number of desirable properties: It cleaves specifically after lysine and arginine residues, and thereby creates peptides with at least two positive charges (at the C- and N-terminus). This is important since the mass spectrometer can only measure charged analytes. The average length of tryptic peptides is 14 amino acids[22], which means that their average mass-to-charge ratio is well within the range of modern instruments.

2.1.2 Analyte Separation

The complex nature of omics samples often necessitates a separation step. It is virtually impossible to identify or quantify the proteins in a complex proteomic sample (e.g., whole cell lysate or serum) by recording and analyzing a single mass spectrum, due to the large number of overlapping signals (especially at lower resolutions[23]). Instead, the sample complexity is first reduced by means of a separation step. Traditionally, two-dimensional gel electrophoresis has been a popular method for peptide separation. In today's era of high-throughput omics technologies, however, this has widely been replaced by more modern methods that can be

automated and coupled online to the mass spectrometer and thus allow for a faster analysis of the proteome during separation.

The most commonly used peptide separation technique is high-performance liquid chromatography (HPLC)[24]. Here, the peptide solution is forced through a chromatography column at high pressure. Peptides are retained by the column for a certain amount of time. This achieves a separation of the peptides because the so-called retention time (RT) is different for different peptide species. It is (to some degree) reproducible for the same analyte across multiple chromatography runs using the same setup.

HPLC involves a stationary phase (the column) and a mobile phase (a solvent, usually containing water, acetonitril, and/or methanol). At $\mathrm{RT} = 0$, the peptide solution is injected into the mobile phase, and the resulting mixture is forced through the column at high pressure. Retention time is a function of certain physico-chemical properties of the analytes (e.g., hydrophobicity, ionic interactions), properties of the chromatography column, composition of the mobile phase, pressure, and temperature. The composition of the solvent can be changed over time. The resulting profile of solvent composition over time is called the HPLC gradient. The solvent composition has an impact on the interactions of the analyte with the two phases and thus affects the retention behavior of peptides in the column. A certain peptide species will only begin to elute from the column when its solubility in the mobile phase is high enough and outweighs the physico-chemical effects retaining it in the stationary phase. A typical HPLC gradient for complex samples has a duration of a few hours.[25]

2.1.3 Mass Spectrometry

Mass spectrometers are scientific instruments that can measure the mass – more precisely: the mass-to-charge ratio or m/z – of charged molecules and quantify their signal. Here, m is the molecular mass in Dalton (Da) or u, and z is the number of elementary charges. m/z is commonly considered a unitless quantity. The unit Thompson ($\mathrm{Th} = \frac{u}{e}$) has been proposed as an alternative notation but is rarely being used today[26].

The result of a single mass spectrometric scan is a mass spectrum with m/z on the x-axis and signal intensity (a unitless measure for the abundance of detected ions) on the y-axis. Modern instruments are sufficiently accurate to allow for reliable charge state determination of the measured analytes and hence the determination of their actual mass. Compound identification can then be achieved either by comparing the accurate mass of an intact analyte with a set of expected masses of candidate molecules, or by analyzing the masses of certain fragments of the compound resulting from a purposefully induced fragmentation step.

Two numbers describe important characteristics of any mass spectrometer: the *resolution* and the *mass accuracy* of an instrument[27]. For some instruments, the resolution decreases with

increasing mass-to-charge ratio. Per convention, it is thus usually reported for $m/z = 400$ and defined as

$$R = \frac{(m/z)}{\Delta(m/z)}, \tag{2.1}$$

where $\Delta(m/z)$ is defined as the full width at half maximum (FWHM) of the approximately Gaussian shaped raw peak signal for any given analyte measured at this m/z. The resolution is thus inversely correlated with the minimum m/z distance between the maxima of two neighboring (overlapping) peaks such that they can still be told apart by the mass spectrometer. More intuitively, $\Delta(m/z)$ is the peak distance between two overlapping peaks of equal intensity at this m/z that are separated by a valley which has a minimum at 50% of the peaks' maximum intensity.

The mass accuracy is the relative error of the measured mass of an ion compared to its theoretical mass in parts per million (ppm). Mass accuracy can vary from measurement to measurement, but the average mass accuracy is an informative instrument-specific property of mass spectrometers.

$$\text{Mass accuracy} = \frac{(m/z)_{measured} - (m/z)_{theoretical}}{(m/z)_{theoretical}} \tag{2.2}$$

The basic components of a mass spectrometer are depicted in Figure 2.1: The ion source, the mass analyzer, and the detector. The latter two operate in a vacuum. There are different varieties for each of these modules. In the following, we will discuss the most notable examples used in modern instruments.

Ion Source

There are essentially two different kinds of ion sources that are commonly used in mass spectrometry-based proteomics experiments: electrospray ionization (ESI)[28] and matrix-assisted laser desorption/ionization (MALDI)[29] sources. Both are so-called soft ionization techniques, as they leave the analyte mostly intact.

For MALDI, the peptide sample is first mixed with a solution of matrix molecules. This solution is then spotted onto a metal plate, the solvent evaporates, and the matrix molecules crystallize. Now, the peptides are embedded (co-crystallized) in the matrix. In order to achieve the actual ionization, short laser pulses are shot at the crystallized spots, and the resulting ions enter the mass spectrometer. MALDI is an offline technique, as it cannot easily be coupled directly to a continuous separation technique like HPLC. If separation is performed prior to mass spectrometry, fractions of eluting analytes have to be collected and spotted onto the plate individually. One important advantage of MALDI is the fact that the crystallized samples can be stored and re-analyzed, because only a fraction of the analytes in the crystal is ionized by a

single laser pulse. Thus, it is possible to perform an initial experiment to gather preliminary data, and then use these data to optimize instrument parameters for the main run. In some applications (e.g., clinical or forensic), the ability to store and re-analyze samples later may also be of significance from a legal perspective.[30]

The most widely used ionization technique is ESI. Electrospray ion sources can be coupled online to the HPLC, which makes it the method of choice for high-throughput applications. A simple schematic of the electrospray ionization process is shown in Figure 2.2. The analyte solution entering the ESI source is forced through a fine spray needle at the tip of the HPLC column. A high voltage is applied to this needle, leading to the formation of positively charged droplets of peptide solution. These droplets are directed towards the (negatively charged) mass spectrometer through a near-vacuum heated region in which the solvent evaporates. Once the positive charge of an evaporating droplet exceeds a certain limit, it can dissociate explosively. At the entrance to the mass spectrometer, all peptides are effectively desolvated, leading to a constant stream of positively charged peptide ions entering the mass analyzer. The exact physical processes involved in ESI are not yet completely understood.[31]

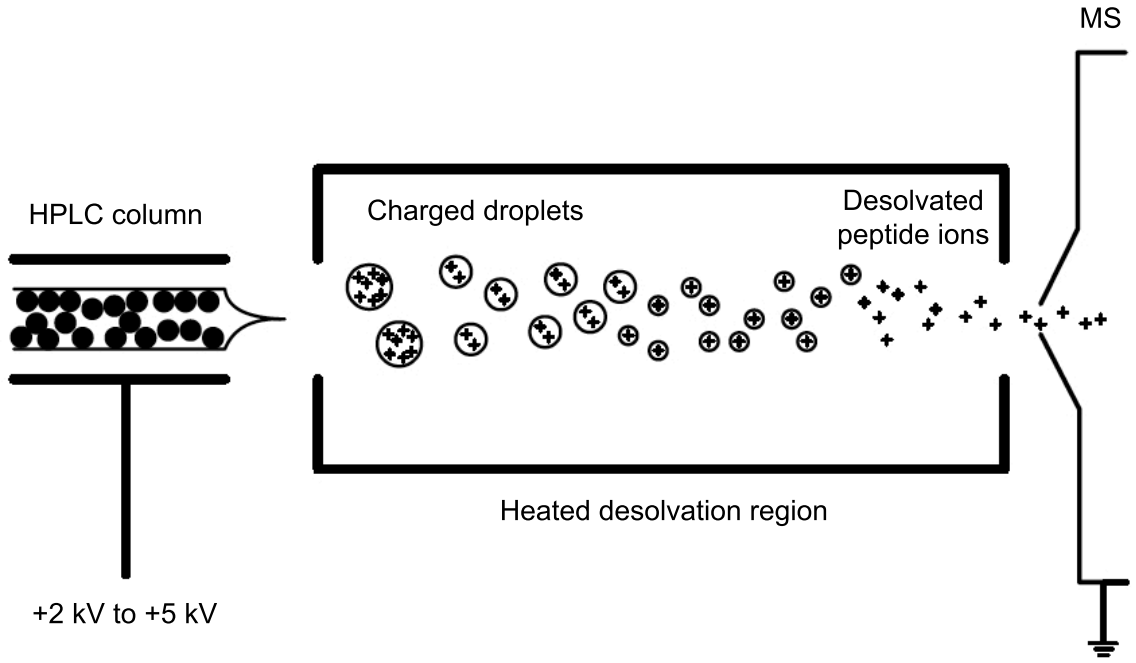

Figure 2.2: Schematic of an ESI source. At the tip of the charged spray needle, charged droplets of peptide solution form and are directed through the heated desolvation region towards the mass spectrometer. On their journey, the solvent evaporates and the charged peptides are completely desolvated when they enter the mass spectrometer. Adapted with permission from Kinter and Sherman[32]. Copyright 2005 John Wiley and Sons.

Mass Analyzer

The mass analyzer of a mass spectrometer is responsible for determining the m/z of the injected ions. A variety of different mass analyzer designs are available today. They can be roughly divided into three different groups:

- **Selective mass filters**, which create an electromagnetic field that lets only ions of a certain m/z ratio pass on to the detector and discards all others. A prominent example is the linear quadrupole mass filter.[33]
- **Time-of-flight (TOF)** mass analyzers, which infer the m/z of an ion by accelerating it through an electric field and measuring the time elapsing until it reaches the detector. The time of flight of an ion is proportional to the square root of its m/z ratio.[34]
- **Fourier transform MS (FTMS)** analyzers, which infer the m/z of a set of analytes by Fourier-transforming the measured current induced by ions oscillating in an electric (Orbitrap) or magnetic (Fourier-transform ion cyclotron (FTICR)) field.[35]

All datasets analyzed in this thesis have been generated by Orbitrap[36] hybrid instruments, in which the Orbitrap is combined with an additional linear trap quadrupole (LTQ) or with a linear quadrupole mass filter (Thermo Fisher's QExactive instrument family). These analyzer types shall thus be highlighted briefly.

Linear quadrupoles have found multiple uses in the design of mass spectrometers. They can either serve as a selective mass filter in the way described above, or – in combination with an additional electric field constraining the ions' movement in the axial direction – they can serve as a linear ion trap that selectively stores ions over time. Quadrupoles are also used as collision chambers, where priorly selected ions can be fragmented by means of collision-induced dissociation (see Section 2.1.3 for more information on this fragmentation step). The basic setup of a linear quadrupole is depicted in Figure 2.3: four parallel cylindrical metal rods create an electric field through which only ions of a certain m/z can reach the detector. Ions arriving at the detector at any given time can be attributed to the m/z currently selected by the quadrupole. In order to record a full mass spectrum, the entire mass range has to be scanned like this. In some modern instrument designs, the quadrupole has been replaced by a more efficient hexapole or octupole, but the general functional principle remains the same.[33]

The Orbitrap[36] is one of the most popular high-resolution mass analyzers in mass spectrometry-based proteomics. As a matter of fact, the traditional distinction between mass analyzer and detector is not entirely appropriate here. In FTMS instruments like the Orbitrap, it would be more adequate to speak of a unified "mass analyzer and detector" module, which uses a subtly different strategy to perform the measurement: Here, a packet of ions (of different m/z) is trapped in an electric field orbiting around a central spindle-like electrode while oscillating

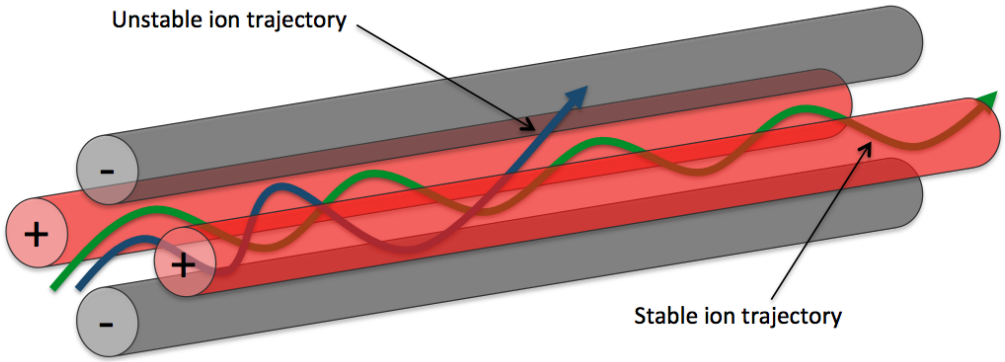

Figure 2.3: A linear quadrupole. The two diagonal pairs of rods are connected to a DC offset voltage of equal magnitude and opposite sign. In addition, a radio frequency (RF) voltage is applied, where the phases of the two pairs are shifted by 180°. This setup creates an oscillating electromagnetic field through which only ions of a specific m/z can pass on a stable spiral trajectory, whereas all other ions collide with the rods and are thus filtered out. By adjusting the applied voltages, the quadrupole can be set to select ions of a certain m/z ratio.

back and forth along the spindle axis. The current induced by the oscillation of these ions is measured on an outer barrel-like electrode. The recorded signal corresponds to the sum of all currents induced by each individual ion. It can thus be Fourier-transformed to reconstruct the m/z and abundance of all ions contained in the packet. As the frequencies of the induced current can be determined very accurately, Orbitraps and other FTMS instruments have a comparatively high mass accuracy and resolution. Trapping and analyzing ion packages for longer periods of time allows more accurate averaged measurements but comes at the expense of longer cycle times. The basic setup of an Orbitrap mass analyzer and detector is explained in Figure 2.4.

Detector

While FTMS instruments like the Orbitrap use a combined "analyzer/detector" module as explained above, TOF and quadrupole mass spectrometers have a separate detector module whose sole purpose is the quantification of incoming charged particles. The detector itself is agnostic of the mass of these ions, it only registers the incoming charge. In TOF mass spectrometry, the time elapsed between acceleration and detection determines the m/z of the detected ion. With quadrupole mass filters, the m/z of an ion registered by the detector corresponds to the current voltage setting of the quadrupole. The most basic detector design is

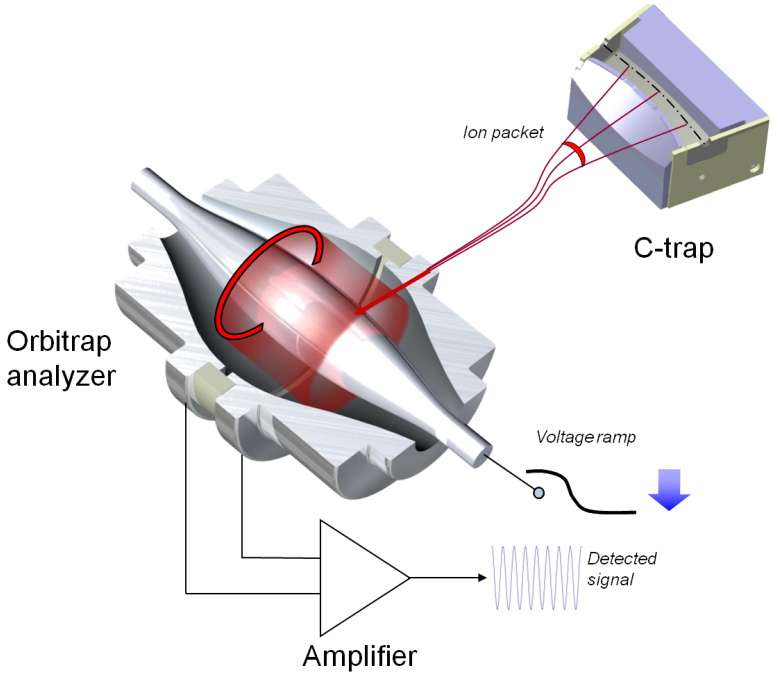

Figure 2.4: Schematic of an Orbitrap. Before entering the mass analyzer, ions are collected in a curved ion trap (C-trap). The ion beam is focused and an ion packet is directed into the Orbitrap. Here, ions orbit in an electric field around the central spindle-like electrode while oscillating back and forth along the spindle axis. The frequency of an ion's axial oscillation corresponds to its m/z. The frequency spectrum for the entire ion packet can be computed via Fourier transformation of the measured signal. Artwork by Thermo Fisher Scientific (CC-BY-SA 3.0)[37].

the Faraday cup. Here, incoming ions collide with a metal plate carrying the opposite charge. The current compensating for the resulting charge transfer between the ions and the plate can be measured and corresponds to the abundance of incoming ions. A more sensitive variant is the electron multiplier, where incoming ions trigger the emission of electrons, which in turn trigger the emission of even more electrons, leading to a positive feedback loop that amplifies the original signal by several orders of magnitude. Both Faraday cups and electron multipliers are so-called destructive detection methods, since the ions are immediately decharged upon contact with the detector and thus cannot be analyzed any further. In contrast, the "analyzer/detector" modules utilized in FTMS instruments are non-destructive.

Tandem Mass Spectrometry

It is infeasible to unambiguously identify higher-molecular-weight biomolecules (such as peptides and proteins) based on their accurate mass alone. Even if mass spectrometers had sufficient resolution and mass accuracy to allow for the unambiguous identification of the analytes' sum formulae, this would still not provide any insight into the structural arrangement of the atoms. However, an important breakthrough in mass spectrometry, the invention of tandem mass spectrometry, has enabled scientists to reliably identify peptides and other analyte species using an additional fragmentation step.

In tandem mass spectrometry, mass spectrometric measurements are performed at two different levels, the so-called MS^1 and MS^2 level. The MS^1 scan (or survey scan) produces a conventional full mass spectrum of the intact analytes currently emitted by the ion source as described above. This spectrum is referred to as the MS^1 (or simply MS) spectrum. Now, the key idea is that analytes detected in this survey scan can be selected and redirected to a collision cell, where they are purposefully fragmented by collision with a natural gas in a process called *collision-induced dissociation (CID)*. As a result, the peptides break into characteristic fragments that can be further analyzed to reconstruct their amino acid sequence. Thus, after CID, the fragments are measured in an additional MS^2 scan. The resulting fragment spectrum is called an MS^2 or MS/MS spectrum. The parent ion selected for fragmentation is often referred to as the *precursor* ion, its fragments are also called *product* ions. For more information on the data analysis aspects of peptide identification via tandem mass spectrometry, see Section 2.2.2.[38,39]

2.2 Computational Mass Spectrometry

In a typical 2-4 hour HPLC-MS proteomics experiment, a modern mass spectrometer can produce several gigabytes of raw data. The ever-growing volumes of generated data and the inherent variety of algorithmic and statistical data analysis challenges have made computational mass spectrometry an indispensable interdisciplinary research field on the interface of analytical

chemistry, statistics, computer science, and biology. The topics addressed span a wide range, including (but not limited to) the compact storage of raw data in free formats, consistent representation of associated metadata, efficient algorithms for signal detection, data reduction, peptide identification, statistical validation of identifications, peptide quantification, protein-level inference from peptide data, downstream statistical analyses, and automation of high-throughput data processing using workflow systems.

We will begin with an overview of basic concepts, data characteristics, and nomenclature conventions in HPLC-MS/MS data analysis, followed by an overview of common approaches to the most fundamental problems in computational proteomics: peptide identification and quantification. Finally, we will provide a brief introduction to OpenMS, an open-source framework for mass spectrometry data analysis which has served as the primary development library and toolbox for the work described in this thesis.

2.2.1 Basic Concepts and Terminology

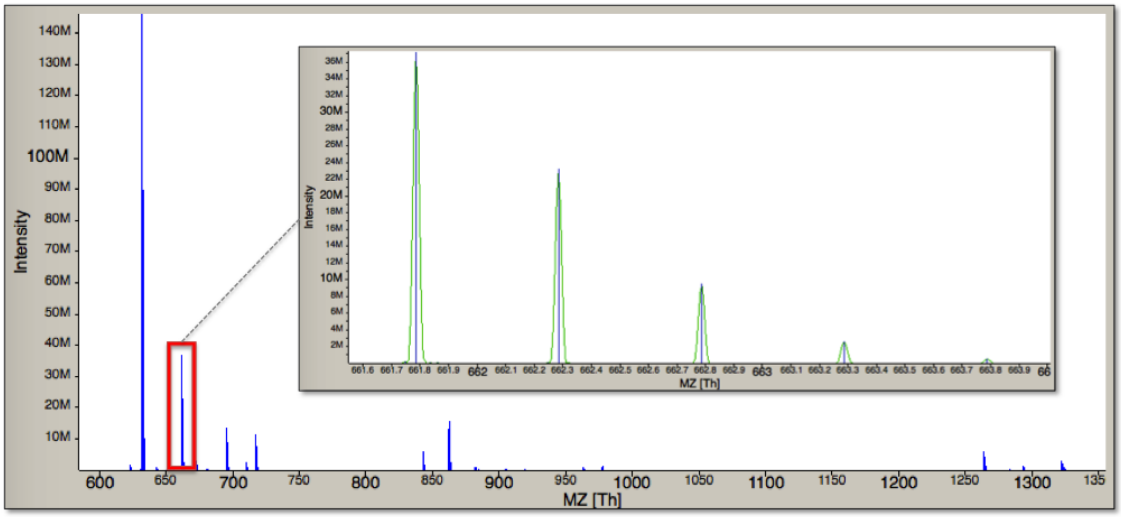

Figure 2.5: Example of a mass spectrum. The outer graph depicts an MS spectrum for a selected m/z range. The embedded graph shows a zoomed-in version of one of the mass spectrometric signals in which the characteristic isotope pattern of the analyte becomes visible. Raw data peaks (as recorded by the instrument) are shown in green, centroided peaks resulting from a post-acquisition peak picking step are shown in blue.

Mass Spectra and Peaks

A mass spectrum is the entirety of all mass spectrometric signals recorded in a single scan across the instrument's m/z range. The x-axis of a mass spectrum is the m/z, and the y-axis is the recorded intensity. A selected region of a typical mass spectrum is depicted in Figure 2.5.

An important detail to note is that the peaks recorded by the instrument are not simply pairs of m/z and intensity, but instead have an approximately Gaussian shape with a certain peak width that depends on the instrument resolution (see Section 2.1.3). While some instruments have optional built-in peak picking algorithms developed by the vendor, it is sometimes preferred to record the spectra in raw profile mode and perform a post-acquisition peak picking step using a third-party peak picking algorithm, such as the ones implemented in OpenMS (see 2.2.4) or the popular raw file conversion tool msconvert[15]. With modern high-resolution mass spectrometry data, the loss of information due to peak picking is negligible. The difference in raw data disk space requirements and efficiency of downstream data processing algorithms, however, is significant.

Isotope Patterns

With sufficient instrument resolution, the signal of a single analyte in a mass spectrum consists of more than one peak, due to the incorporation of isotopes. Isotopes are atoms of the same element with the same number of protons but different numbers of neutrons (and thus different mass). In nature, isotopes occur with element-specific abundances. The most abundant isotope of carbon, for instance, is carbon-12 (^{12}C) with 6 protons and 6 neutrons. It accounts for around 99% of all carbon on earth. The remaining 1% is ^{13}C with one additional neutron. One distinguishes between the *monoisotopic mass* and the *average mass* of an element's atoms: the monoisotopic mass is the mass of its most abundant isotope, whereas the average mass is the weighted average mass of all naturally occuring isotopes (weighted by relative abundance). These two different notions of mass can be transferred to the molecular level simply by summing up the monoisotopic or average masses of a molecule's atoms. Otherwise identical molecules containing different isotopes, i.e., differing only in the number of neutrons, are called *isotopologs*. Naturally, the isotope distribution for isotopologs is more complex than atom-level isotopic abundances. For molecules comprised of a single element, it corresponds to a multinomial distribution[1]. With increasing number of contained elements, however, computing the isotope distribution of a molecule becomes more involved. The OpenMS implementation used in this thesis employs a convolution-based approach, which performs $O(\log n)$ convolution operations for an upper bound n on the *isotopic rank* (number of additional neutrons). In shotgun proteomics, $n < 10$ is usually sufficient, since the probability to observe more is vanishingly low for almost any peptide.[38,40]

For proteomics data, a simple trick allows us to approximate the isotope distribution for arbitrary peptides based on their mass alone: Since peptides and proteins are comprised of amino acids, we can simply compute the sum formula of a hypothetical "average" amino acid, the so-called *averagine*, comprising average amounts of the elements carbon, hydrogen, ni-

[1] binomial if the element has exactly two isotopes

trogen, oxygen, and sulfur (CHNOS) found in amino acids. The fractional sum formula of averagine is $C_{4.9384}H_{7.7583}N_{1.3577}O_{1.4773}S_{0.0417}$, its molecular mass is 111.1254 Da[41]. For any given peptide mass, the numbers of CHNOS atoms can thus be approximated by dividing the mass by 111.1254 Da, multiplying the resulting "number of averagines" with the respective average number of atoms for each of these elements, and filling the remainder with additional hydrogen atoms. With these numbers, the isotope distribution of the peptide can be approximated as described above. This will be relevant in Section 4.2.3, where we use the averagine model to score hypotheses in a peptide signal detection algorithm.

It is important to note that the mass difference between an isotope with N neutrons and an isotope with $N+1$ neutrons does not exactly equal one neutron mass. This is because the presence of an additional neutron also has an impact on the nuclear binding energy. With $E = mc^2$, this translates to a certain mass difference, contributing to the element-specific *mass defect* of the atom. For instance, the mass difference between ^{12}C and ^{13}C amounts to -1.003355 Da, whereas the masses of ^{1}H and ^{2}H differ by -1.006277 Da. Thus, on the molecular level, this means that there is not only a single mass for isotopologs with one additional neutron, but one for each contained element. Modern ultrahigh-resolution instruments are capable of resolving this so-called *isotopic fine structure* to a certain degree (depending on the molecular weight and element composition of the analyte).

Figure 2.5 shows the isotope pattern of a doubly charged peptide. Because the most abundant isotopes of CHNOS are also the lightest isotopes of these elements, the monoisotopic peak of a peptide is always the one with lowest m/z. Here, the monoisotopic (leftmost) peak is also the most intense one. Apart from the monoisotopic peak, we can observe additional isotopic peaks for isotopologs with up to four additional neutrons. The corresponding peptide has charge $z = 2$. The m/z distance between two consecutive isotopic peaks is thus approximately $\frac{1.003355\text{Da}}{z} \approx 0.501678$. Note that we use the mass difference between ^{12}C and ^{13}C for this calculation. Although not entirely correct, this works well in practice as ^{13}C isotopes dominate the isotopic fine structure due to their comparatively high natural abundance and the high carbon content of peptides.

LC-MS Maps

LC-MS datasets consist of a series of mass spectra, recorded at a certain interval (e.g., every second) for the duration of the HPLC gradient. So far, we have only considered individual mass spectra. Adding retention time as an additional dimension (one time point per spectrum) yields a three-dimensional representation of the whole LC-MS dataset, which we refer to as an *LC-MS map* or *peak map*. As the total signal caused by an analyte is three-dimensional (a set of points [RT, m/z, intensity]), algorithms for peptide signal detection and quantification usually operate in this space rather than on individual mass spectra. Figure 2.6 shows a 3D

visualization of a small selected region from a map of an LC-MS experiment measuring yeast whole-cell lysate, acquired on an LTQ Orbitrap XL mass spectrometer[10].

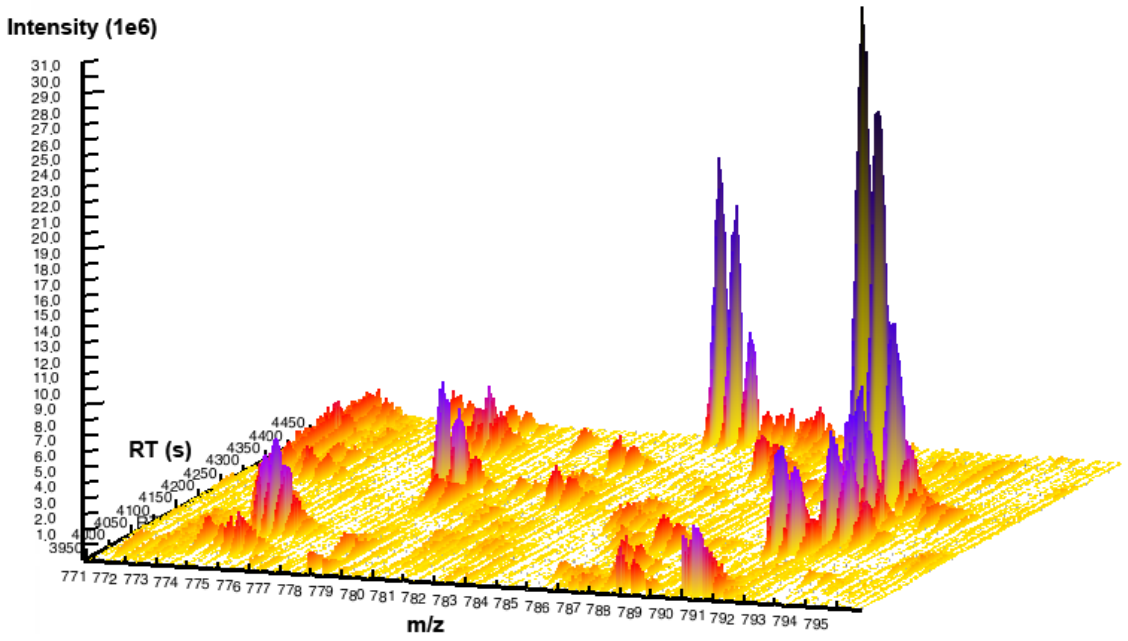

Figure 2.6: 3D view zoomed into a small region of an LC-MS map. Several peptide features can be distinguished, each showing its characteristic isotope pattern in the m/z dimension and an approximately Gaussian elution profile in the RT dimension.

Chromatograms

Another fundamental data type in LC-MS data analysis is the *chromatogram*, the chromatographic counterpart of the mass spectrum. Here, signal intensity is described as a function of retention time rather than m/z. For LC-MS data, a chromatogram can be obtained simply by summing up the mass spectrometric signals within a certain m/z window for each spectrum. The so-called *total ion chromatogram (TIC)* sums across the entire m/z range. It is a valuable diagnostic measure for HPLC quality control, as it approximates[2] the total amount of eluting analytes over time. In contrast, *extracted-ion chromatograms (XIC)* isolate a narrow m/z window, typically to extract and quantify the signal of a single analyte as a function of retention time. The shape of chromatographic peaks varies with certain HPLC parameters and experimental conditions. It is usually modeled as a Gaussian, or as an exponentially-modified Gaussian if peaks show significant asymmetry (tailing).

[2] not "equals" since the TIC can be distorted by varying analyte-specific ionization efficiencies

Mass Traces and Features

In the context of MS-based quantification, the term *feature* refers to the entire MS signal caused by a specific charge variant of a single analyte (potentially with adducts). In Figure 2.6, we can recognize a number of different peptide features. Each consists of a few (3-5) co-eluting *mass traces* showing an approximately Gaussian elution profile. The m/z distances between mass traces as well as their intensity distribution is determined by the analyte-specific isotope distribution. Feature detection (or feature finding) denotes the process of detecting and quantifying all analyte features present in the data. The overall intensity of a feature is usually computed as the sum of the areas under the chromatographic peaks of its isotopic traces. Thus, in feature detection, the entire signal of a peptide, consisting of hundreds or thousands of mass spectrometric peaks, is basically condensed into four numbers: the m/z and RT of the chromatographic apex of the monoisotopic mass trace, the charge of the corresponding peptide, and the overall intensity.

2.2.2 Identification

Basic Idea

As mentioned in Section 2.1.3, tandem mass spectrometry allows the identification of peptides. In this section, we want to address the data analysis aspects of peptide identification via tandem mass spectrometry. The basic idea is that, due to collision with natural gas in the collision cell, the peptide breaks into characteristic fragments, and comparing these fragment masses to the known masses of amino acids (or to sums of several) allows the reconstruction of the peptide sequence. The most frequently breaking bond is the amide bond between the α-carboxyl group of one amino acid and the α-amino group of another, leading to the formation of b- and y-ions (see Figure 2.7).[39]

De Novo vs Database Search

Peptide identification from tandem mass spectrometry data without using external sequence information (candidate sequences) is referred to as *de novo* sequencing. Current state-of-the-art tools for *de novo* identification include PepNovo[42], NovoHMM[43], and Antilope[44]. Here, the identification is done solely by detecting fragment ion series in the spectrum and attributing their mass differences to matching amino acids. Although conceptually very attractive, *de novo* sequencing has difficulty identifying spectra with missing peaks and struggles with the combinatorial complexities inherent in this kind of analysis, especially in the presence of noise. In general, these methods thus typically require relatively long runtimes and offer moderate identification rates[45].

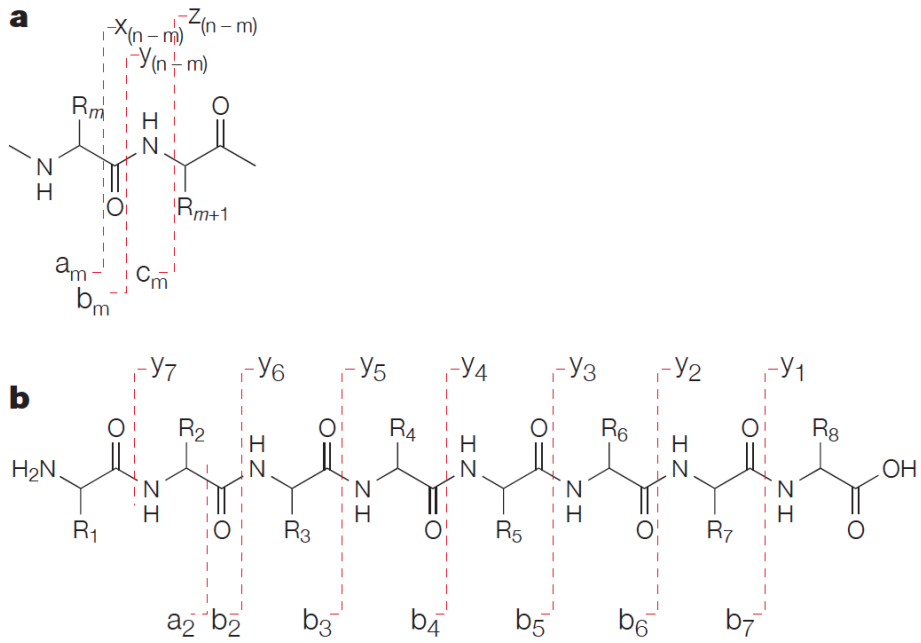

Figure 2.7: Different types of peptide fragments occuring in CID. (a) depicts the three most frequent fragmentation sites along the backbone of the peptide. Each of those corresponds to a characteristic set of resulting fragment ions: a/x, b/y, or c/z-ions. a/b/c ions start with the N-terminal amino acid of the peptide and stop before the fragmentation site, x/y/z ions are the respective C-terminal complements. (b) shows the typical b- and y-ion series that are most relevant for peptide identification. Reprinted with permission from Steen and Mann[39]. Copyright 2004 Springer Nature.

Therefore, *database search* algorithms have become the more popular alternative when the investigated organism is known and protein sequences are available. Here, we generate a list of theoretically expected peptide fragment masses for a database of candidate proteins. We start by *in silico*-digesting the proteins using the enzyme-specific cleavage rules (see Section 2.1.1) in order to obtain peptide sequences, and then compute the theoretical masses of all potentially occuring b- and y-ions for all of these theoretical peptides. The observed fragment spectra can then be compared against theoretical spectra of peptides with matching precursor mass. If a sufficient similarity score is achieved, a peptide-spectrum match (PSM) is reported. A variety of different database search algorithms are available. Some of the most popular ones are Mascot[46], SEQUEST[47], OMSSA[48], X!Tandem[49], MS-GF+[50], Comet[51], and Andromeda[52].

Validation

Database search engines differ in their algorithmic approaches and, most notably, in their scoring functions for PSMs. A common property of these PSM scores, however, is that they have little to no statistical meaning as such. For this reason, it is usually required to perform an additional re-scoring step, where the statistical significance of identifications is assessed and insignificant hits are filtered out. The basic approach for this is the so-called *target-decoy search*: Prior to the search engine run, we add decoy sequences to the database, one for each target sequence. Decoys are constructed in such a way that they are unlikely to coincide with target peptide sequences, but show similar statistical properties (precursor mass distribution, amino acid composition). Simply using the reverse sequence of each protein is the recommended approach, as it preserves amino acid frequencies and results in decoy peptide sequences unlikely to coincide with target peptide sequences.

Now, the key idea is that we can use the distributions of target and decoy PSM scores to control the false discovery rate (FDR) of an identification run. The simplest approach is to sort PSMs by search engine score and to then assign a *q-value* to each PSM by computing the relative amount of decoy PSMs among all those PSMs that have a score better or equal to the PSM currently considered. Discarding all PSMs with a q-value greater than a user-specified FDR threshold t controls the overall PSM-level FDR of the identifiation results. More advanced alternatives exist, for instance the semi-supervised machine learning-based tool Percolator[53,54], which learns to discriminate between correct and decoy spectrum identifications and thus achieves higher percentages of correctly assigned peptide identifications than the traditional q-value filtering approach.

Another approach to increase the number of correctly identified peptides at a certain FDR is the combination of search results from several database search engines. In order to establish comparability between PSMs produced by different search engines, the engine-specific PSM scores must first be transformed to a universally meaningful statistical measure, such as poste-

rior (error) probability. Then, analyzing the degree of consensus or dissent among the search engines allows for better estimates of PSM probabilities, and thus yields an increased number of correct identifications at a fixed FDR. The two most important tools for combining results of different search engines are ConsensusID[55] (part of OpenMS) and iProphet[56] (part of the Trans Proteomic Pipeline (TPP)[57,58]).

Since a single eluting peptide can cause multiple LC-MS peptide signals due to the occurence of different charge variants, and because proteins consist of multiple peptides, the FDR of identification results propagates from PSM level to peptide level to protein level. In general, PSM-level FDRs are an underestimate of the peptide-level FDR, and an even grosser underestimate of the protein-level FDR. For this reason, a number of tools have been proposed to control peptide- and protein-level FDRs of identification results. Popular tools include PeptideProphet[59], ProteinProphet[14], and Mayu[60].

2.2.3 Quantification

A key challenge in quantitative mass spectrometry arises from the fact that the efficiency of ionization in the ion source is analyte-specific. Let us consider a sample containing two different analytes A and B in equal concentration. More likely than not, the MS signals of these two analytes will *not* have equal intensity, as one is more likely to ionize than the other. This effect can distort quantification by several orders of magnitude. For this reason, quantitative signals of different analytes can in general not be compared. An exception are isotopologs measured in the same run. These will show practically identical ionization behavior. Thus, quantification in LC-MS usually means relative quantification of the same analyte across different runs (or channels, see Section 2.2.3), or absolute quantification by adding spike-in isotopologs in known concentration. In the following, we will provide an overview of the most widely used quantification strategies and technologies.

Targeted vs Untargeted

A fundamental classification of quantification strategies is the distinction between targeted and untargeted approaches. In targeted approaches, the analyte (or set of analytes) to be quantified is already specified *a priori*, whereas in untargeted quantification, the goal is simply to identify and quantify *all* analytes. An important experimental technique for targeted analysis is *selected reaction monitoring (SRM)*, which is typically done on triple quadruople (QqQ) mass spectrometers. A QqQ MS consists of a linear series of three quadrupoles where the first one selects a certain set of precursor masses (one for each target analyte), the second quadrupole acts as a collision chamber for CID, and the third quadrupole selects a set of fragment masses for each precursor mass. Such a pair of precursor mass and fragment mass is also called a *transition*.

Each analyte can then be quantified by computing the area under the chromatographic peaks of their XICs.

Data analysis for untargeted approaches is inherently more complex than for targeted quantification. When we already know which analytes we are looking for, there is obviously no need for an identification step. Furthermore, quantification is much easier since we can simply compute the theoretical masses of the analytes of interest (e.g., peptide mass from amino acid sequence, masses of expected b- and y-ions) and then extract and quantify the signal at these positions. For untargeted analysis, on the other hand, we need an algorithm that first detects all signals corresponding to arbitrary analytes present in the sample, i.e., we need to be able to model the signal of a peptide and then try to find all regions in the data that resemble our model. In addition, we need an identification step in order to attribute each of those anonymous quantified features to the analyte that caused it. This thesis mainly addresses approaches for and applications of *untargeted* quantification.

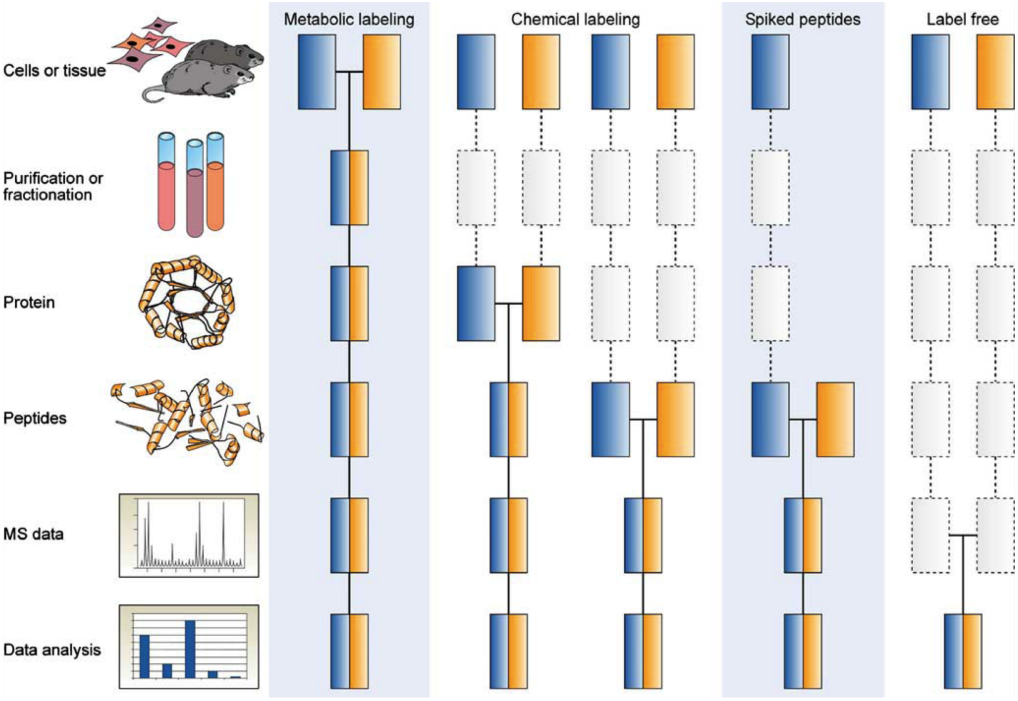

Figure 2.8: Common quantification strategies in mass spectrometry-based proteomics. Boxes in blue and yellow represent two experimental conditions. Horizontal lines indicate when samples are combined. Dashed boxes indicate steps where experimental variation between samples can introduce quantification errors. Reprinted with permission from Bantscheff et al.[61]. Copyright 2007 Springer-Verlag.

Labeled Quantification

Another important distinction to be made is between labeled and label-free approaches. Figure 2.8 depicts the basic approaches for three widely used labeling strategies: metabolic labeling, chemical labeling, and isotope-labeled spike-in peptides. Common to all of them is the idea that samples, once labeled, can be combined and then measured in one go. This approach is commonly referred to as *multiplexing*. Labeling techniques differ in the number of samples that can be multiplexed[3]. These are also referred to as *channels*. The main advantage of labeled approaches over label-free quantification is the fact that samples of different conditions can be combined early in the protocol, and thus all subsequent sample preparation steps and measurements are performed together, which minimizes technical variation between investigated conditions.

In metabolic labeling, labels are incorporated *in vivo*, i.e., the organism is fed a medium containing stable isotope-labeled amino acids which are then incorporated into all of the organism's proteins as the organism grows. The most common approach for metabolic labeling in proteomics is stable isotope labeling with amino acids in cell culture (SILAC)[62,63]. SILAC medium contains isotope-labeled lysine and arginine, which ensures that all fully tryptic peptides will have exactly one such label, as trypsin cuts after K and R. Now, for two samples of different conditions with different SILAC labels, MS peptide signals will appear in characteristic co-eluting pairs with identical chromatographic elution profile, separated only by a mass shift of 8 Da (lysine) or 10 Da (arginine). Thus, the main task in SILAC data analysis is the detection and quantification of these pairs[4] of corresponding peptide signals. Since both conditions have been measured in one and the same instrument run, no further normalization is required; the abundances of two corresponding peptides can be compared immediately. SILAC is mostly restricted to cell culture. An alternative better suitable for higher multi-cellular organisms is ^{15}N labeling[64], where data analysis is similar, but the m/z shift between corresponding peptides is variable due to the varying amount of nitrogen atoms in peptides.

In chemical labeling, labels are applied at a later stage of sample preparation, either to extracted proteins or to digested peptides, using chemical reagents. These approaches can be further subdivided into MS- and MS/MS-based methods. MS-based approaches are similar to the metabolic labeling techniques described above in that they produce pairs (or groups) or co-eluting peptide signals shifted by a certain mass. Therefore, data analysis is essentially identical. Common MS-based methods are isotope-coded afinity tagging (ICAT)[65] and dimethyl labeling[66]. MS/MS-based methods, on the other hand, use a very different approach. Here, different but isobaric[5] labels are applied in each channel. For a single analyte, the differently labeled versions thus all contribute to one and the same MS signal, they are isolated and

[3] 2 - 10
[4] or singlets, if a peptide is present in only one of the samples
[5] same mass

undergo CID together. Now, the key idea is that the different isobaric labels fragment into different characteristic product ions, the so-called *reporter ions*, which can then be quantified in the MS/MS spectrum and compared across channels. Popular methods are isobaric tags for relative and absolute quantitation (iTRAQ)[67] and tandem mass tags (TMT)[68].

Stable isotope-labeled spike-in peptides enable absolute quantification of peptides in targeted analyses. Here, isotopologs of the target peptides are added to the sample in a known concentration prior to LC-MS. Absolute quantification can thus be achieved by comparing the signal intensities of the two versions of a peptide, as the concentration of the spiked standard is known. As in SILAC, they will appear in the MS data as a pair of co-eluting peptide features shifted by a known mass difference.[61]

Label-Free Quantification

The main advantage of label-free quantification (LFQ) methods over label-based approaches is the absence of an upper limit on the number of samples that can be compared in LFQ. On the experimental side, label-free approaches have the additional benefit that the sample preparation protocol is simpler and cheaper, as no labeling steps are required. Samples are prepared and measured independently of each other, and correspondence of analyte signals across runs is established solely during data analysis. This approach, however, comes with significant computational challenges, mainly due to the moderate reproducibility of HPLC, which makes it complicated to establish correspondence between signals across the different runs. With the ever-increasing speed and resolution of modern instruments and the consequent massive amounts of raw data, computational data analysis has largely become the limiting factor in LFQ studies.

Label-free quantification methods can be further divided into spectral counting and MS intensity-based approaches. Spectral counting is an MS/MS-based approach where the abundance of a compound is approximated simply by counting the number of MS/MS spectra in which it was identified. The basic idea is that higher-abundant analytes elute for a longer time and are thus more likely to be selected for fragmentation – and hence identified more often – than less-abundant ones. The advantage of spectral counting over MS-based methods is the fact that data analysis scales linearly with the number of samples here. Nevertheless, this is a very crude way of quantifying. While it may be sufficient for an initial discovery of strong quantitative trends in the data, spectral counting has, unsurprisingly, been demonstrated to show relatively poor quantification accuracy[10,69].

In the context of this thesis, we have worked with MS intensity-based approaches for label-free quantification. A basic conceptual workflow for this type of analysis is depicted in Figure 2.9. Starting with the raw data, the workflow can be divided into two initially independent branches: one for peptide identification (using only MS/MS data), and one for quantification

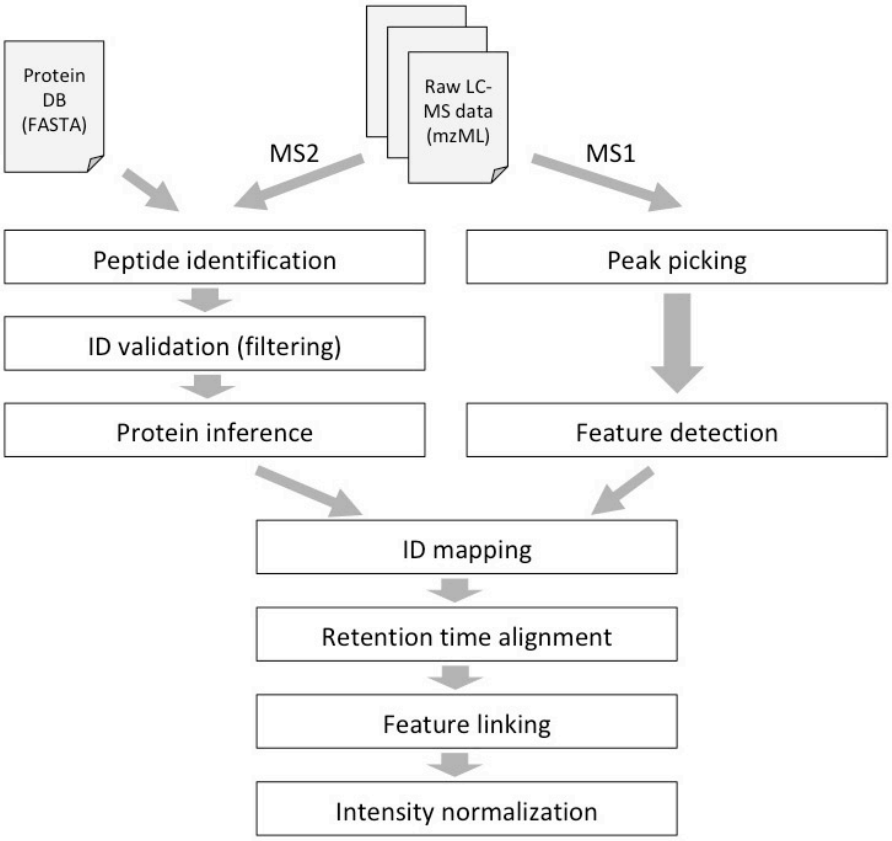

Figure 2.9: Conceptual workflow for label-free quantification.

(using only MS data). The quantitative branch typically starts with peak picking, followed by run-wise feature detection for each LC-MS map. After peptide identification, FDR filtering, and protein inference, peptide IDs are mapped to the quantified features in the ID mapping step. Here, we are combining a set of anonymous peptide features resulting from feature detection, and a set of validated PSMs. Each PSM is annotated with the m/z and RT of the isolated precursor that gave rise to its fragments, and thus can be mapped to the closest compatible[6] quantified feature within a certain m/z and RT tolerance window around the precursor peak. In the retention time alignment step, one tries to find a transformation for each experiment that warps peptide signals in the RT dimension such that the RT difference between corresponding peptides across different experiments is minimized. This facilitates the subsequent feature linking step, where correspondence between signals across different runs is actually established. By doing so, peptide identifications are transferred between corresponding signals. This is a crucial step since many quantified peptide features remain unidentified in a typical experiment. If a peptide can be quantified across several runs, but identified only in a single run, this single identification can now easily be transferred to all linked features in the other experiments. Otherwise, quantified features without identification would remain anonymous and hence useless. Finally, intensity distributions are normalized in order to make them comparable across the different MS runs.

Popular tools for analyzing LFQ data include open-source software like OpenMS[11,12,70], Proteios[71], SuperHirn[72], and msInspect[73]; the free (but closed-source) MaxQuant[74,75]; and commercial applications like Progenesis QI (Nonlinear Dynamics), SIEVE (Thermo Fisher Scientific), and Spectrolyzer (MedicWave AB). They all carry out more or less the same fundamental steps, although the degree of algorithmic emphasis on the different subtasks may vary. A key advantage of the workflow-based OpenMS approach compared to monolithic applications like MaxQuant or Progenesis lies in the flexibility and configurability of the analysis. Because of the high degree of modularity in the OpenMS-based LFQ workflow, the computational analysis is very flexible and can easily be adapted to different study designs and experimental setups. Monolithic applications are typically easier to use but offer only a single pre-determined workflow.

2.2.4 OpenMS – An Open-Source Framework for Mass Spectrometry Data Analysis

OpenMS[11,12,70] is a comprehensive open-source framework for the analysis of LC-MS data in proteomics and metabolomics. It has served as the primary development library for the algorithms and tools developed as part of this work, and has been the most heavily used toolbox for all mass spectrometry data analyses performed in this context. Figure 2.10 illustrates the

[6] same charge

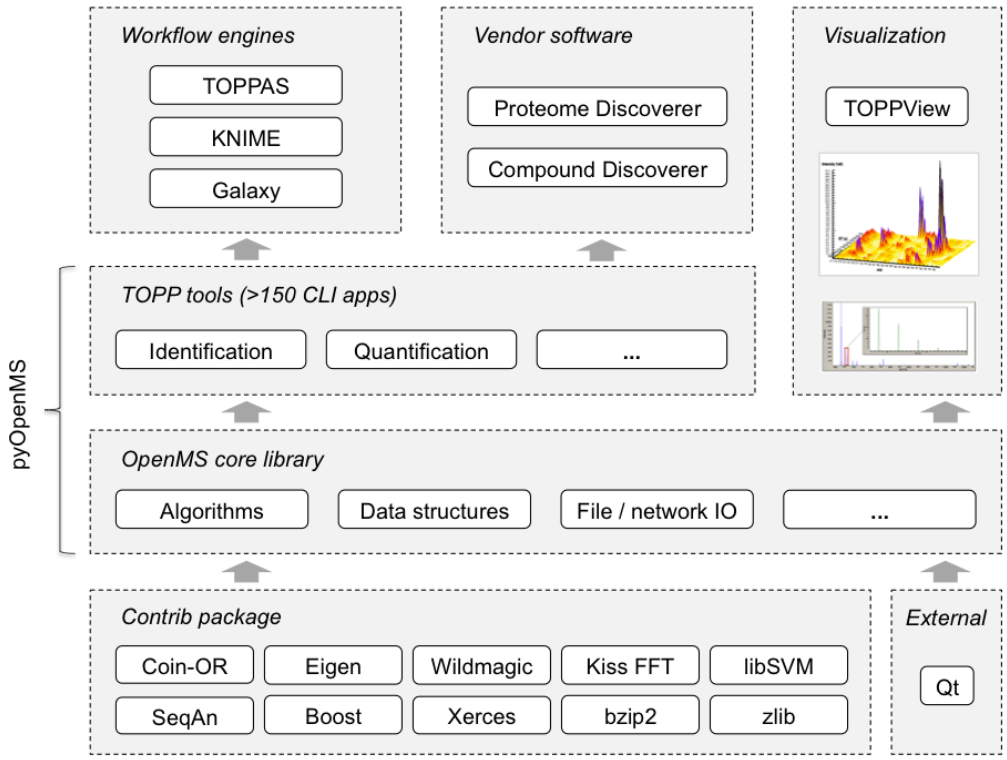

Figure 2.10: Overview of OpenMS architecture and dependencies.

overall architecture of OpenMS. The framework can roughly be subdivided into the following layers:

- **OpenMS core library**: The OpenMS core is written in C++ and provides extensive infrastructure for the development of data analysis algorithms and tools in computational mass spectrometry. It includes numerous data structures for the representation of LC-MS data, classes for reading, writing, and converting between different open mass spectrometry file formats, as well as a multitude of algorithms for common tasks in LC-MS data analysis.[11]

- **The OpenMS Proteomics Pipeline (TOPP)**: Built on top of the core library, the TOPP tool layer comprises more than 180 command line tools, where each is designed to solve a very specific task in LC-MS data analysis. The high degree of modularity and interoperability between different TOPP tools allows the development of custom-tailored data analysis pipelines without requiring programming skills.[12]

- **Python bindings (pyOpenMS)**: Most of the functionality of the OpenMS core and the algorithms implemented as TOPP tools can also be accessed via pyOpenMS. This is useful for rapid prototyping of algorithms and for developers feeling less comfortable with C++.[76]

- **Visualization**: OpenMS is shipped with TOPPView, a powerful GUI application for visualization of mass spectrometry raw data and (intermediate) analysis results.[11,70]

- **Workflow engines**: A more convenient way to leverage the functionality of TOPP is provided by a number of OpenMS-enabled workflow systems. TOPPAS is the in-house OpenMS workflow engine shipped with the software package itself. KNIME is a popular generic workflow system which supports TOPP via plugin and in addition provides powerful tools for downstream data analysis, machine learning, and statistics. For an in-depth discussion of automation and workflows in computational proteomics, see Chapter 3.[13,16,77]

- **Vendor software**: Last but not least, TOPP workflows have even been integrated into popular vendor software tools, such as Proteome Discoverer and Compound Discoverer (Thermo Fisher Scientific). The integration of a label-free quantification workflow and a protein-RNA cross-linking pipeline into Proteome Discoverer will be presented in Section 3.5.[78]

Chapter 3

Automation of Proteomics Workflows

Adapted with permission from

TOPPAS: A Graphical Workflow Editor for the Analysis of High-Throughput Proteomics Data

Johannes Junker+, Chris Bielow+, Andreas Bertsch, Marc Sturm, Knut Reinert, and Oliver Kohlbacher

J Proteome Res. 11(7):3914-20 (2012)

+ These authors contributed equally.

LFQProfiler and RNP^{xl} - Open-Source Tools for Label-Free Quantification and Protein-RNA Cross-Linking Integrated into Proteome Discoverer

Johannes Veit, Timo Sachsenberg, Aleksandar Chernev, Fabian Aicheler, Henning Urlaub, and Oliver Kohlbacher

J Proteome Res. 15(9):3441–48 (2016).

3.1 Introduction

The vast amounts of raw data generated by modern mass spectrometers in the context of today's large-scale proteomics studies necessitate efficient and highly automated tools for data analysis. In addition, many different techniques for quantification and identification and a wealth of different instrument types give rise to a broad range of computational problems. Not surprisingly, bioinformatics and data analysis have turned out to be the bottlenecks and a key research focus in proteomics. Numerous algorithms and software tools have been developed over the past years. There are basically two types of software solutions for MS-based proteomics

data analysis: monolithic applications, usually with graphical user interfaces tailored towards specific applications (identification, quantification), and pipeline-based tool kits.

Examples of the former category include open-source tools like Skyline[79], free but closed-source tools like MaxQuant[74,75] or commercial applications like Progenesis QI (Nonlinear Dynamics). The disadvantage of many of these systems, however, is their lack of flexibility: it is basically impossible to use the software for purposes other than those envisioned by its developers. Hence, adapting the data analysis workflow to even a small change in the experimental workflow can sometimes pose an insurmountable obstacle. Pipeline-based tool kits like the Trans-Proteomic Pipeline (TPP[80]) or The OpenMS Proteomics Pipeline (TOPP[12]), on the other hand, consist of a set of many rather small computational tools which can be flexibly combined to form powerful data analysis workflows. Here, it often suffices to exchange one or two building blocks in order to adapt the data analysis to a change in the experimental workflow. However, they are harder to use and often deployed in large core facilities only. Common to all open source platforms is the support of open standard formats, like mzML[81], mzIdentML[82], or TraML[83], as a way to facilitate tool interoperability.

Scientific workflow management systems, such as Galaxy[84–86], Taverna[87,88], Conveyor[89], Mobyle[90], Pegasus[91], or Kepler[92] can provide a more user-friendly interface to command line-based pipeline tool kits. A proof-of-concept implementation[93] supporting multi-core CPUs (no remote parallelization) integrated parts of the Trans-Proteomic Pipeline[80] and X!Tandem[49] into the Taverna Workbench. However, integrating new tools into these generic workflow systems can be difficult. An alternative specifically tailored to the analysis of HPLC-MS data is Proteomatic[94]. Here, a selection of scripts for analyzing proteomics data is available and can be incorporated into custom workflows using a graphical user interface (GUI). In addition, Proteomatic provides adapters to external tools, such as OMSSA[48].

In the following, we will present solutions for workflow-based automated processing and downstream statistical data analysis based on OpenMS/TOPP: our in-house workflow engine TOPPAS, which is now included in every OpenMS installation, will be covered in Section 3.2. Subsequently, we will present our efforts to make OpenMS tools accessible from within the powerful KNIME workflow and data analysis platform in Section 3.3. An open-source KNIME extension allowing to export KNIME workflows to the Grid and Cloud User Support Environment (gUSE)[18] format will be described in Section 3.4. This closes the gap between the user-friendly KNIME workflow environment and powerful high-performance computing (HPC) resources, which are otherwise only accessible via commercial versions of KNIME. Last but not least, Section 3.5 will cover the tight integration of OpenMS workflows into our latest target platform, the popular vendor software Thermo Proteome Discoverer.

3.2 TOPPAS - The OpenMS Proteomics Pipeline Assistant

In order to provide a user-friendly but yet powerful and productive way of using the OpenMS/TOPP tool kit, we have developed TOPPAS, The OpenMS Proteomics Pipeline Assistant[13]. TOPPAS is a graphical workflow editor and engine fully integrated into the OpenMS/TOPP framework[11,12] which enables fast construction of custom analysis workflows using all the TOPP tools from OpenMS as well as arbitrary external programs like ProteinProphet[14]. TOPPAS also facilitates sharing established workflows by simply sending a single file to a collaborator or through our online repository of shared standard workflows. In this section, we will describe the architecture and features of TOPPAS and showcase its use with several examples, ranging from very simple to rather complex workflows.

TOPPAS is suitable for a wide range of applications without the need to write shell scripts or to do any programming whatsoever. In contrast to generic workflow management systems, setting up and using TOPPAS is straight-forward. TOPPAS is included in version 1.9 or later of OpenMS. Installation takes only a few minutes using readily available binary packages for all major operating systems. TOPPAS has the complete functionality of all TOPP tools available out-of-the-box. This includes a wealth of efficient algorithms for signal processing and preprocessing, peptide property prediction, quantification using different experimental techniques, e.g., SILAC, iTRAQ, and label-free analyses, as well as adapters for identification using several popular search engines, including OMSSA[48], Mascot[46], and X!Tandem[49]. Moreover, almost any other external program can be integrated into TOPPAS by providing a simple configuration file. Sample configuration files for some tools of interest can be found on the OpenMS website.

Workflows can be created and run locally on the user's machine. Alternatively, a command line version of TOPPAS without the graphical user interface enables workflow execution for batch processing of a larger number of data sets. In order to take advantage of modern multi-core CPUs, all processing steps that are independent of each other can be executed in parallel. The user can choose the number of parallel jobs to be executed on the machine. No additional configuration steps are required.

3.2.1 Usage and Features

User Interface

TOPPAS features a user-friendly GUI which allows to create, edit, save, and run workflows. The parameters of all involved tools can be adjusted within the application and are also saved as part of the pipeline definition in a workflow file. Furthermore, TOPPAS interactively performs validity checks during the pipeline editing process and before execution.

Figure 3.1 shows the TOPPAS main window. A simple pipeline is just being created. The user has added several tool nodes to a workflow by dragging them from the TOPP tool list on

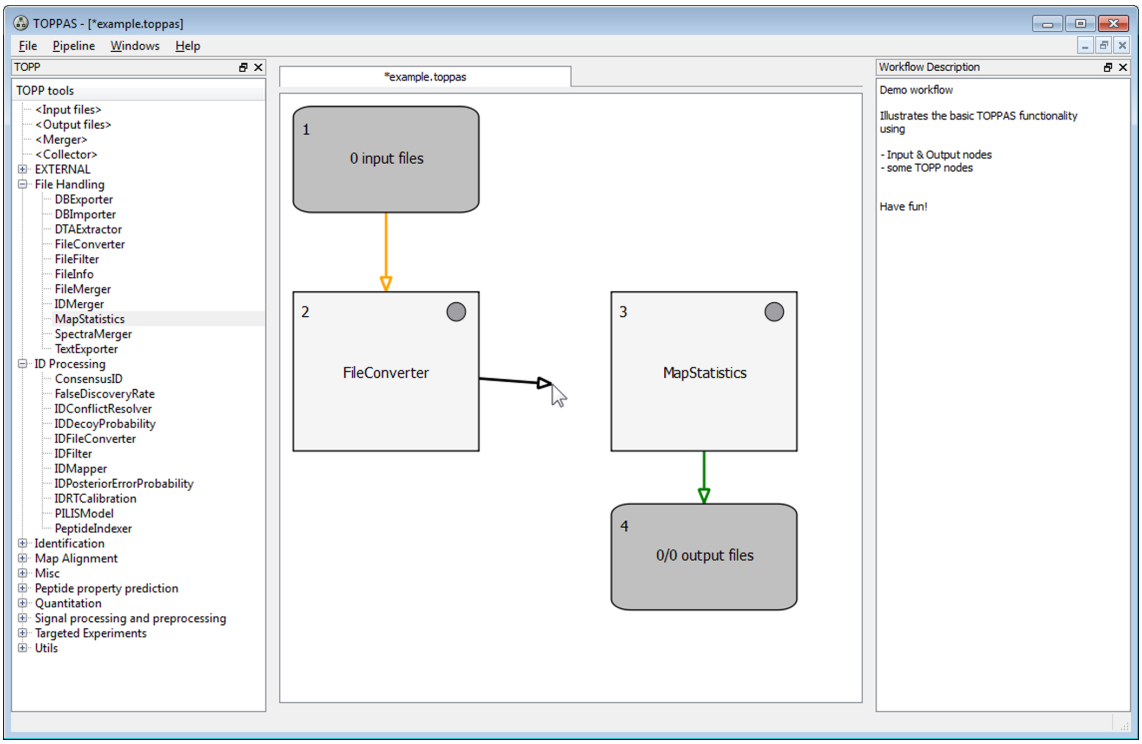

Figure 3.1: Creation of a simple workflow. Tool nodes can be dragged and dropped from the list of tools (left pane) to the workflow canvas (center). Documentation for the whole workflow or individual tools is displayed on the right. Reprinted from Junker et al.[13]

the left to the central area. Additionally, special nodes for input and output files have been added. Edges were drawn between the nodes which determine the data flow of the pipeline. An edge maps an output file of a source node to an input file of the target node. A TOPP node might have more than one input or output file parameter, e.g., the OMSSAAdapter has two input files – an mzML file and a FASTA database file. When an edge is created and either source or target node have more than one input or output parameter, an input/output parameter mapping dialog is displayed to the user to select the output parameter of the source node and the input parameter of the target node. In order to facilitate workflow construction, TOPPAS does not permit to add edges whose source and target file types are not compatible with each other or edges that would lead to a cyclic workflow. Figure 3.2 shows the parameter editing dialog which appears when a tool node is double-clicked.

Once the pipeline is set up, input files have to be specified before it can be run. This is done by double-clicking an input node and selecting the desired input files in the dialog that appears. As soon as a valid set of input files has been selected, the corresponding edge will turn green and the workflow is ready for execution.

During pipeline execution, the circles in the top-right corner of the tools indicate whether a tool has finished successfully (green), is currently running (yellow), has not been executed

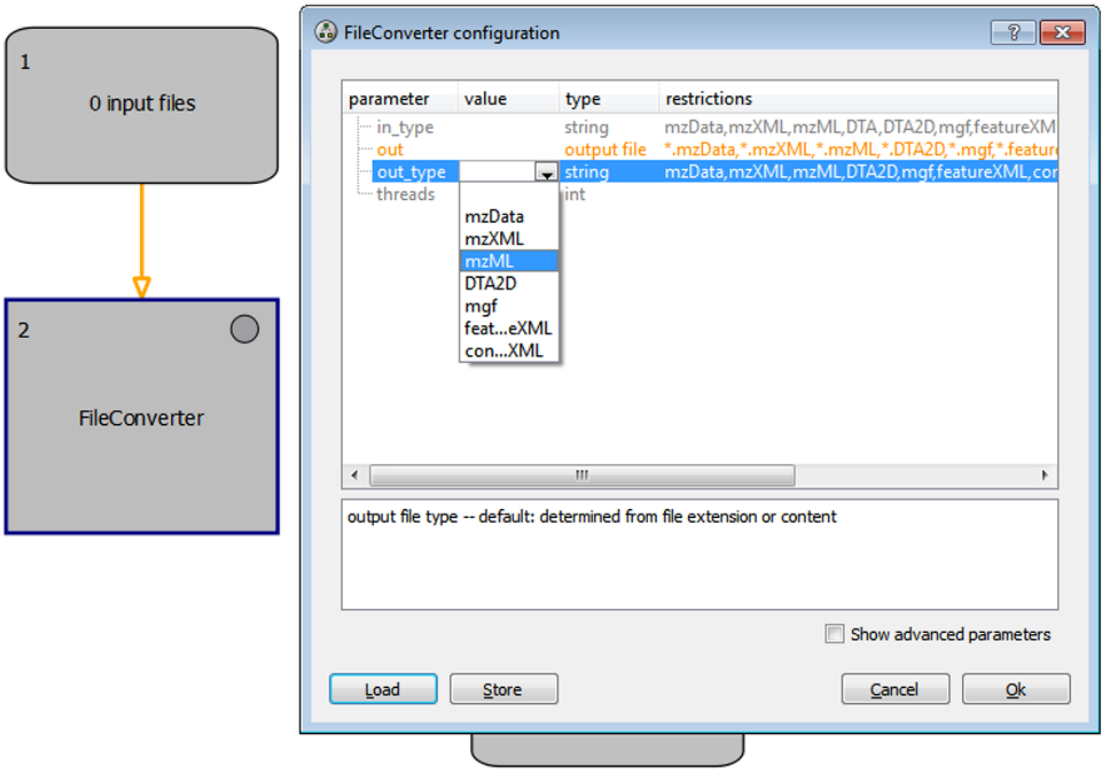

Figure 3.2: Each tool has parameters that can be adjusted through a dialog. Each parameter is explained in the lower part of the dialog, a simple validity check on the parameters is automatically performed. Reprinted from Junker et al.[13]

yet (gray), or could not be executed successfully (red). When the execution has finished, the output files generated by each of the workflow's nodes can be inspected quickly by selecting *Open output in TOPPView* from its context menu.

Workflow Concepts

TOPPAS pipelines follow a round-based concept: in the simplest scenario, the entire workflow is traversed exactly once for each input file. We refer to each traversal as one round of processing. However, only the most basic workflows are strictly linear, meaning that a set of input files is sequentially processed by one or more tools and exactly one output file is produced for each input file. More complex workflows may contain more than one input node. An example pipeline with two different input nodes is illustrated in Figure 3.4.

Even more advanced workflows require results from two or more different processing branches to be merged or certain files to be re-used multiple times (e.g., in identification, when several datasets are searched against one and the same FASTA database). For these purposes, we introduce three additional elements of our workflow language: two special nodes, called *Merge* and *Collect*, which combine the results of multiple incoming workflow branches, and the *Recycle* mode which allows the same file to be re-used over multiple rounds.

A *Merge* node can have arbitrarily many incoming connections from preceding nodes. In each round, it compiles a new list of files consisting of exactly one file per connected predecessor and passes this list of files to its successor node(s). Thus, the lists of files from all preceding nodes must have equal length and they must be in the same order, such that corresponding files are merged together. A *Collect* node behaves similarly but waits for all rounds to finish before passing on a combined list of all output files from all its predecessors. Thus, successors of a *Collect* node will be called only once during the entire pipeline run.

Another useful concept is *Recycling*, where the (output) files of a node can be re-used in multiple rounds. For example, the database input node in Figure 3.3 is set to constantly feed the same FASTA database to the OMSSAAdapter node, which is called three times (once for each mzML input file). In this case, the workflow is valid although its two input nodes contain different numbers of input files, since the FASTA database can be re-used in every round. Without recycling, one would need to specify a list of three identical FASTA files instead.

A use case of both merging and input recycling is illustrated in Figure 3.5. Figure 3.4 demonstrates the usage of a *Collect* node.

External Software Tools

In addition to all TOPP tools, which are included with the OpenMS/TOPP distributions, it is also possible to add custom nodes to a pipeline. These nodes can represent almost any external command line tool, from analysis tools like ProteinProphet to R[95] for statistical data analysis. It

has recently been shown that in some scenarios, heterogeneous workflows incorporating LC-MS analysis tools from different software suites can achieve higher performance than homogeneous workflows[96]. Thus, the ability to also include external tools is highly desirable.

Integrating an external tool into TOPPAS requires a TOPP tool description (TTD) file. This is an XML file specifying the input and output parameters of the tool and how they should be exposed in TOPPAS. For convenience, preconfigured TTD files are available on the OpenMS website (`http://www.openms.org/`) for a number of common tools.

TTD files have a simple structure and the examples given can be easily modified for new tools within a few minutes based on the documentation of the tool.

Once the TTD file is in place, the corresponding node can be found in the EXTERNAL section of the tool menu and used in the same way as any other tool node. For the node to run, the external program has to be installed first. An example TTD file and details on its format can be found in the supplemental material.

Using Preconfigured Workflows

Stable workflows are often re-used by collaborators, maybe in a slightly modified form. Thus, sharing workflows should be as easy as possible. In TOPPAS, the whole pipeline and parameter information is stored in a compact file, which can be distributed conveniently.

As a good starting point, a selection of standard workflows is available on our website in an online repository, which can either be downloaded through a standard web browser or directly from within TOPPAS (see *File* → *Online Repository* in the TOPPAS menu).

Batch Execution of Workflows

Once set up and saved, a workflow can also be run without the GUI using the TOPP tool ExecutePipeline. As input files (e.g., in mzML format) change frequently, the user can also provide a resource file to ExecutePipeline which specifies the input to the pipeline. Pipelines can thus be developed and tested on a desktop machine and then easily deployed in high-throughput environments for automatic processing of larger datasets (e.g., in core facilities).

3.2.2 Application Examples

In order to review the features of TOPPAS, we will describe several examples of varying complexity.

The first example is a basic identification pipeline using the database search engine OMSSA[48]. Figure 3.3 shows the overall layout of the workflow. It accepts one or more mzML files containing the tandem spectra on input node 1. Note that vendor-specific formats, e.g., RAW, can be used after appropriate conversion[15]. On the Windows operating system, this conversion can also be performed within TOPPAS. Input node 2 contains the FASTA database. The

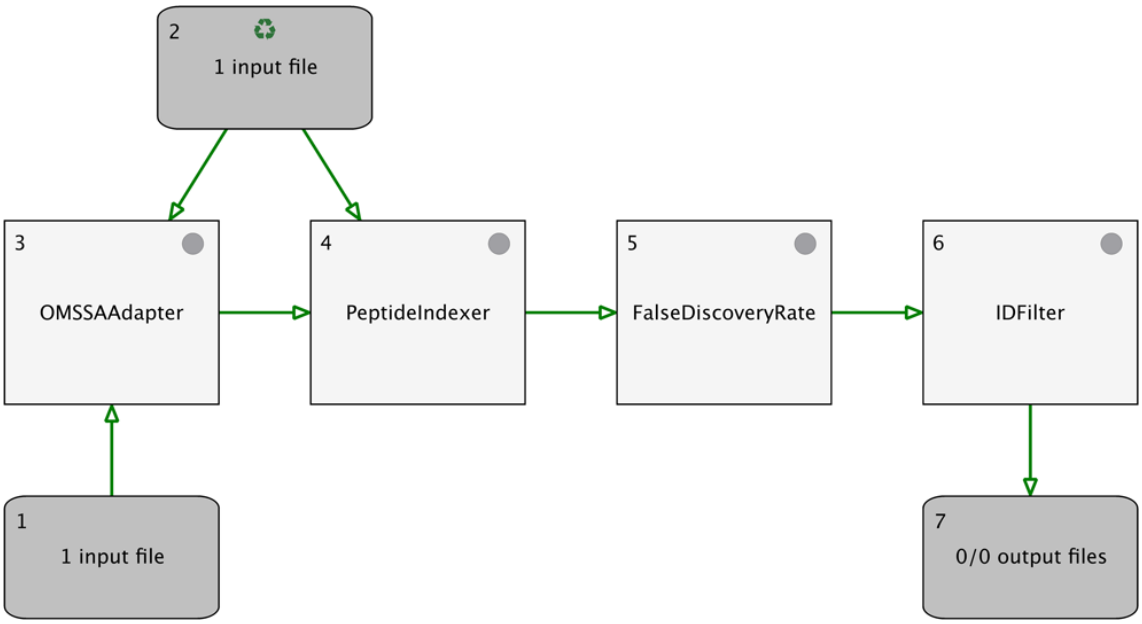

Figure 3.3: Basic identification workflow. Reprinted from Junker et al.[13]

database also contains decoy versions of all protein sequences in order to allow calculation of false discovery rates (FDRs). After identification, PeptideIndexer annotates for each search result whether it originates from the target or from the decoy part of the sequence database. With this information, the FalseDiscoveryRate tool is able to estimate the FDR for each of the peptide-spectrum matches. Finally, the IDFilter is used to retain only those peptide-spectrum matches with an FDR of at least 5%. A possible extension of this pipeline would be to do the spectrum annotation using multiple search engines and combine the results afterwards, using the ConsensusID tool. The results may also be exported using TextExporter for further analysis with external tools, for example Microsoft Excel.

Our second example is the basic label-free quantification pipeline illustrated in Figure 3.4. Input node 1 contains three mzML files. FeatureFinderCentroided finds the peptide features in each of these maps and passes on three featureXML files. Corresponding peptide identifications in idXML format (obtained in advance) are mapped to each of these featureXML files using IDMapper, which then produces one featureXML output file (now including sequence annotations) for every pair of corresponding featureXML and idXML input files. The Collect node waits for all three rounds to finish, then runs FeatureLinkerUnlabeled once, with all three annotated featureXML files as input, which creates a single consensusXML output file.

A more complex pipeline is shown in Figure 3.5. It combines identification using multiple search engines and quantification of iTRAQ reporters. The database used for peptide identification is easy to substitute as it is represented as a dedicated input node in *Recycle* mode.

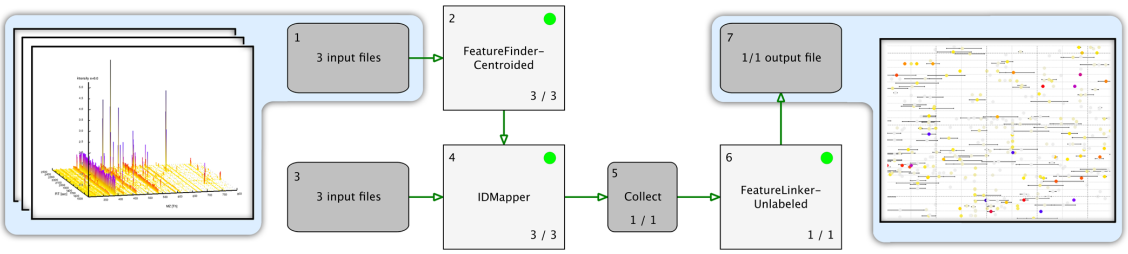

Figure 3.4: This basic label-free quantification workflow is one of the example pipelines included in TOPPAS. It ships with three mzML files containing MS scans of varying concentrations of bovine serum albumin (BSA) as well as three idXML files containing identifications from a search engine run on the corresponding MS/MS data. The context menu of any node can be used to open its output in either TOPPView or a file system browser. Visualizations from TOPPView of the mzML input files as well as the single consensusXML output file are superimposed for illustrative purposes. Reprinted from Junker et al.[13]

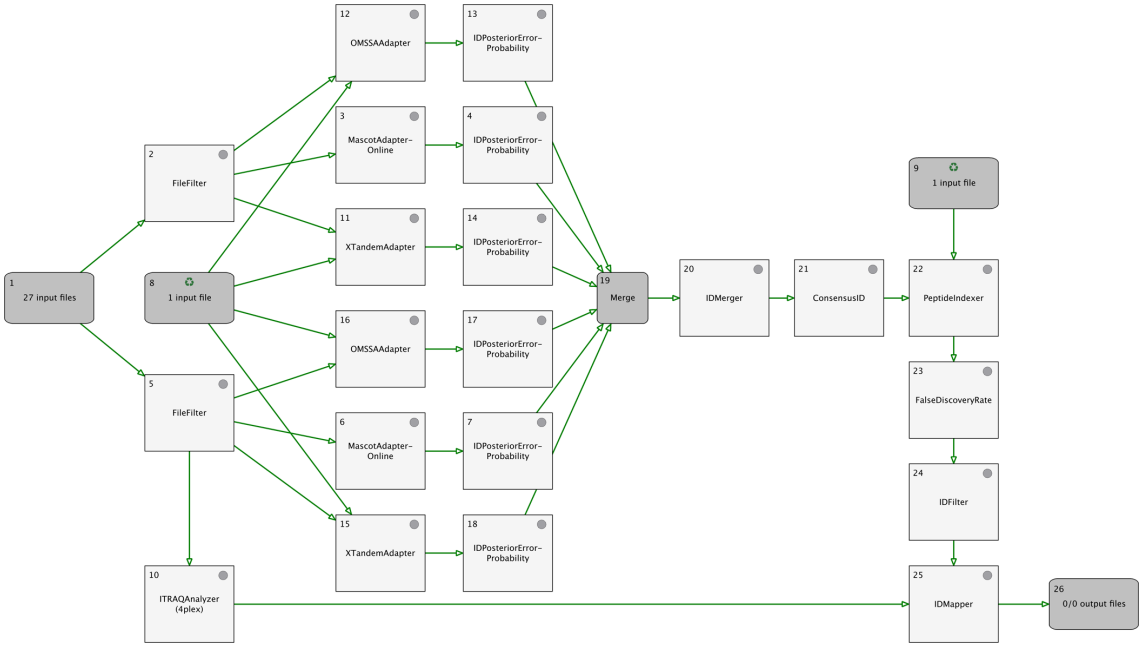

Figure 3.5: iTRAQ identification and quantification pipeline featuring multiple search engines, easy to substitute protein databases, and FDR filtering. Reprinted from Junker et al.[13]

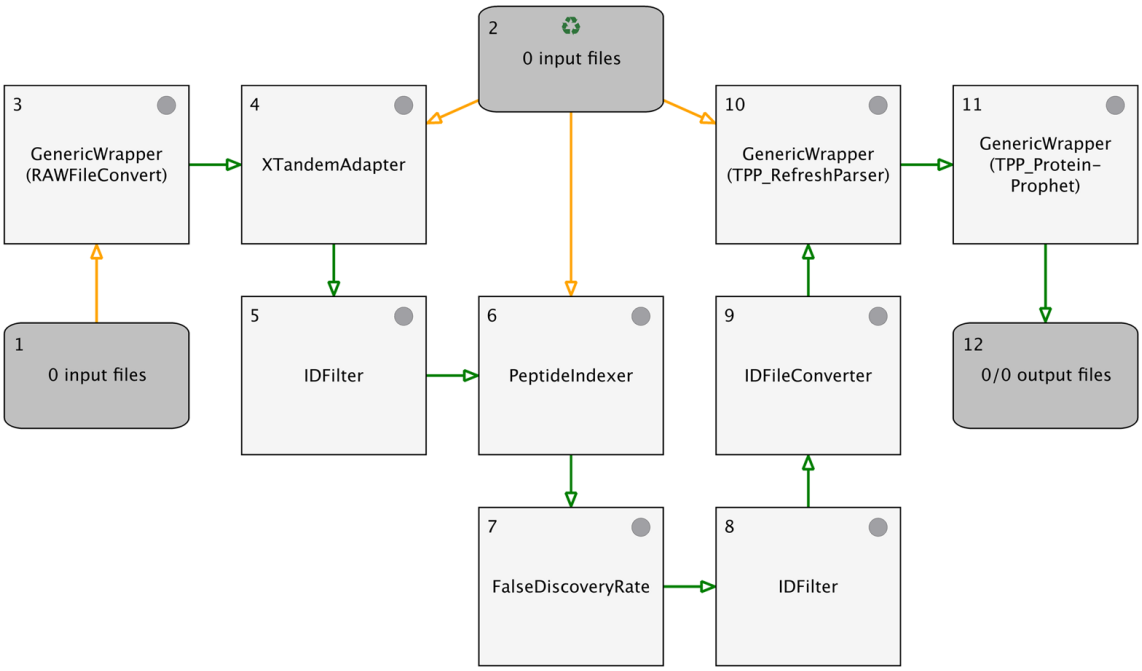

Figure 3.6: Workflow showing the use of external tools: msconvert and ProteinProphet. Reprinted from Junker et al.[13]

To demonstrate the ability of TOPPAS to integrate external software tools, we wrote TTD files for ProteoWizard's msconvert and the widely known ProteinProphet included in the TPP. See Figure 3.6 for an example. The output of the ProteinProphet node is either an Excel-compatible file or an XML file. All TTD shown here described in this publication are included in TOPPAS by default, i.e., these external tools can be used out-of-the-box.

3.3 KNIME Integration

TOPPAS has proven to be an invaluable tool for rapid workflow design, parameter tuning, and automated batch processing of large-scale proteomics analyses using OpenMS/TOPP, and has received lots of positive feedback over the past years. A minor inconvenience, however, is the lack of built-in tools for downstream statistical analyses. A TOPPAS pipeline defines an entire TOPP workflow, the sequence of tools to be run together with the respective input and output files and exact parameter settings for each tool. These workflows usually end with a tabular file (preferably mzTab[97]) containing the "raw" processing results (e.g., peptide and protein identifications, quantified peptide feature intensities across several samples, etc.) and do not include the downstream statistical analyses. These are then usually implemented in an external environment like R[95].

The distinction between efficient data reduction and processing using OpenMS/TOPP on the one hand and downstream statistical analyses on the other hand is somewhat natural: the former steps are often performance-critical and need to be implemented in a very efficient manner. OpenMS/TOPP is written in C++ and contains many highly-optimized algorithms suitable for processing very large datasets fast and using an affordable amount of memory. Furthermore, while many TOPP tools are versatile and reusable in various contexts, the statistical analysis of a study is usually tailored specifically for its particular study design. Often times, it cannot be reused on another dataset. Hence, it is less evident why this part of the data analysis should be implemented using workflows.

There are, however, several advantages of joining data processing and statistical analysis into a single workflow: obviously, such an approach is simply more practical and less error prone, since the processing results do not have to be exported to a file first and then read back into another environment for statistical analysis. More importantly, having a single workflow carry out all analysis steps, from raw data files to publication-ready visualizations and tables, is a great way of ensuring integrity of the results and of documenting the entire data analysis of a study, and thus a big step towards reproducible science. Since the potential wish list for supported downstream data analysis and visualization capabilities is virtually unlimited, implementing and maintaining a satisfactory set of tools would go way beyond the scope of TOPPAS.

This is why Aiche et al.[16] instead opted for developing a plugin for the KNIME[17] workflow system adding so-called Generic KNIME Nodes (GKN) wrapping all OpenMS/TOPP tools and thus making them readily available for building workflows in KNIME including its powerful downstream data analysis capabilities. GKN is a general framework for wrapping arbitrary command line tools and making them available as KNIME nodes. In GKN-based workflows, data is passed between nodes in a file-based manner, just like in TOPPAS, whereas native KNIME nodes pass in-memory tables. Interoperability between GKN-based workflow nodes and native KNIME nodes is established by a selection of adapters for loading processing result files as KNIME tables. Figure 3.7 illustrates an example workflow for label-free quantification of metabolomics data based on a feature detection approach implemented in OpenMS/TOPP[20], together with downstream statistical analysis including intensity normalization, statistical testing for differential abundance, multiple hypothesis correction, plotting of visualizations, and report generation.

3.4 Workflow Conversion

KNIME[17] is a great tool for interactive scientific workflow design and execution, especially due to its integrated reporting and visualization capabilities and the plethora of readily available data analysis nodes. In many cases, there is no need to use any other software: the entire

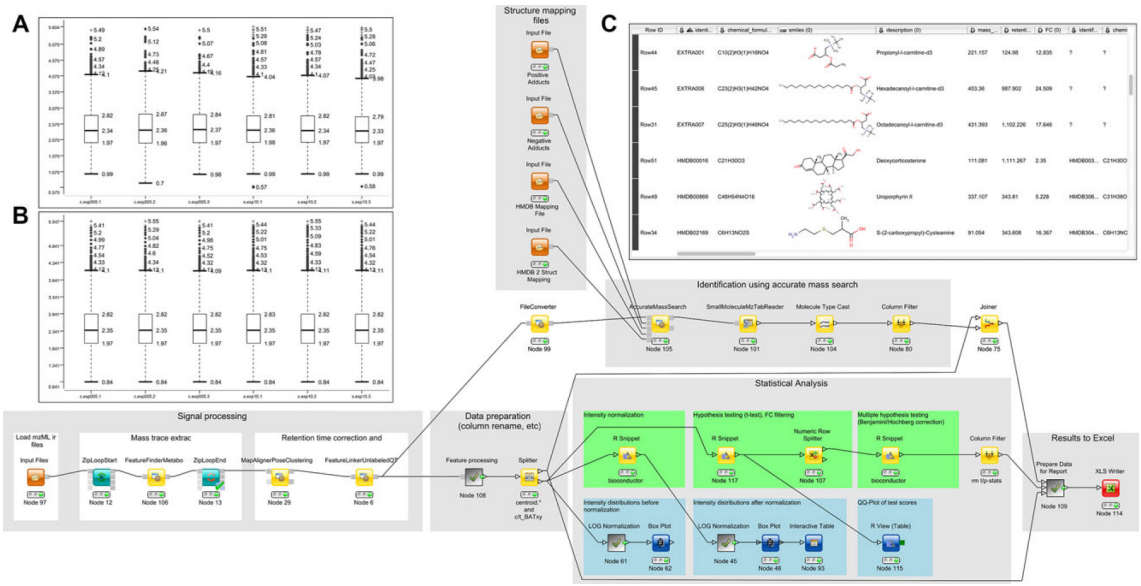

Figure 3.7: Label-free quantification workflow of metabolomics data including raw data processing using OpenMS/TOPP via GKN and downstream statistical analysis and visualization, all within the same KNIME workflow. The insets depict generated visualizations and result tables. A/B: Intensity distributions of invidual samples before/after normalization. C: Summary table including visualizations of chemical structures for all identified and quantified metabolites. Adapted from Aiche et al.[16].

data analysis – from raw data to publication-ready results including figures – can be carried out within the KNIME environment. Some of today's large-scale proteomics studies, however, generate such vast amounts of data that analyzing them on a desktop computer becomes virtually infeasible. In these cases, researchers are forced to resort to high-performance computing (HPC) infrastructures, such as compute clusters, grids, or clouds.

In addition to the free and open-source KNIME Analytics Platform, there are advanced versions of KNIME which do offer support for compute clusters, namely KNIME Cluster Execution and the KNIME Server Edition. However, these solutions are not royalty-free and the support for parallelization on the cluster is somewhat limited. For instance, in GKN-based workflows, the most fundamental way of parallelization, namely the parallel execution of loops over lists of input files, is currently not supported. For an increasing number of today's high-throughput proteomics studies, however, parallel data processing is an absolute necessity due to the sheer amount of acquired raw data.

The Grid and Cloud User Support Environment (gUSE)[18], on the other hand, is a powerful web-based workflow framework designed to run on HPC resources. It is free and open-source and flexible enough to implement workflows utilizing arbitrary tools that run on linux and offer a command line interface (CLI). Workflows can be designed, executed, and monitored through its web-based GUI component, the Web Services Parallel Grid Runtime and Developer

Environment Portal (WS-PGRADE). In principle, it is possible to work with OpenMS workflows using this platform exclusively. One arguable disadvantage of gUSE / WS-PGRADE, however, is the limited usability of its GUI components. Setting up the individual nodes representing tools – and combining them to form entire workflows – can be a very tedious, error-prone, and frustrating process.

Because the workflow languages of TOPPAS, KNIME (using GKN), and gUSE feature comparable concepts, there is a large overlap of workflows that can be implemented in any of the three platforms. This is what motivated us to implement a workflow conversion tool termed KNIME2gUSE which is able to translate arbitrary KNIME workflows to the gUSE workflow language. This includes – but is not limited to – GKN-based OpenMS workflows. Using KNIME2gUSE, users can use the convenient KNIME workflow editor to design and test their workflows and optimize parameters. Once a workflow is set up and tested, converting it to a gUSE workflow is as easy as clicking a button in KNIME. The exported workflow is ready to be executed on a powerful compute cluster or grid running gUSE. Figure 3.8 illustrates a workflow conversion example featuring an OpenMS-based label-free quantification workflow.

3.5 Integration into Thermo Proteome Discoverer

Last but not least, there is still a gap between monolithic GUI applications and proper workflow systems for LC-MS data analysis. As mentioned above, monolithic applications are often too rigid and cannot be adapted to suit the specific needs of its user. Workflow systems, on the other hand, are extremely flexible but this comes at the price of potentially high complexity of the employed workflows. The more explicit a workflow language is, the more powerful and flexible, but also more complex. The Thermo Proteome Discoverer (PD) approach represents a tradeoff between these two extremes: It features a convenient GUI in which users can easily load raw data directly from the instrument and explore and analyze it, all within one and the same platform. Data analysis can be performed using a variety of workflows, and Proteome Discoverer can be extended by plugins, which makes it easy to adapt the software to novel use cases. In comparison to the aforementioned workflow systems, however, the Proteome Discoverer workflow language is somewhat less explicit and thus less complex, as it consists of fewer and larger building blocks which have more built-in logic and sanity checks. Although this takes away some of the power of full-fledged workflow systems, this approach is often a good compromise between user-friendliness and flexibility. We have thus developed so-called meta nodes for the Proteome Discoverer workflow engine, i.e., nodes that contain a more complex workflow under the hood which is hidden from the user for the sake of simplicity and usability.

In Section 3.5, we present a freely available plugin for Proteome Discoverer providing software solutions for two important problems in computational proteomics: LFQProfiler for

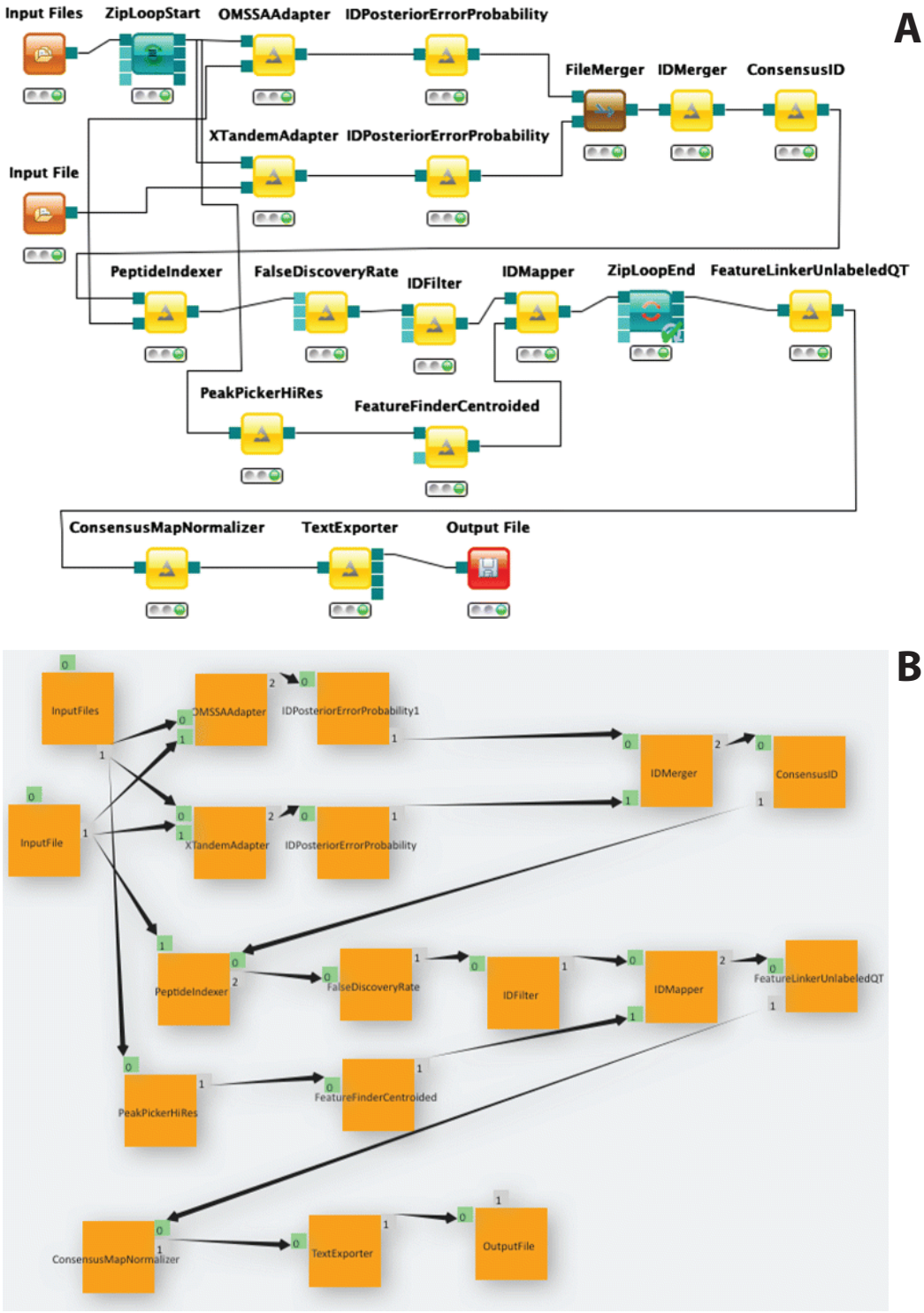

Figure 3.8: KNIME2gUSE workflow conversion example. A: OpenMS-GKN workflow for label-free quantification and peptide identification using multiple search engines implemented in KNIME. B: The same workflow implemented in gUSE, as generated by the KNIME2gUSE workflow converter. Note how there is no one-on-one correspondence of nodes between the two workflow languages. For instance, the ZipLoop constructs that enable sequential processing for lists of input/output files in KNIME are not required in gUSE. Here, iteration over lists of input files is accomplished by setting certain properties in the input / output ports of nodes. The translation between these concepts is done automatically by KNIME2gUSE. Adapted from de la Garza et al.[77].

label-free quantification (LFQ) and RNP^{xl} for protein-RNA cross-linking data analysis. Those two applications were chosen out of the hundreds of other algorithms and tools contained in OpenMS because we felt there was an urgent need for an improved, user-friendly label-free quantification tool on the one hand. On the other hand, we wanted to explore to what extent the tight integration with the raw data visualization could improve the complex annotation and curation still required for protein-RNA cross-linking analysis. These are thus the first two OpenMS tools to be integrated into Proteome Discoverer, but most likely not the last. LFQ is a well-established technique, and a number of commercial and free software solutions exist for analyzing LFQ data (e.g., MaxQuant (MaxLFQ)[74,75], MFPaQ[98], Progenesis QI (Nonlinear Dynamics), SuperHirn[72], or the various feature-finding tools contained in OpenMS / TOPP[11,12]). Until now, Proteome Discoverer did provide only rather limited means to analyze LFQ data: natively, it supports spectral counting, and a rather basic way of MS intensity-based quantification. However, comparing abundances across different samples becomes difficult using this approach, because the crucial step of matching between runs, also known as retention time (RT) alignment or feature linking, is missing. This cannot simply be overcome by adding an additional node for matching between runs, since this would require an algorithm that can detect and quantify peptide features in the data independently of identification results. Such an algorithm is currently missing. Proteome Discoverer can only quantify XICs at those positions where an identification is already present. Since many (especially low-intensity) peptides are identified in only one or few runs, but might be consistently quantifiable across several runs in the MS data, this has a big impact on the number of peptides that can be identified *and* quantified.

A less established but trending topic in proteomics is protein-RNA complex analysis using ultra violet (UV) light induced cross-linking. Protein-RNA complexes are essential components in all life forms. They play pivotal roles in a wide range of biological processes, including bacterial anti-termination, spliceosomal cleavage of intronic regions, small RNA maturation, translational control by miRNA / non-coding RNA, epigenetic modulation, regulation of DNA degradation, etc. In many cases, the structural arrangement of the individual subunits is still unknown and the biological processes these complexes are involved in are poorly understood. Thus, protein-RNA complexes represent a highly interesting target of biological research. RNP^{xl} is a powerful and convenient tool for analyzing protein-RNA cross-linking data. Our new Proteome Discoverer workflow is based on the work of Kramer et al.[99] but contains critical algorithmic improvements compared to the original version and provides convenient built-in visualization within the Proteome Discoverer platform.

3.5.1 Implementation

Thermo Proteome Discoverer is a versatile and user-friendly software for 64-bit Windows platforms enabling proteomics data analyses for a wide range of experimental techniques. It already supports multiple sequence database search engines (e.g., Sequest HT[47] or Mascot[46]), spectral library searching, peptide-spectrum-match validation (e.g., using Percolator[53]), as well as various quantification techniques, such as isobaric mass tagging (iTRAQ[67], TMT[68]) or SILAC[62]. Besides data processing workflows, Proteome Discoverer also offers powerful integrated visualization options including spectrum viewers, scatter charts, histograms, Venn diagrams, and many more.

In PD 2.x, data analysis is workflow-driven. Workflows are always split into two parts: the processing step and the consensus step. The idea behind this distinction is that some computationally expensive tasks have to be run only once (or few times), while other parts further downstream in the analysis might involve more tweaking and optimization and thus have to be run more often using different parameter settings or even different workflows. The results of the processing step (e.g., the sequence database search results) can thus be computed once and various consensus workflows can be tried on them, which usually run much faster. Another intuition for this two-step approach is that tasks in the processing step can usually be computed individually for each input file, one after another, and hence can be parallelized using the batch processing mode, whereas the consensus step requires the processing results of all input files at once for a combined analysis.

Proteome Discoverer is written in the C# programming language and offers an application programming interface (API) for node development enabling the community to write their own PD workflow nodes. Today, a number of PD community nodes is available free of charge, e.g., the popular search engine MS Amanda[100], or the modification site localization tools phosphoRS and ptmRS[101]. A selection of useful nodes can be found on `pd-nodes.org`.

All algorithms utilized by LFQProfiler and RNP[xl] are implemented as standalone executable tools contained in the OpenMS/TOPP tool suite. In order to make these tools and workflows accessible from PD 2.x, we developed a plugin that adds two new processing nodes and two corresponding consensus nodes to the PD node repository. These control the data flow between individual tools and perform conversion of input/output data formats used by PD and OpenMS. This is necessary because OpenMS data storage and exchange is based on XML files, whereas PD uses a relational database approach together with object-relational mapping for storing and accessing all its data. The results of a PD analysis are stored in a pdResult file which contains a SQLite database. From a node developer's point of view, accessing this data from within the node is seamless, since the object-relational mapper (the so-called EntityDataService of PD) takes care of loading objects from and persisting them to the database. From a conceptual point of view, our PD workflow nodes represent so-called meta nodes which encapsulate and

allow execution of larger workflows while hiding complexity from the user. In addition, the plugin implements logic to facilitate usage and provides visualization capabilities.

3.5.2 Results and Discussion

LFQProfiler

An overview of the LFQProfiler workflow is depicted in Figure 3.9. The workflow is split into two parts: The processing step starts with loading raw files using the "Spectrum Files" node. This node is followed by a "Spectrum Selector" for optionally filtering spectra based on various criteria. After that, the workflow branches: peptide features are quantified in all runs by "LFQProfiler FF". In parallel, MS/MS spectra are identified using the native PD node Sequest HT. Subsequently, peptide identifications are validated using Percolator.

As soon as the processing step has finished, the consensus step combines the individual processing results. It starts with the obligatory "MSF Files" node for loading the processing results stored in a Thermo MSF file. "LFQProfiler" then exports peptide identifications from the Proteome Discoverer format to a file in OpenMS's idXML format. Then, for each run, peptide identifications are mapped onto their corresponding quantified features contained in the featureXML files from "LFQProfiler FF". The resulting ID-annotated features are again stored in a featureXML file. At the same time, all peptide-level identification results are used for protein inference using Fido[102] via the TOPP tool FidoAdapter. The result of this step is a set of protein groups that plausibly explain the observed peptides. These will be quantified in the remaining steps.

In order to match peptide signals between runs, chromatographic shifts are first reduced by retention time alignment on the feature level. To this end, RT transformations are computed for each map based on the deviating retention times of corresponding features across different runs. Features are assumed to correspond to one another if they have identical peptide annotations and lie within a (user-defined) m/z and RT tolerance window. The computed transformations are applied to warp the retention times of all peptide signals. This initial warping facilitates the actual task of establishing correspondence between (potentially unidentified) peptide features across runs, the so-called feature linking step, which is achieved using a quality threshold clustering algorithm[19]. Once correspondences are established, the linked peptide signals together with the transferred identifications are stored in a consensusXML file. Feature intensities are normalized across all runs in order to make them comparable.

Both the quantified and identified features from the consensusXML file and the protein inference results in idXML format are used as input for the final protein quantification step. Here, feature intensities are summarized to peptide intensities (i.e., different charge states of the same peptide are merged), and finally, peptide intensities are summarized to protein (group) intensities using the results from the Fido protein inference. The final result are tables

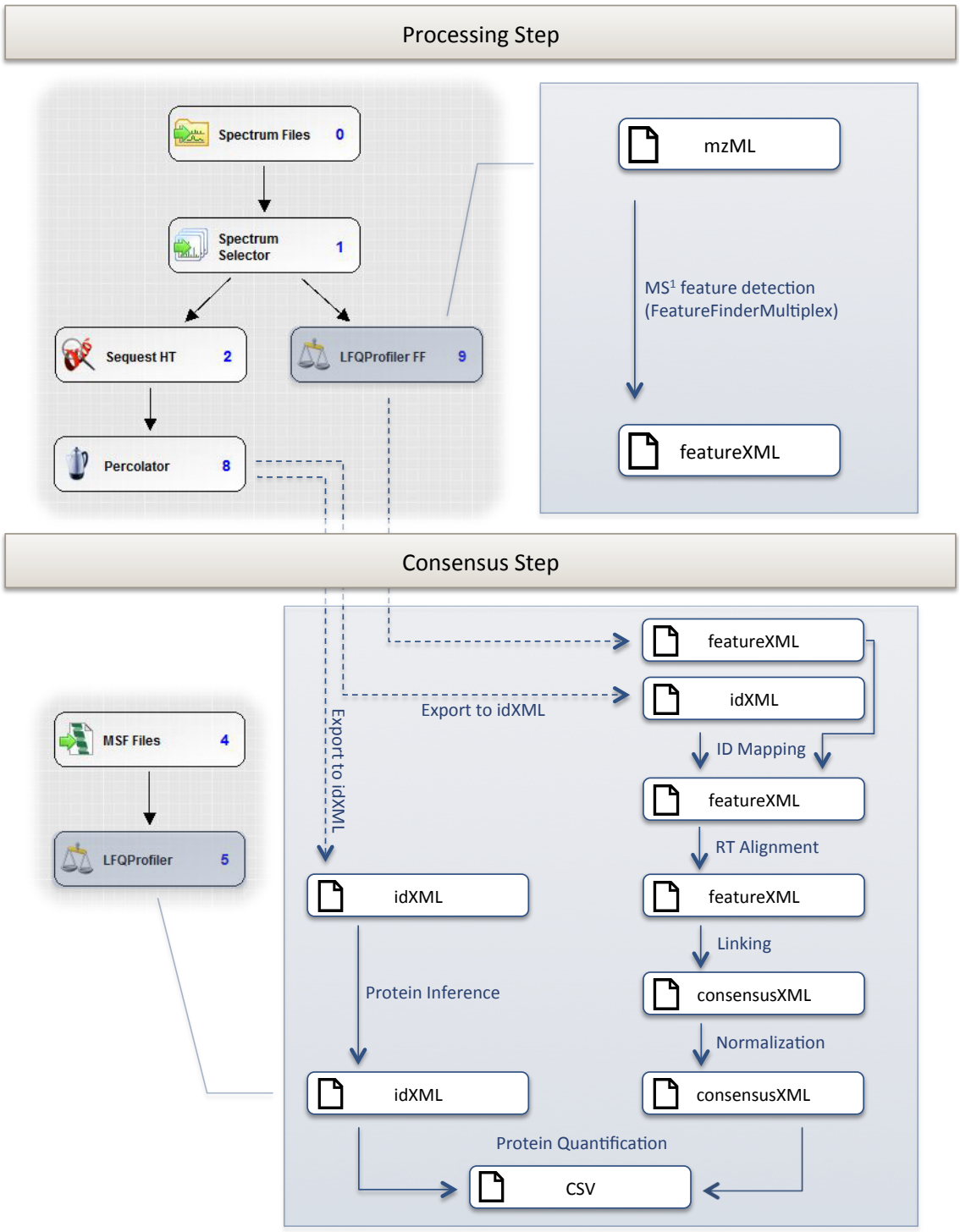

Figure 3.9: Processing and consensus workflow of LFQProfiler. Screenshots from the Proteome Discoverer workflow editor are shown on the left. The two nodes highlighted in blue are the ones provided by our plugin, all others are native PD nodes. The numbers on the nodes are assigned by PD during workflow creation and reflect a topological order of the workflow graph. The inner workings of our nodes are elucidated on the right. Here, boxes correspond to the intermediary files that are produced and arrows represent processing actions carried out by OpenMS/TOPP tools. Reprinted from Veit et al.[78]

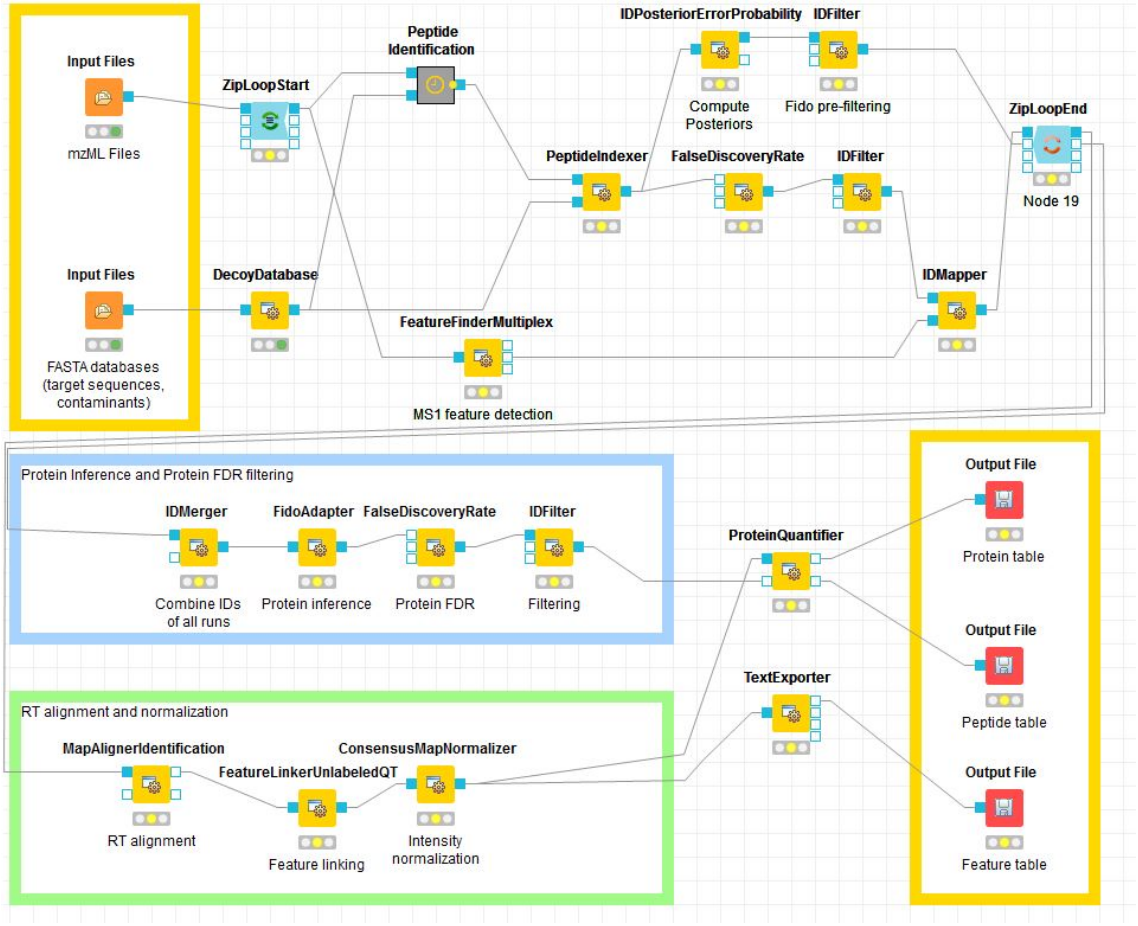

Figure 3.10: The closest equivalent KNIME representation of the wokflow underlying LFQProfiler. Reprinted from Veit et al.[78]

of feature, peptide, and protein (group) abundances for all MS runs in CSV format. These are parsed back into Proteome Discoverers result tables.

LFQProfiler is based on an established workflow for label-free quantification described by Weisser et al.[19] It provides functionality currently missing in Proteome Discoverer and features a number of improvements compared to the original version of the workflow[19], including substantial speed-ups and a greatly reduced memory footprint of several employed algorithms, higher sensitivity, and a more advanced protein inference and quantification step. Figure 3.10 illustrates a KNIME workflow resembling the LFQProfiler workflow as closely as possible. A completely equivalent representation is currently not possible because the search engine and validation nodes (Sequest HT and Percolator) of Protome Discoverer are not available in KNIME. LFQProfiler is fully integrated into Proteome Discoverer and interacts with a selection of the existing Proteome Discoverer data processing nodes.

We have evaluated our method on a recently published publicly available benchmark data set for label-free LC-MS data processing workflows from Ramus et al.[10] They have measured a proteomics standard consisting of an equimolar mix of 48 human proteins (Sigma UPS1) spiked into a complex yeast cell lysate background at nine different concentrations in three replicates. The entire data set thus comprises 27 full LC-MS runs.

The main performance metric is a receiver operating characteristic (ROC) curve of a classifier determinining differential abundance between two conditions with different spike-in concentrations. An ideal classifier would detect all spike-in proteins as differential and all background proteins as non-differential and would thus achieve an area under the curve (AUC) of 1. The ROC is plotted for the combined result containing three comparisons: 50 vs 0.5 fmol/μg, 50 vs 5 fmol/μg, and 25 vs 12.5 fmol/μg (each condition measured in three replicates). For each of the three comparisons, six LC-MS runs (two conditions, three replicates) were processed at once by the different investigated workflows. The results for all three comparisons were then merged into a single table as in the original study for further statistical analysis. The full table of quantified protein groups including statistical test results is available in the Supplementary Material.

We have replicated the exact statistical analysis described in this publication in order to assess the performance of LFQProfiler in comparison with MaxLFQ, MaxQuant, MFPaQ, and Skyline. In order to ensure a fair comparison, we recomputed the performance evaluation metrics for these tools based on the result tables accompanying the publication. The same statistical analysis was then applied to the results of LFQProfiler.

Where possible, we tried to choose settings comparable to the ones used in the other workflows from the benchmark publication, e.g., both the PSM-level and protein-level FDR thresholds were set to 1%. As in the original publication, missing values were imputed on the protein level as the 5-percentile of all protein abundances in the respective LC-MS run. The exact parameter settings of the LFQProfiler workflow are described in Supplementary Table S1. With these settings and after applying all filtering criteria[10], LFQProfiler was able to quantify a total of 2,535 proteins across the combined data set of all three comparisons (MaxLFQ 2,625; MaxQuant Intensity 2,644; MFPaQ 2,721; Skyline 2,620). Figure 3.11 shows the ROCs of the different workflows. Since all investigated software solutions score very high on this data set (AUCs in the 97% - 99% range), we limited the ROC plot to the more informative false positive rate (FPR) interval $[0, 0.05]$ and computed the corresponding relative partial AUCs (pAUC). Note that the relative pAUC of a perfect classifier would equal 100% and correspond to a pAUC of 0.05, which is the maximum pAUC possible for FPRs between 0 and 0.05. A random classifier would have a relative pAUC of 50%, corresponding to a pAUC of 0.025. LFQProfiler achieves a relative pAUC of 97.32%, which is the best performance among the evaluated workflows (MaxLFQ 96.44%; MFPaQ 93.05%; MaxQuant Intensity 91.27%; Skyline 85.70%). The slightly lower number of quantified proteins might be due to a more conservative protein-level FDR

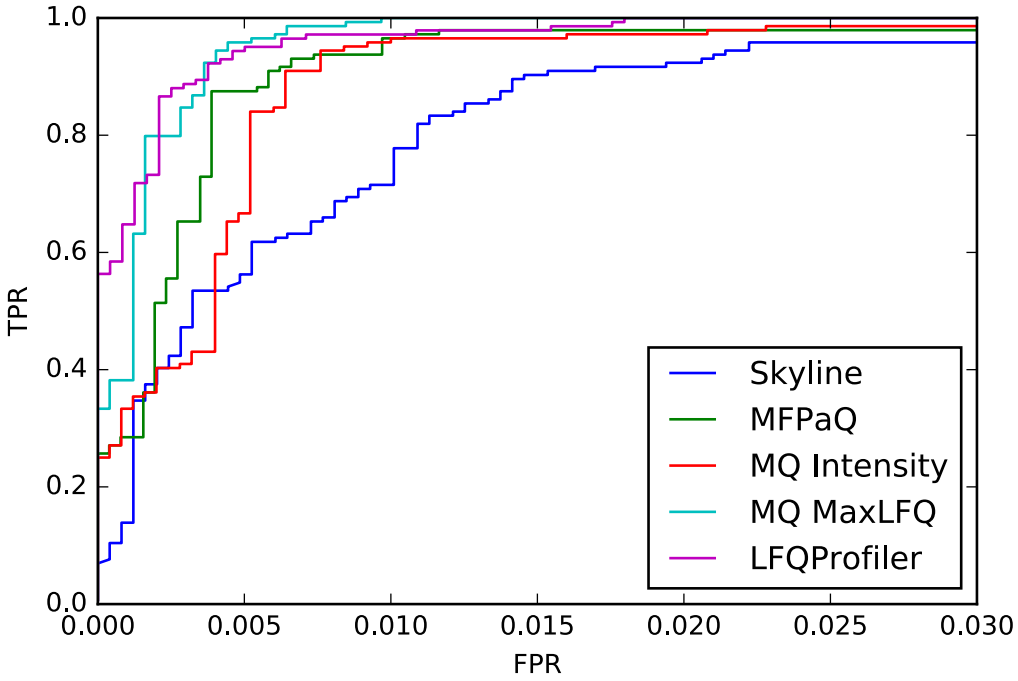

Figure 3.11: ROC curves for the different LFQ workflows participating in the benchmark. True positive rate is plotted against false positive rate for a varying classification threshold (p-value of Welsh's t-test) with a fixed absolute z-score threshold of 1, as described by Ramus et. al[10]. Reprinted from Veit et al.[78]

filtering strategy based on protein inference results in LFQProfiler. Supplementary Figure S1 shows the volcano plot corresponding to these results. Running the entire LFQProfiler workflow on six input files took approximately 2h 20min (1h 47min processing step, 31 min consensus step) using a single core of a 3.20 GHz Intel Core i5-3470 machine with 16 GB of RAM. For comparison, running MaxLFQ on the same machine and input files took 5h 30min.

RNP^{xl}

Beside LFQProfiler, we introduce RNP^{xl}, a Proteome Discoverer workflow based on the work of Kramer et al.[99] for identification and localization of peptide-RNA cross-links which is easy and quick to handle. UV cross-linking of proteins with RNA and identification of the resulting products has been used to assign novel binding regions and exact binding sites in proteins. The large number of potential cross-linked amino acids and oligonucleotides poses a data analysis challenge and has to be accounted for in MS database searching. Kramer et al. introduced an experimental MS workflow together with a processing pipeline implemented in the OpenMS framework, termed RNP^{xl}, to tackle this problem. In essence, the method runs a modified

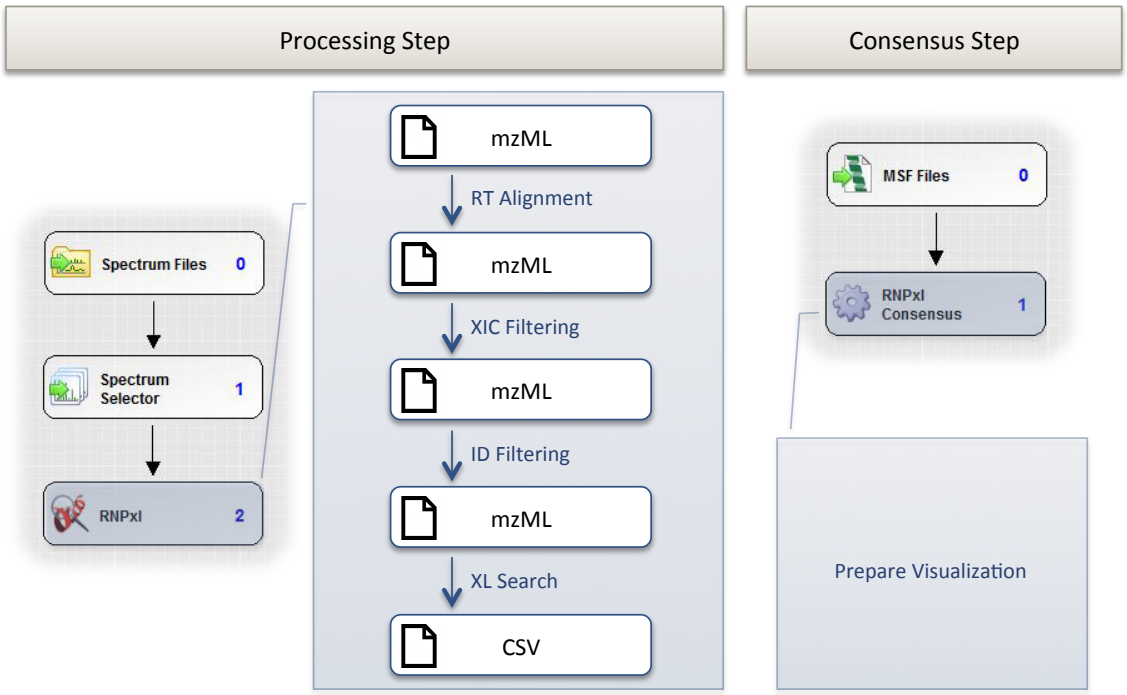

Figure 3.12: Processing and consensus workflow of RNP^{xl}. Screenshots from the Proteome Discoverer workflow editor are shaded in gray. Here, most of the functionality is implemented in the "RNPxl" processing node. If an (optional) control file is provided, the workflow starts with aligning the UV and the control run to each other in the retention time dimension and continues with the XIC filtering step in order to remove signals not originating from cross-links. After that, the ID filtering step removes MS/MS spectra that can be identified as a (non-cross-linked) peptide with high confidence, in order to further reduce the number of false positive cross-link identifications. Finally, cross-links are identified using the dedicated peptide-nucleotide cross-link search engine. The resulting CSV file is parsed back into a Proteome Discoverer result table. In the consensus step, the results are merely preprocessed for visualization. Reprinted from Veit et al.[78]

database search algorithm that is able to identify cross-linked peptides by taking into account the masses of characteristic marker ions, cross-linked immonium ions, and characteristic product ions from the cross-linked peptides. In addition, cross-link identifications are filtered in different ways in order to increase reliability of the results.

A schematic of the RNP^{xl} workflow is shown in Figure 3.12. The "RNPxl" processing node encapsulates two conceptually distinct subworkflows. The first subworkflow can be seen as a computational cross-link enrichment step that aims at removing all tandem mass spectra of non-cross-linked analytes. To this end, all spectra that can be assigned to a (non-cross-linked) peptide, given a user-provided false discovery rate, are removed. If an (optional) non-cross-linked control is provided, the UV and the control file are first aligned in order to correct for chromatographic shifts. Extracted ion chromatograms (XIC) from potential cross-linked precursors are compared between control and UV, and all tandem spectra that show a strong

signal in the control are discarded. The idea behind using a control to filter tandem spectra in the cross-linked sample is that signals in the control are known to not correspond to cross-linked peptides. Hence, they can be used to discard co-eluting precursors and the corresponding tandem mass spectra in the UV file. Now, spectra that originate from non-cross-linked peptides or contaminants have been removed and an enriched set of potential cross-linked tandem mass spectra are used in the second subworkflow performing the actual cross-link search.

The workflow presented here represents both a substantial advancement on the algorithmic side and a great improvement of usability of the original approach[99]. To account for the complex fragmentation behavior of cross-linked moieties, we implemented a novel search engine designed specifically for peptide-RNA cross-link identification. This enhanced workflow runs approximately 90 times faster (on a large Orbitrap XL run with ∼30,000 spectra) than the original approach by Kramer et al.[99] and is able to automatically localize the cross-linked amino acids in the peptide sequence based on characteristic product ions of the cross-linked species. Moreover, localizations are scored using a heuristic, in order to assess the confidence of localizations. Results can be manually inspected for validation using a convenient integrated peptide-RNA fragment spectrum viewer. It thus meets the needs of structural biologists, researchers working in the field of RNA binding proteins as well as proteomic researchers that investigate RNA binding in various cellular contexts.

In addition to the algorithmic improvements, our plugin offers an integrated cross-link spectrum visualization including annotations of peptide ions and cross-linked nucleotide ions, as illustrated in Figure 3.13. This feature substantially facilitates manual validation of the cross-link identifications, which is a crucial step when analyzing these data. For a thorough evaluation of the method and a complete example data set, see Kramer et al.[99] Due to a lack of competing implementations, we cannot give a comparison to other tools in this case.

3.6 Availability

TOPPAS is included in version 1.9 or later of the open source C++ software library OpenMS[11], running on all major platforms (Windows, Linux, Mac OS X). Binary and source packages together with installation instructions are available at `http://www.openms.org/`. GKN and KNIME are available as KNIME extensions through the KNIME Community Contributions repository directly from within the KNIME environment. Installation is as easy as selecting the OpenMS plugin and pressing the Install button in the KNIME GUI. KNIME2gUSE is available at `http://workflowconversion.github.io/` and published under the GNU General Public License (GPL). LFQProfiler and RNP^{xl} binary installers for Proteome Discoverer 2.0 and 2.1, the user manual, as well as example workflows demonstrating the basic usage of our nodes are available free of charge at `http://www.openms.org/pd/`. OpenMS as well as the C# plugins for Proteome Discoverer are open-source, published under a BSD 3-clause license.

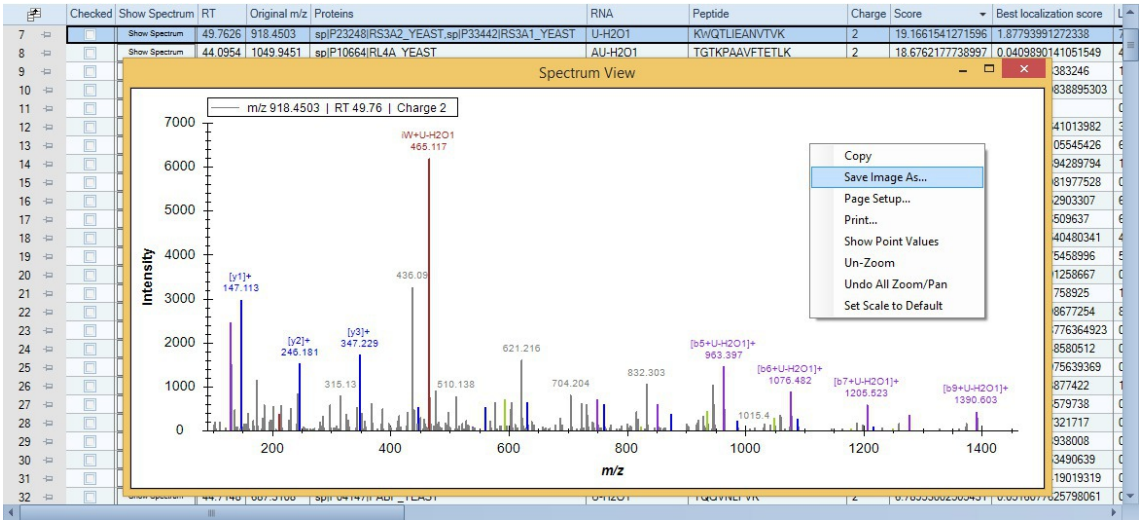

Figure 3.13: Annotation and visualization of cross-links. A tabular view for quick validation of spectrum match information contains protein accessions, precursor charge, m/z and RT, detected cross-linked peptide and RNA, charge, and score. A click on "Show Spectrum" opens an interactive spectrum visualization that highlights all detected fragment ions. In this example, a precursor heteroconjugate with ribosomal peptide and U-H_2O RNA adduct fragmented to y-ions (blue), a-ions (green), cross-linked b-ions (violet) as well as a cross-linked immonium ion of tryptophan (brown). Reprinted from Veit et al.[78]

Source code is available online on GitHub at `http://github.com/OpenMS/OpenMS/` and `http://github.com/OpenMS/PDCommunityNodes/`.

3.7 Conclusion

We have presented a diverse repertoire of available workflow technologies for designing, optimizing, running, and sharing data analysis workflows based on OpenMS/TOPP. Each of those platforms was developed with a specific design goal in mind, and thus comes with its own advantages and disadvantages.

TOPPAS allows non-computer scientists to easily set up new data analysis workflows for mass spectrometric data. It is a valuable tool for designing custom analysis pipelines, while facilitating sharing of existing solutions. Even for bioinformaticians, building a workflow prototype with TOPPAS is much faster and more robust than with custom shell scripts. The entire workflow together with the parameters of all involved tools as well as a workflow description is stored in a single file. It is thus simple to share and document the final pipelines. One of the main advantages over generic workflow management systems is its straight-forward setup and usability. The graphical workflow language is simple enough to be readily used by everyone. Yet, in our experience, it is sufficiently expressive to describe a wide range of MS data analysis workflow. Even complex, branched workflows can be easily modeled and

the interdependencies of the separate branches are resolved correctly, while processing tasks independent of each other can be run in parallel on a multi-core CPU. By default, TOPPAS is equipped with all TOPP tools. These implement a variety of efficient algorithms for numerous tasks in computational analysis of HPLC-MS data. Arbitrary external command line tools can be easily integrated by writing a simple configuration file describing its interface. Established workflows can be run without the GUI using the ExecutePipeline TOPP tool. This enables the use of TOPPAS pipelines in a high-throughput setting, where a visual interface is no longer needed once the pipeline has been tested.

Our experiences with the TOPPAS workflow language have influenced the design of the corresponding language constructs in the Generic KNIME Nodes (GKN) plugin for the KNIME workflow platform. For this reason, any TOPPAS workflow can be represented as an equivalent KNIME-GKN workflow. The main advantage of KNIME over TOPPAS is the huge variety of KNIME nodes for downstream statistical data analysis and visualization. Thus, the entire data analysis workflow – from raw data to publication-ready figures and result tables – can be stored and documented in a single place, thus ensuring the integrity of raw data, intermediate results, analysis results and corresponding figures. Complete data analysis workflows and results can be shared conveniently between researchers. TOPPAS, on the other hand, has better built-in support for parallel execution of pending processing jobs that are independent of each other, and workflow creation is often faster than in KNIME, as KNIME-GKN workflows require a number of explicit control structures that are implicit in TOPPAS. The KNIME2gUSE workflow converter is the latest addition to the repertoire of available workflow technologies for OpenMS/TOPP. It allows to convert KNIME workflows to the gUSE workflow language, and thus enables the seamless deployment of workflows designed and tested in the KNIME environment on powerful compute clusters or cloud environments.

Last, but not least, we have developed user-friendly plugins for Proteome Discoverer adding novel community nodes powered by OpenMS, aiming to combine the convenience of dedicated GUI applications and the flexibility of workflow-driven approaches in a single tool. Our plugins enable two powerful data analysis workflows in PD: LFQProfiler for label-free quantification of peptides and proteins, and RNP^{xl} for peptide-RNA cross-linking data analysis. LFQProfiler is valuable for PD users who want to perform label-free quantification. Until now, the tools for label-free quantification in PD were rather unsatisfactory and limited to spectral counting and calculating the area under the extracted ion chromatogram (XIC) of identified precursor ions. Proper intensity-based label-free quantification, however, requires a number of additional algorithmic steps (e.g., feature detection, mapping of identifications to quantified features, retention time alignment) which are all taken care of by LFQProfiler. LFQProfiler uses a modified version of an established workflow described and benchmarked by Weisser et al.[19]. We have demonstrated its performance and compared it to a selection of other tools in a benchmark setting defined by Ramus et al.[10] We could show that LFQProfiler performs at

least on par with other state-of-the-art tools for label-free quantification. The second workflow, termed RNP^{xl}, represents the first software solution to date for identification of peptide-RNA cross-links including automatic localization of the cross-links at amino acid resolution and localization scoring. Compared to the original version described by Kramer et al.[99], it is substantially faster and more convenient to use, as it is fully integrated into the Proteome Discoverer GUI and comes with a customized interactive peptide-nucleotide cross-link spectrum viewer for convenient manual inspection of the results.

Chapter 4

OptiQuant – A Novel Approach to Label-free Quantification

4.1 Introduction

Relative label-free peptide quantification is the problem of estimating the relative abundances of peptides in multiple LC-MS samples based on their raw LC-MS signals. The main advantage over labeled approaches is that with label-free quantification, the number of samples to analyze and compare is virtually unlimited. While the experimental workflow is straightforward (no labeling required), additional challenges arise on the computational side. Various algorithms and tools for label-free quantification have been proposed in the past. A brief overview of existing solutions can be found in Section 2.2.3. Most of these tools share a certain set of core funtionalities. The task of detecting and quantifying peptide signals in raw LC-MS data inherently requires algorithms for signal detection and data reduction. In fact, data reduction is implied by the problem definition itself: given a set of n raw LC-MS data sets consisting of hundreds of millions of peaks, compute a list of all peptides present in the samples together with their relative abundances in each of the n runs.

In label-free quantification algorithms, data reduction is usually performed multiple times at different levels in order to achieve feasible runtimes for downstream algorithmic steps. Existing solutions differ in the types of employed algorithms and in the order in which data reduction steps are carried out. But data reduction always comes at a price. At each reduction step, the signal is condensed into a more compact, more concrete, and more readily interpretable representation. Ideally, the reduced data set still contains all relevant information required for correct quantification of the original signal, but less irrelevant information. In practice, however, data reduction steps are error-prone. Different instrument types and experimental setups produce data with different noise profiles. Sample complexity and ambient conditions during data acquisition are other important factors contributing to the characteristics of any

given LC-MS dataset. Designing a robust universal data reduction algorithm for LC-MS data is a challenging task mainly due to this multitude of different shapes and sizes.

Traditional approaches to label-free peptide quantification usually perform a significant amount of data reduction at an early stage in the quantification workflow. For each detected peptide feature, its hundreds or thousands of raw data peaks are basically condensed into four numbers: m/z of the monoisotopic mass trace, RT of the chromatographic apex of this trace, overall feature intensity, and charge. This reduction simplifies the subsequent retention time alignment and feature linking problems, but it can also introduce errors that cannot be fixed in the downstream steps anymore. If feature assembly groups the wrong set of signals together in one run, chances are that the respective peptide cannot be linked to its corresponding peptides in other runs as its feature centroid position may not be within a reasonable tolerance window of the true signals, leading to a missing value for this feature.

One possible solution to avoid these kinds of problems is to quantify based on individual detected mass traces rather than assembled features. If followed by an ID mapping step, peptides with identifications matching detected mass traces can be quantified. Instruments try to select the monoisotopic mass of a putative peptide for fragmentation and recording of an MS/MS spectrum. Hence, these identifications should match the monoisotopic mass trace of their corresponding MS signals. A peptide can thus be quantified across all runs in which its monoisotopic mass trace was detected.

Thus, skipping feature assembly as a potential source of error can improve sensitivity. Mere mass trace detection is a less error-prone subproblem of feature detection and errors made here tend to have less catastrophic effects on the overall quantification result. The chromatographic peak estimation of the mass trace may be slightly inaccurate, or the total width of the mass trace may be a bit too long or too short (depending on the noise level and how many missing peaks one wants to allow before canceling the extension of a mass trace in the RT dimension). But in general, the chances of being able to detect and link mass traces across different runs are better than for fully assembled features due to the higher degree of preservation of the true underlying signal.

However, the gain in sensitivity when quantifying based on unassembled mass traces is accompanied by a potential loss of quantification accuracy. A fully assembled feature consisting, for example, of six isotopic mass traces, provides a much better estimate of the true abundance than its monoisotopic mass trace alone. This is especially true for heavier peptides (multiply charged peptides in the higher m/z regions), since their isotope distributions are significantly shifted towards larger numbers of heavy isotopes, and therefore the monoisotopic mass trace accounts for a relatively small proportion of the total signal. Thus, the signal-to-noise ratio of the monoisotopic mass trace is smaller than for the more intense heavier isotopic traces. This can be critical when quantifying low-intensity features close to the noise level. Another shortcoming of purely mass trace-based quantification is the lack of the charge information in

the MS signals. This is because the charge of a peptide can only be determined if at least two consecutive isotopic peaks have been detected, in which case the charge z can be determined based on the m/z distance $\delta_{m/z}$ between consecutive peaks: $z = \left[\frac{1}{\delta_{m/z}}\right]$. Thus, if multiple peptide identifications with conflicting charge states match a single mass trace, it is impossible to choose which one is correct. This problem does not occur with fully assembled features. Here, we can simply select the identification with the charge state matching that of the MS feature. Last but not least, when information about the isotopic envelope is ignored, any mass trace could potentially represent a monoisotopic mass trace of a feature that we want to quantify. We have no way of identifying and then ignoring higher isotopic traces during the ID mapping process. Thus, in complex datasets, the number of candidate mass traces that could be mapped to a given peptide identification is simply too high.

To overcome the shortcomings of traditional approaches in the scenarios described above while achieving the superior quantification accuracy and charge state determination of feature assembly-based approaches, we have developed a novel method termed OptiQuant. It is based on two fundamental ideas: First, we expect that feature detection will be more reliable when considering evidence found across *all* LC-MS runs at once during feature assembly, as opposed to the traditional approach of first detecting feature signals in individual runs and only then establishing correspondence. Instead of assembling features first, our first step consists in run-wise mass trace extraction. Subsequently, corresponding mass traces are linked across runs. Using this intermediate result, we then perform a so-called consensus feature assembly, where consensus features are assembled from linked consensus mass traces. The advantage of this delayed assembly approach is that we can, at this stage, make a more informed guess about the presence or absence of features. By considering the combined evidence across all runs at once, we have a much better chance of choosing the correct feature hypothesis from a set of conflicting hypotheses and of quantifying it across all samples in which the corresponding peptide was present. Moreover, while traditional algorithms usually employ greedy heuristics to select a conflict-free subset from a set of overlapping feature hypotheses, OptiQuant provides a framework for efficiently computing a *globally optimal* solution with respect to an objective function and a set of constraints. We expect a global optimization approach to be superior to greedy local heuristics.

4.2 Concepts

The OptiQuant approach involves three essential steps: mass trace detection, mass trace linking, and consensus mass trace assembly. Figure 4.1 shows a schematic of the basic workflow.

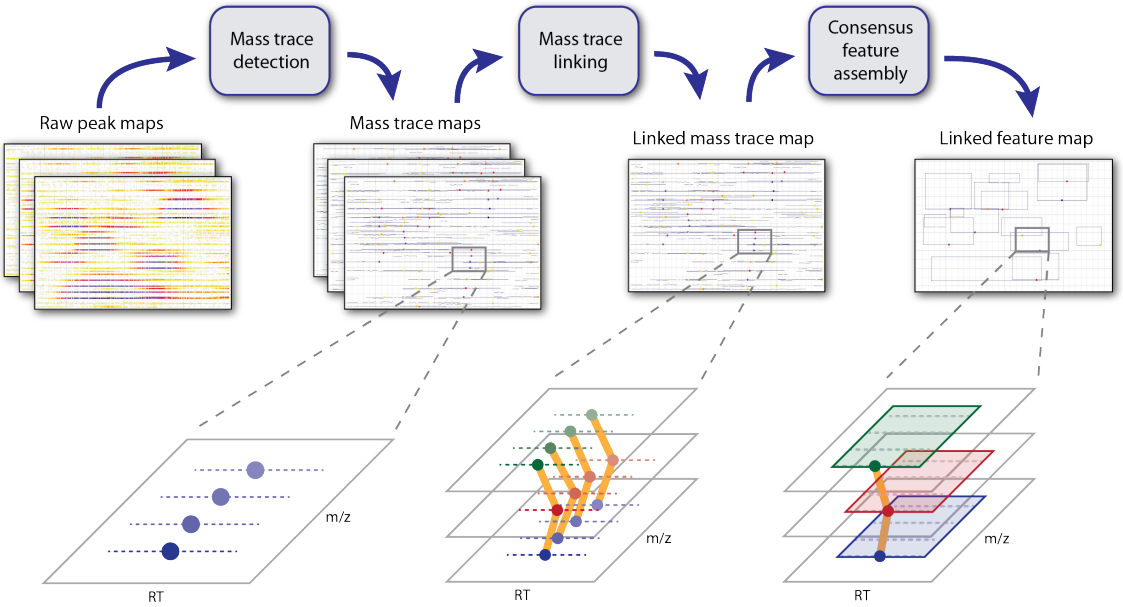

Figure 4.1: Basic OptiQuant workflow.

4.2.1 Mass Trace Detection

Our approach builds upon an established algorithm for mass trace detection, originally published as a submodule of a method for sensitive feature detection in LC-MS-based metabolomics data by Kenar et al. For detailed information on the algorithmic approach, the reader is referred to their original publication[20].

4.2.2 Efficient Mass Trace Alignment and Linking

Mass trace linking is essentially a variant of the feature linking problem in which the objects being linked do not possess a charge state attribute. The charge of a peptide can only be determined if at least two consecutive isotopic traces have been detected, in which case the charge z can be determined based on the m/z distance $\delta_{m/z}$ between consecutive mass traces: $z = \left[\frac{1}{\delta_{m/z}}\right]$. Apart from this difference, existing algorithms for retention time alignment and linking of LC-MS features are readily applicable to mass trace data as well.

A plethora of linking algorithms for LC-MS signals have been developed and published over the past decade. They can roughly be divided into different groups based on the type of input data they operate on (raw data, assembled features), whether they perform pair-wise iterative linking or a global simultaneous approach where all input runs are used at once, and whether or not they perform an alignment or warping step to bring corresponding features into closer proximity to each other before actually establishing correspondence. Smith et al.'s[103] otherwise excellent review mentions OpenMS, but unfortunately does not include the latest

available OpenMS tools for retention time alignment and feature linking. The review refers to an approach by Lange et al.[104] involving the OpenMS tools MapAlignerPoseClustering and FeatureLinkerUnlabeled. MapAlignerPoseClustering selects a user-specified number of high-intensity features in each run as potential anchor points for the alignment and then employs a pose clustering algorithm to estimate matches. Based on matched anchor points, it computes linear transformations for each run to bring its features' RTs closer to those of a reference run. This reference run can be specified by the user or chosen automatically (in which case the map with the largest number of features is selected). The limitation to linear transformations is quite unfavorable, as it has been shown that RT shifts between different runs often show rather strong non-linear effects[19]. Subsequently, FeatureLinkerUnlabeled operates on the transformed feature maps and groups features in a pair-wise greedy manner, loading one map at a time and adding each feature to the closest compatible cluster found so far.

In the meantime, more recent tools for feature alignment and linking have been developed and incorporated into OpenMS, notably MapAlignerIdentification and FeatureLinkerUnlabeledQT[19]. MapAlignerIdentification represents a significant improvement over its predecessor. Most importantly, it allows to compute non-linear transformations using, e.g., cubic B-splines or LOWESS regression. The basic idea, as its name implies, is to exploit peptide identifications annotated to the features in order to determine the initial set of corresponding objects. Granted these identifications are correct, this approach produces a very reliable set of anchor points for computing the warping function. It has been demonstrated to be clearly superior to the linear approach[19]. Obviously, however, this method works only if identifications are available. While this is not a serious restriction in many practical applications in label-free proteomics, where MS/MS spectra are routinely acquired and identified using peptide search engines, it renders the tool useless in situations where identifications are, for whatever reason, unavailable. Conceivable examples include metabolomics or lipidomics datasets where identification of compounds is sometimes less straightforward than with proteomics data. In general, we prefer to consider identification and quantification as two independent problems.

Both MapAlignerPoseClustering and MapAlignerIdentification require a reference run. This is undesirable for a number of reasons. First, it is unclear which of the runs to choose as the reference *a priori*, i.e., without first determining how well a particular run represents the consensus of all maps. In order to guarantee an optimal choice, one would have to try each map as the reference once and then compare certain quality metrics on the final linking results (such as the average RT difference between all features from other maps linked to features from the reference, or the number of linked consensus features without missing values). Such an approach is clearly unfavorable as it causes excessive runtimes. This is why the tools mentioned above just leave the choice of the reference up to the user, or, alternatively simply choose the map with the largest number of features. However, there is no guarantee whatsoever that this is a particularly good choice. This map could just as well be the one with the largest

retention time shifts compared to all other maps. Moreover, the automatic choice implies another potential pitfall: linking results might change considerably if only a single input file is removed or added, if this file just happens to be the map with the largest number of features.

MapAlignerIdentification's counterpart, FeatureLinkerUnlabeledQT, is superior to its predecessor FeatureLinkerUnlabeled as it operates on all maps at once rather than using a pair-wise greedy heuristic approach. The computed result is thus globally optimal with respect to its optimization criteria, regardless of the order of input files and without the need for a reference run. For the development of the OptiQuant label-free quantification approach, we initially chose to work with FeatureLinkerUnlabeledQT. This approach employs a variant of the quality threshold (QT) clustering algorithm[19] for feature grouping. In general terms, clustering is defined as the task of grouping a set of objects in such a way that objects within the same cluster are more similar to each other than to objects of other clusters. In feature linking, our aim is to group signals from different runs originating from the same analyte. The problem can thus be formalized as a clustering problem with additional constraints: objects are represented by their retention time, m/z, and intensity (fully assembled features also have a charge attribute, mass traces do not). A cluster represents a set of corresponding peptide signals from different maps, a so-called consensus feature. The additional constraints are that a cluster cannot exceed a certain diameter (implied by the linking tolerance thresholds for m/z and RT), cannot contain more than one element from each map, and cannot contain features with conflicting charge states or peptide identifications.

FeatureLinkerUnlabeledQT is the current state-of-the art linking tool of the traditional OpenMS label-free quantification workflow. This workflow has been demonstrated to perform at least on par with the most popular competing software solutions for label-free quantification to date[19,78]. FeatureLinkerUnlabeledQT has been a very useful tool during the early prototyping stages of the OptiQuant approach. However, preliminary tests on realistically sized mass trace datasets revealed considerable performance issues. We have carried out a test run using FeatureLinkerUnlabeledQT on a dataset comprising 15 Orbitrap Velos runs with approximately 700,000 detected mass traces (compared to approximately 50,000 detected features) per run. Feature linking was canceled after a total runtime of eight days without making any observable progress.

The problem in this scenario is that mass trace extraction produces much larger result files than full feature detection. Assuming that an average feature has approximately five or six observable isotopic mass traces, the number of datapoints is at least five or six times higher than for feature data. Depending on the mass trace detection parameter settings, false-positive noise traces often add another significant amount of data points. As a consequence, the subsequent retention time alignment and feature linking steps become computationally much more challenging. Thus, a crucial prerequisite for the OptiQuant consensus feature assembly approach was the development of very efficient algorithms for retention alignment and mass

trace linking which are able to operate on hundreds of maps, each containing up to a million mass traces. In the following two sections, we present novel algorithms that are able to achieve this goal.

Retention Time Alignment

The input of our retention time alignment and linking algorithm is a set of featureXML files, each representing an MS run of a label-free experiment. We start by constructing a k-d tree for fast region queries in the RT and m/z dimension. A node in the k-d tree is represented by an object storing the mandatory primary attributes (RT and m/z) as well as the unique index of the feature it represents, in order that the original object a node corresponds to can always be accessed via the node returned by a k-d tree query. For each input map, we create a k-d tree node for each of its features and additionally store the map index of the corresponding input map. Once the entire set of input features has been added, the tree must be balanced in order to optimize query speed.

Next, we construct a compatibility graph on the entire set of features from all maps combined. This graph will allow us to find a reliable set of anchor points for computing the alignment function without the need for a reference run. Let

$$F = \{f_1, \ldots, f_n\} \tag{4.1}$$
$$= \bigcup_{j=1,\ldots,n} M(f_j) \tag{4.2}$$
$$= \bigcup_{k=1,\ldots,m} M_k \tag{4.3}$$

denote the entire set of features (from any input map M_k), where n is the total number of features, m the total number of maps, and $M(\cdot)$ denotes the input map of the feature passed as argument. We define the compatibility graph $G = (V, E)$ with

$$V = \{v_i : i = 1, \ldots, n\} \tag{4.4}$$
$$E = \left\{(v_i, v_j) : c(i, j) = 1\right\}, \tag{4.5}$$

where

$$c(i,j) = \begin{cases} 1, & \text{if } f_i \text{ and } f_j \text{ compatible} \\ 0, & \text{otherwise} \end{cases}$$

denotes the pair-wise compatibility function indicating whether two features are compatible, i.e., potentially representing corresponding signals. To compute the set of features compatible with a feature f_i, we perform a rectangular region query around the feature centroid of f_i using the k-d tree (the rectangle is defined by the user-specified RT and m/z tolerances). The set of features within that region shall be denoted F_i. These are candidates for features compatible with f_i, i.e., potential corresponding signals. We continue by checking additional constraints: Any true corresponding features f_{i_k} must originate from a map $M(f_{i_k}) \neq M(f_i)$ and must have the same charge state z as f_i (if the charge state is known). In addition, we define a maximum absolute log fold change threshold $\lambda_{\max}$ for any pair of features to be considered compatible. This is a measure to increase the specificity of pair-wise correspondence estimation. It is based on the (usually safe) assumption that the dataset contains a sufficiently large number of non-differential signals. Note that this means we might fail to establish compatibility between pairs of highly differential but truly corresponding features. However, at this stage, we are not actually linking features yet, but are only interested in finding a *sufficiently large* number of corresponding features in order to compute retention time alignment functions for all input maps. For this reason, a few missing features do not have a significant effect on the result. The benefits of using this stricter definition of compatibility will be detailed further below. Thus, the actual set $\hat{F}_i$ of compatible features for feature f_i is defined as

$$\hat{F}_i = \left\{ f_{i_k} \in F_i : f_{i_k} \notin M(f_i) \wedge z(f_{i_k}) = z(f_i) \wedge |\log(\frac{f_i}{f_{i_k}})| \leq \lambda_{\max} \right\}. \tag{4.6}$$

Next, we compute the connected components (CCs) on this compatibility graph by means of breadth-first search (BFS) in $O(n)$ time and space. These CCs are preliminary candidates for sets of corresponding features. Ideally, each CC would form exactly a clique (i.e., a subgraph in which each distinct pair of nodes is connected by an edge). In this case, retention time alignment would actually be unnecessary as the feature correspondence problem has a trivial solution. Each CC represents exactly one set of corresponding features.

In practice, however, it is often the case that not all pairs of nodes within a CC are adjacent, for example if the user-specified tolerances are set rather tight and two of the input maps are extreme outliers (in opposite directions) in terms of the overall systemic retention time shift. In this scenario, a pair of true corresponding features from these two maps might not be connected by an edge, but each of them might still be connected to one or more corresponding features from the other maps, which in turn are all closer to the average RT and highly interconnected. This case is still unproblematic, since all true corresponding features would nevertheless end up in the same CC, even though some of the nodes are not connected by a compatibility edge. The situation becomes more difficult when a CC has internal conflicts, i.e., contains several features from the same map or conflicting charge states. This can occur if the m/z and RT tolerance window is too wide, if true positive features from two different CCs are simply very

close to each other, or if additional false positive features are present. As an example, a small subset of the compatibility graph generated for the alignment of a total of 15 label-free LC-MS runs of tryptic human platelet proteome digest in different conditions[105] is illustrated in Figure 4.2.

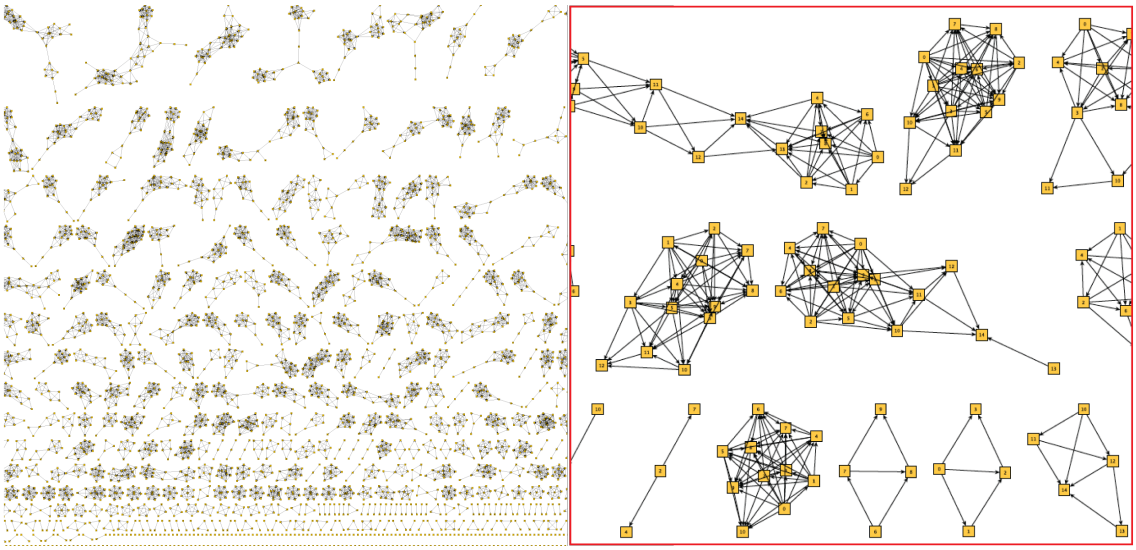

Figure 4.2: Subset of the compatibility graph generated for the alignment of 15 label-free LC-MS runs in organic layout. Figure created using yEd (yWorks GmbH, Germany).

Looking at some of these connected components, one might be tempted to suggest to simply try and resolve the occuring conflicts. For example, some of the CCs consists of two perfect cliques that are joined only by a single edge as a feature from one clique happens to fall into the tolerance window of a feature from another clique (and vice versa). While some of these cases may seem trivial to resolve to human eyes, the general case is in fact computationally very expensive. For instance, one might want to find maximum cliques within CCs that exhibit internal conflicts. It follows from the definition of our set of edges E that a maximum clique can be at most of size m. Hence, our example case could be solved by finding the two maximum cliques contained in this CC. Unfortunately, finding a maximum clique in a graph is an NP-complete problem. Beyond that, it is fixed-parameter intractable and hard to approximate. A slightly more promising idea would be to repeatedly determine the minimum cut of a CC with internal conflicts until all resulting subgraphs are conflict-free. A cut of a graph defines a partitioning of the graph into two disjoint subgraphs. A minimum cut is one which cuts through a minimal number of edges. There exist polynomial time approximations for this problem, such as the randomized Karger-Stein algorithm, which has a “high probability” of finding a min-cut in a graph in $O(n^2 \log^{O(1)} n)$. Such an approach might still be computationally feasible.

However, the fundamental problem with these types of problem definitions is that they cannot deal well with data that contains errors. The presence or absence of nodes and edges

in the graph is a function of the set of detected mass traces (which may cause false-positive and false-negative nodes) *and* the user-specified alignment tolerance parameters controlling the tradeoff between false-positive and false-negative edges. In the general case, the compatibility graph must thus be assumed to contain various types of errors. Even if the result of a graph-theoretical approximation of the abovementioned problems is actually optimal in a graph theoretical sense, this wouldn't guarantee that we found the true solution of the feature correspondence problem. Hence, we might as well resort to simpler, more efficient heuristics for conflict resolution, such as repeatedly deleting edges between nodes with small node degree until the CCs are conflict-free, or something along those lines.

Luckily, however, there is another option. We can simply circumvent conflict resolution altogether, as we don't actually need *all* of the true sets of corresponding features in order to compute the warping functions for all maps. Thus, instead of trying to separate coalescent CCs from one another, we can simply discard problematic CCs. As an additional means to ensure we use only "high quality" connected components for computing the warping function, we define a minimum size (number of maps in which a feature was detected) for the CCs to keep, relative to the total number of input maps. For example, we might want to keep only the set of conflict-free CCs C_i explaining a signal in at least 80% of the input maps $\{C_i : |C_i| \geq 0.8m\}$.

We want to point out that there is virtually no limit on the coalescence of true sets of corresponding features. A single pair of compatible features from different CCs suffices to merge them into one. The degree of connectedness depends to a large extent on the choice of the m/z and RT threshold parameters. Both an extremely small and an overly large tolerance window result in fewer CCs fulfilling the filtering critera: If the window is too narrow, some truly corresponding features might not be joined by an edge and thus might not end up in the same CC. Thus, CCs will be smaller and fewer of them will pass the minimum size threshold. On the other hand, if the tolerance window is too wide, many true sets of corresponding features will merge with others and form larger connected components. However, any connected component with a size larger than m inevitably contains a conflict (at least one non-unique map index) and will be discarded. This undesirable side-effect caused by more generous tolerances is the reason why we introduced $\lambda_{\max}$ in Equation (4.6). It decreases the probability of unwanted coalescence and thus allows to increase alignment tolerances without having to discard too many fused CCs.

Let C denote the set of filtered "high-confidence" CCs. We proceed with computing the actual data points on which we will fit our RT transformation functions. We want to compute one such function for each input map. Therefore, we iterate over all CCs and for each CC C_i, we compute the average retention time $\bar{t}_i$ of all contained features. Then, for each input map M_k we define a corresponding set P_k of pairs of RTs t_{ij} and average RTs $\bar{t}_i$ for all features f_{ij} originating from M_k and belonging to C_i:

$$P_k = \bigcup_{C_i \in C} \left\{ (t_{ij}, \bar{t}_i) : f_{ij} \in C_i \wedge M(f_{ij}) = M_k \right\}. \tag{4.7}$$

Now, we obtain the desired warping function l_k for map M_k by performing a LOWESS regression on the data points contained in the corresponding set P_k. Finally, the RTs t_i of all features $f_i \in M_k$ are transformed by replacing them with the LOWESS evaluation $l(t_i)$. In our case, since we want to keep the original RTs for reporting, we do not actually replace RTs but store an additional $O(n)$ array of transformed RTs for internal use in the subsequent feature linking steps.

The entire retention time alignment algorithm requires $O(n)$ space, since all n features need to be kept in memory at the same time and the additional k-d tree also requires $O(n)$ space. If we would physically generate the actual compatibility graph as described above, storage of the edges would require an additional $O(\frac{n}{m}m^2) = O(nm)$ since there are $O(\frac{n}{m})$ connected components C_i, each of which contains $O(|C_i|^2)$ compatibility edges, where $|C_i| \leq m$. In order to avoid this excessive memory consumption, we compute the CCs of this graph without ever explicitly storing the graph in memory. Instead of traversing all compatibility edges of a node in the BFS algorithm computing the CCs, we simply compute the set of compatible features *on-the-fly* and add them to the current connected component. The runtime complexity is $O(n \log(n))$ on average and $O(n^2)$ in the worst case, since computing the CCs by means of BFS requires $O(n)$ time and for each node, we need to perform a k-d tree region query in order to find compatible features, which requires $O(\log(n))$ on average and $O(n)$ in the worst case.

Feature Linking

Our linking algorithm is based on the same fundamental idea as the FeatureLinkerUnlabeledQT[19] algorithm described above. The basic approach of this greedy QT clustering variant is to select a cluster in each iteration that maximizes a global quality criterion, followed by the removal of the newly clustered features from the set of objects remaining to be grouped. Every feature $f_i \in M_k$ is considered as the center point of a potential cluster C_i containing at most one feature from any map $M_{k'}, k' = 1, \ldots, m$, namely the feature $f_{i'} \in M_{k'}$ most similar to f_i. It follows that these potential clusters are overlapping (sharing features) with other potential clusters in their neighborhood. For this reason, all remaining potential clusters affected by the removal of the features of a finalized optimal cluster must be updated in each iteration. All features $f_{i'} \in C_i$ must lie within the user-specified m/z and RT tolerance window around the center f_i and must have the same charge state as f_i. If no such feature exists, the final consensus feature will have a missing value for map $M_{k'}$. The global quality criterion for choosing an optimal cluster in each iteration is a combination of the similarity of the grouped features and the size of the cluster (i.e., the number of runs in which it explains a signal): The cluster selected in each iteration is always guaranteed to have maximum size among all remaining potential clusters.

Ties are broken by similarity score. It follows that the final number of clusters explaining the entire dataset is minimal. Hence, the algorithm satifisfies the principle of maximum parsimony. The major performance improvement of our algorithm over FeatureLinkerUnlabeledQT is due to the use of considerably more efficient data structures and strategies for organizing and querying the objects to be linked, maintaining the set of potential clusters, efficiently choosing an optimal cluster in each iteration, and updating all remaining potential clusters affected from that choice.

The linking algorithm operates on the transformed RTs from the alignment step (while the original RTs are kept for reporting the final results). Therefore, the original k-d tree must first be rebalanced using the transformed RTs. Now, we start by creating the initial set of potential clusters. To this end, we iterate over all features (from any input map) and for each feature f_i perform a region query of the rectangular region defined by the user-specified RT and m/z tolerance around the feature centroid position. Again, let F denote all features falling within that rectangle. We build the potential cluster C_i by adding the center point f_i itself and selecting for each other map $M_{k'} \neq M(f_i)$ the feature $f_{i'}$ most similar to f_i:

$$C_i = \{f_i\} \cup \bigcup_{k' \neq k} \left\{ \underset{f_{i'} \in F \cap M_{k'}}{\operatorname{argmin}} \, d(f_i, f_{i'}) \right\} \tag{4.8}$$

with the parametrized distance function

$$d(f, f') = \frac{w_{mz} \frac{|\mathrm{mz}(f) - \mathrm{mz}(f')|^a}{\tau_{mz}} + w_{rt} \frac{|\mathrm{rt}(f) - \mathrm{rt}(f')|^b}{\tau_{rt}} + w_{int} \frac{|\log \mathrm{int}(f) - \log \mathrm{int}(f')|^c}{\max_{i=1,\ldots,n} \log \mathrm{int}(f_i)}}{w_{mz} + w_{rt} + w_{int}}, \tag{4.9}$$

where τ_{mz} and τ_{rt} denote the m/z and RT tolerance thresholds, respectively.

Note that the individual components of this distance function can be weighted (thus can be turned off by setting their respective weight to zero) and their exponents can be adjusted. Including the log intensity difference may be counterproductive in some use cases, as we want to be able to link corresponding differential features regardless of their fold change. However, if lots of noisy features are present, using the intensity component can be a very effective measure to avoid grouping low-intensity false-positive features that happen to be close to a cluster center together with true positives.

Now, in order to significantly reduce space complexity, we do not store the entire set of $O(\frac{n}{m})$ potential clusters of size $O(m)$ in memory, but instead keep only so-called *cluster proxy* tuples $C_i^* = (|C_i|, \bar{d}(C_i))$ storing the size of the cluster and the average distance

$$\bar{d}(C_i) = \frac{\sum_{f_{i'} \in C_i} d(f_i, f_{i'})}{|C_i| - 1} \tag{4.10}$$

of all contained points to the cluster center f_i. Thus, our space requirements stay in $O(n)$ rather than $O(nm)$ (which would be quadratic in m). The resulting cost of retrieving the actual cluster C_i again later when its cluster proxy C_i^* is selected as the current optimal cluster is negligible.

In order to efficiently select the current optimal cluster proxy in each iteration, we keep all cluster proxies in a sorted binary tree. The less-than operator $l(\cdot,\cdot)$ defining the actual sorting is defined as follows:

$$l(C_i^*, C_j^*) = \begin{cases} 1, & \text{if } |C_i| > |C_j| \\ 0, & \text{if } |C_i| < |C_j| \\ 1, & \text{if } |C_i| = |C_j| \wedge \bar{d}(C_i) < \bar{d}(C_j) \\ 0, & \text{otherwise} \end{cases}$$

This implies a sorting by priority on the cluster proxies where a cluster of larger size will always be preferred over a smaller one and if the size of two clusters is equal, the one with the smaller average distance $\bar{d}(C_i)$ will be preferred. Since the binary tree is sorted in ascending order, the current optimal cluster proxy is always the first element of the tree and can thus be retrieved in $O(1)$ time. In each iteration, we select the current best cluster proxy C_i^* and now retrieve the actual corresponding cluster C_i a second time, using the same procedure as described above on the center feature f_i. The potential cluster C_i is now finalized. A consensus feature containing all subfeatures $f_{i'} \in C_i$ is added to the final linking result, the $f_{i'}$ are marked assigned so they will not be available to other potential clusters anymore, and the cluster proxy C_i^* is removed from the binary search tree.

Consequently, all remaining potential clusters still grouping any of the now unavailable $f_{i'}$ must be updated before we proceed with the next best cluster. By definition, these features $f_{i'}$ are contained within the rectangular region defined by the m/z and RT tolerance centered on f_i. In the most extreme case, a feature $f_{i'}$ could lie exactly on the border of the tolerance window around f_i, and at the sime time on the border of another potential cluster with center f_j, where f_j is at a distance twice the tolerance in the m/z and/or RT dimension. Hence, we perform a rectangular region query on the k-d tree using twice the specified m/z and RT tolerance to retrieve all features f_j that could possibly be affected by the finalization of f_i and therefore need updating. We recompute the optimal clusters C_j for them (ignoring the now unavailable $f_{i'}$), we remove the outdated cluster proxies C_j^* from the binary search tree and instead add the new cluster proxies $\hat{C}_j^*$ describing the updated clusters C_j. The whole procedure is repeated until the search tree containing the potential cluster proxies is empty and hence all features have been clustered into consensus features.

Like the RT alignment part, the linking part and thus the algorithm as a whole requires $O(n)$ space and $O(n \log n)$ time, since we use the same k-d tree as the primary datastructure (complexity discussed above) and we store a set of $O(n)$ many potential cluster proxies of size

$O(1)$ in a sorted binary search tree, which in turn requires $O(n)$ space and $O(\log n)$ time for insertion and lookup.

Additional Improvement

Although the algorithm is very efficient as is, we have implemented another optimization to even further reduce the runtime and memory requirements in practical application (while not improving on the theoretical complexity). This approach is carried over from an improvement that has been made to FeatureLinkerUnlabeledQT after its initial publication. The idea is to not run the algorithm on the entire dataset at once, but to first partition all input maps in m/z space (using the same boundaries for each map) such that this partitioning is guaranteed not to have an impact on the final results. We partition the dataset D into partitions:

$$\begin{aligned} D &= \{f_1, \ldots, f_n\} && (4.11)\\ &= P_1 \cup \ldots \cup P_m && (4.12)\end{aligned}$$

$$P_i \cap P_j = \emptyset \quad \forall\, i, j = 1, \ldots, m : i \neq j \tag{4.13}$$

where the partitioning fulfills

$$\mathrm{mz}(f) < \mathrm{mz}(f') \quad \forall\, f \in P_i,\, f' \in P_{i+1},\, i = 1, \ldots, m-1, \tag{4.14}$$

and two neighboring m/z partitions P_i and P_{i+1} are seperated by an empty margin in m/z space wider than the user-specified m/z tolerance τ_{mz}:

$$\min_{f' \in P_{i+1}} \mathrm{mz}(f') - \max_{f \in P_i} \mathrm{mz}(f) \;>\; \tau_{mz} \quad \forall\, i = 1, \ldots, m-1. \tag{4.15}$$

This holds because our algorithm guarantees never to link features that have an m/z difference larger than this tolerance τ_{mz}. Note that there is one exception where the partitioning would actually influence our results, namely the computation of the RT warping function. Here, the warping function depends on signals from the entire m/z range and should not be computed on individual partitions. Therefore, we first generate the entire set of datapoints for a global LOWESS fit by accumulating datapoints in an initial full pass over all partitions. Once all datapoints have been collected, a LOWESS transformation is fitted for each input map and this same set of transformations will be applied to each partition in the subsequent steps. With this modification, the partitioning is guaranteed not to have an impact on the results.

4.2.3 Optimal Consensus Feature Assembly Using Mixed Integer Programming

The actual OptiQuant consensus feature assembly algorithm operates on the linked consensus mass traces resulting from run-wise mass trace extraction and subsequent mass trace alignment and linking using the approaches described above. Its output are assembled consensus features corresponding to peptides quantified across the input maps. Again, we employ a k-d tree for fast region queries. This time, the k-d tree is not constructed on the set of input mass traces (subfeatures of the linked consensus mass traces), but only on the consensus mass traces themselves. The m/z and RT of a consensus mass trace is computed as the average m/z and RT of all its linked subtraces.

Now, we generate the set of all possible feature hypotheses H for the user-specified set of charge states Z to consider. To this end, we consider each consensus mass trace as a potential monoisotopic mass trace of a consensus feature. For a given potential monoisotopic consensus mass trace t, we consider all charge states $z \in Z$ and look for up to n_{tmax} (a user-specified parameter) isotopic traces at $\mathrm{rt}(t)$ and $\mathrm{mz}(t) + k\frac{1.0033555\mathrm{Da}}{z}$ for all $k = 1, \ldots, n_{tmax}$ (the constant corresponds to the mass difference between ^{12}C and ^{13}C, see Section 2.2.1). As usual, we allow for a user-specified m/z and RT tolerance around the expected positions of isotopic traces.

In addition to those feature hypotheses for which we have found certain isotopic mass traces at the expected positions, we also consider all hypotheses in which any subset of these traces is missing. The reason for that is that individual mass traces originating from different analytes can easily collide (be observed at the same or a very similar m/z and eluting at the same time) and then cannot be distinguished from one another. In this case, the intensity of the single quantified mass trace is actually the sum of two mass traces intensities. Thus, it wouldn't make much sense to include this mass trace in the quantification of either parent feature (unless we had a way to deconvolve the two true mass traces, which, in the general case, we do not).

In reality, however, we do not actually consider *all* of the possible hypotheses with arbitrary amount of missing mass traces. In order to reduce complexity, and because it is probably safe to assume that some of the more unlikely hypotheses can be neglected (e.g., a hypothesis with only the monoisotopic mass trace, or one with only the monoisotopic mass trace and the sixth isotopic trace, but nothing in between, etc.), we first filter the set of generated hypotheses before selecting a conflict-free subset of hypotheses and promoting them to real features. For details on the filtering criteria, see Table 4.2.

Many of the generated hypotheses are actually in conflict with each other. For instance, all hypotheses sharing the same monoisotopic mass trace are in conflict. We can only select one of them and have to discard all others. The goal of the OptiQuant algorithm is to select the "best" subset of compatible features from each set of conflicting hypotheses. We consider two hypotheses h and h' to be in conflict with each other if they share one or more mass traces.

Most often, a given hypotheses is in conflict with not only one but with a larger number of other hypotheses. Thus, our next step is to determine all clusters of hypotheses $C_j \subseteq H$ for which each $h \in C_j$ is in conflict with at least one other hypothesis $h' \in C_j$ and is in conflict with no other hypothesis $h^* \notin C_j$ outside of the cluster.

Finding these clusters can again be modeled as a graph problem: Let $G = (V, E)$ denote an undirected graph with vertices V representing hypotheses and edges $E = \{(u, v) : h_u \text{ and } h_v \text{ share mass traces}\}$. Then finding the connected components of this graph directly yields the sought conflict clusters. This can be computed in $O(n)$ using breadth-first search (BFS).

The advantage of first finding these clusters is that conflict resolution can now be performed separately for each cluster, ignoring hypotheses from other clusters. This reduces the complexity of the overall optimization problem by reducing a global optimization problem to a set of independent, smaller optimization problems without affecting the optimality of the global solution. In fact, the global solution that we obtain by combining the solutions of the individual problems is identical to the solution of the global optimization problem[1] per definition of the conflict clusters C_j as well as the objective function and constraints (see Equations (4.23) and (4.24) below), which ensure that the solution of the global optimization problem will never contain features involving mass traces belonging to different clusters. The independence of the individual problems allows us to compute their solutions in parallel.

Our algorithm for conflict resolution works as follows: For each hypothesis cluster consisting of n hypotheses, we construct a mixed-integer program (MIP) maximizing an objective function

$$z = s_1 x_1 + s_2 x_2 + \cdots + s_n x_n \tag{4.16}$$

subject to certain constraints (see below), where s_i denotes the score of hypothesis h_i and x_i is a binary variable indicating whether hypothesis h_i is selected as a feature ($x_i = 1$) or discarded ($x_i = 0$).

The score for a hypothesis h_i is defined as

$$s_i = s(h_i) = s_{size}(h_i)^a \, \frac{w_{int}\, s_{int}(h_i)^b + w_{mz}\, s_{mz}(h_i)^c + w_{RT}\, s_{RT}(h_i)^d}{w_{int} + w_{mz} + w_{RT}} \tag{4.17}$$

It is parameterized by the exponents a, b, c, and d as well as the weighting factors w_{mz}, w_{RT}, and w_{int}. The individual score components are the hypothesis size score s_{size}, which equals the number of consensus mass traces the hypothesis explains, the intensity score s_{int} representing the weighted average Pearson correlation of the observed isotope intensity distributions with the expected distribution, as well as s_{mz} and s_{RT} penalizing deviations from the expected m/z and RT in the sub-hypotheses found in individual maps.

[1] The global optimization problem is obtained by maximizing Equation (4.23) s.t. Equation (4.24) for all generated hypotheses at once, without prior grouping into clusters of conflicting hypotheses.

Let h_{ij} denote the j-th subfeature hypothesis (found in input map j out of m maps in total) of a hypothesis h_i. For the user-specified maximum number of considered isotope traces n_{tmax}, let $\text{avg}(h_i) \in \mathbb{R}^{n_{tmax}}$ denote a vector containing the theoretically expected (averagine) intensities of the first n_{tmax} isotopic traces for hypothesis h_i. Let $\text{int}(h_{ij}) \in \mathbb{R}^{n_{tmax}}$ denote the vector of detected trace intensities for hypothesis h_i in map j. Let $n_{ij} = \left|\left\{k = 1, \ldots, n_{tmax} : \text{int}(h_{ij})_k \neq 0\right\}\right|$ denote the number of detected (non-zero) traces supporting hypothesis h_i in map j and let $\hat{n}_i = \sum_{j=1,\ldots,m} n_{ij}$ denote the total number of detected traces supporting h_i across all maps. Then, the overall intensity score component for hypothesis h_i can be defined as

$$s_{int}(h_i) \;=\; \frac{1}{\hat{n}_i} \sum_{j=1,\ldots,m} n_{ij} \frac{r(\text{int}(h_{ij}), \text{avg}(h_i)) + 1}{2}, \tag{4.18}$$

where

$$r(X, Y) \;=\; \frac{\sum_i (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_i (X_i - \bar{X})^2} \sqrt{\sum_i (Y_i - \bar{Y})^2}} \in [-1, 1] \tag{4.19}$$

is the Pearson correlation coefficient. In order to obtain a score $s_{int}(h_i) \in [0, 1]$, the correlations are transformed to this interval by adding 1 and dividing by 2. The intensity score contribution for a map j is weighted by the factor n_{ij} in order to put more emphasis on signals from maps where many traces are explained, as the Pearson correlation is not very meaningful for too small sample sizes. Finally, the sum of scores for all maps is divided by $\hat{n}_i$ in order to constrain the score to $[0, 1]$ again.

For the RT score component, we compute the median absolute deviation (MAD) of the RT values of the consensus mass traces explained by the hypothesis and transform its additive inverse to $[0, 1]$ using the user-specified RT tolerance τ_{RT}. Let $\text{MT}(h_i)$ denote the set of consensus mass traces explained by h_i. Note that, in contrast to above, this is not an n_{tmax}-dimensional vector but a set of mass traces with $|\text{MT}(h_i)| \leq n_{tmax}$, since mass traces can be missing for some of the isotopic masses. For a consensus mass trace $t \in \text{MT}(h_i)$, let $\text{rt}(t)$ denote its RT. Then, with

$$\text{MAD}(X) \;=\; \text{median}\left(\left\{\left|X_j - \text{median}(X)\right|, 0 \leq j \leq |X|\right\}\right), \tag{4.20}$$

the RT score component is defined as

$$s_{RT}(h_i) \;=\; 1 - \frac{\text{MAD}\left(\{\text{rt}(t), t \in \text{MT}(h_i)\}\right)}{\tau_{RT}}. \tag{4.21}$$

Similarly, we compute the m/z variation score. Here, we obviously cannot compute the MAD on the m/z values directly because different isotope traces have different m/z. Instead, we first compute the isotopic mass difference implied by each non-monoisotopic trace based on its m/z value and the position of the trace (an integer between 1 and n_{tmax}) and then compute

the MAD of these mass differences. With $\mathrm{mz}(t)$ denoting the m/z value of a consensus mass trace t, $\mathrm{pos}(t)$ denoting its isotopic position (0 for the monoisotopic trace, 1 for the trace with one heavy isotope, ...), and t_0 denoting the monoisotopic consensus mass trace of h_i, the m/z variation score component is defined as

$$s_{mz}(h_i) = 1 - \frac{\mathrm{MAD}\left(\left\{\frac{\mathrm{mz}(t)-\mathrm{mz}(t_0)}{\mathrm{pos}(t)}, t \in \mathrm{MT}(h_i) : \mathrm{pos}(t) \neq 0\right\}\right)}{\tau_{mz}} \tag{4.22}$$

In order to ensure that only compatible hypotheses are selected in the optimization, we add a set of special ordered set type 1 (SOS1) constraints to the model. Hypotheses are incompatible if they share one or more mass traces. Thus, for every consensus mass trace $t \in T$ (where T is the entire set of detected consensus mass traces), we define the set of hypothesis indices $\mathrm{HI}(t) = \{i : t \in \mathrm{MT}(h_i), i = 1, \ldots, n\}$ of all hypotheses h_i potentially explaining t. For each mass trace, we add an SOS1 constraint enforcing that only one of the hypotheses explaining it can be selected. Thus, the overall optimization problem is defined as

Maximize

$$z = s_1 x_1 + s_2 x_2 + \cdots + s_n x_n \tag{4.23}$$

s.t.

$$\sum_{i \in \mathrm{HI}(t)} x_i \leq 1 \quad \forall t \in T \tag{4.24}$$

After optimization, we then simply select all hypotheses h_i if $x_i = 1$ as features. With that, the description of the basic algorithm is complete. Additional details of our implementation will be described in the following section.

4.3 Implementation

Retention time alignment and feature linking are both performed by the novel tool FeatureLinkerUnlabeledKD. Traditionally, retention time alignment and feature linking have been regarded as two seperate steps in OpenMS workflows and thus implemented in separate, modular tools. While our decision means to break with the paradigm of maximal modularity that we usually follow, we felt that this particular case is an example of over-modularization and the benefits from joining the two steps into one are greater than the disadvantages. If retention time alignment is performed in a separate step prior to feature linking, the intermediate alignment result contains features with transformed RTs and the link to the original features is lost (or at least not trivial to reconstruct). Hence, it becomes difficult to superimpose linking results with the original feature detection results or the raw data, which can be very helpful when

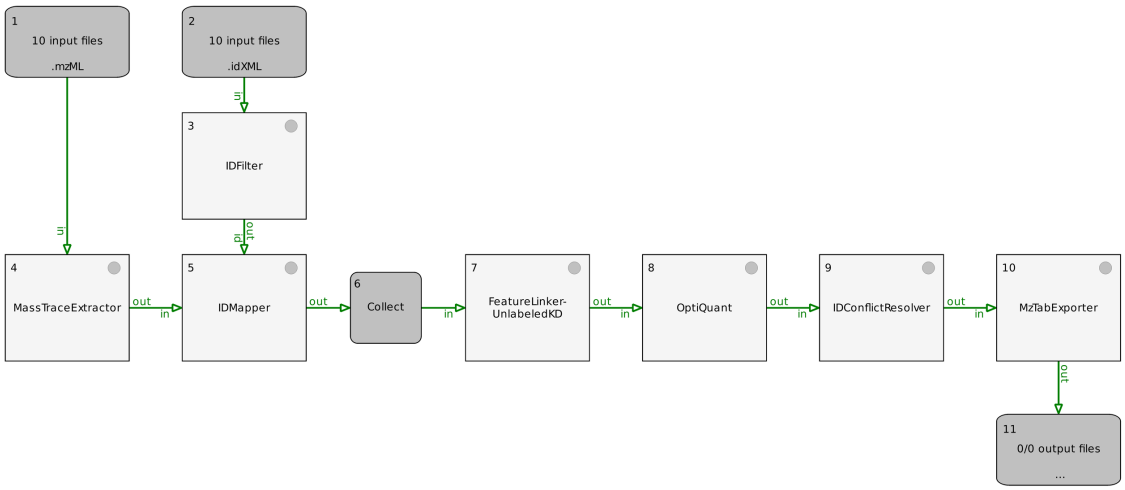

Figure 4.3: Basic OptiQuant TOPPAS workflow featuring the novel tools OptiQuant and FeatureLinkerUnlabeledKD. Input files containing picked MS data in mzML format (node 1) are first processed by MassTraceExtractor, which generates a featureXML file containing the quantified mass traces. In a parallel branch, corresponding identification results are loaded and filtered (we leave out the actual identification sub-workflow for the sake of readability). Subsequently, the filtered peptide identifications in idXML format are mapped to the quantified mass traces in featureXML format using IDMapper. The *Collect* node waits for all preceding nodes to finish (so far, all input files have been processed sequentially), and then calls FeatureLinkerUnlabeledKD on the entire list of the annotated featureXML files. The result is a consensusXML file containing the linked consensus mass traces, serving as input to OptiQuant, which then performs the consensus feature assembly and outputs a consensusXML file containing the final linked consensus features for the entire dataset. Before exporting to mzTab format for downstream data analysis, identification conflicts within consensus features are resolved using IDConflictResolver.

Parameter	Description
mz_unit	unit of m/z tolerance values
nr_partitions	number of partitions in the m/z dimension (more = faster processing times)
warp:enabled	whether or not to transform RTs before linking
warp:rt_tol	RT tolerance during the alignment step
warp:mz_tol	m/z tolerance during the alignment step
warp:max_pairwise_log_fc	maximum absolute log10 fold change between two features in order to be considered compatible in the alignment graph construction
warp:min_rel_cc_size	minimum relative size for connected components to be included in the LOWESS fit (relative to total number of maps)
warp:max_nr_conflicts	maximum number of internal conflicts (features from the same map) for connected components to be included in the LOWESS fit
link:mz_tol	m/z tolerance during the linking step
link:rt_tol	RT tolerance during the linking step
LOWESS:span	fraction of data points to use for each local regression in the LOWESS fit

Table 4.1: Parameters of the FeatureLinkerUnlabeledKD TOPP tool.

troubleshooting an analysis workflow or optimizing workflow parameters. Thus, our algorithm computes and applies RT transformations for internal purposes only. The final result will contain the original retention times for all subfeatures of the linked consensus features, making it easy to investigate corresponding regions in the raw data and original feature detection results in individual runs. Another argument for joining the two steps is simply faster runtime. Disk input/output (I/O) accounts for a significant contribution to the total runtime, and joining the two steps saves half of the runtime spent on I/O. FeatureLinkerUnlabeledKD consumes a set of unaligned featureXML files as input, and outputs a consensusXML file containing linked consensus features with original RTs. After linking individual mass traces using FeatureLinkerUnlabeledKD, the OptiQuant tool runs the consensus mass trace assembly on the consensusXML files containing linked mass traces. Its output is a consensusXML file containing fully assembled and quantified consensus features. The overall TOPPAS workflow of the OptiQuant approach is depicted in Figure 4.3. The parameters of the new tools are listed in Tables 4.1 and 4.2.

4.3.1 Availability

FeatureLinkerUnlabeledKD has become the state-of-the-art feature linking tool of OpenMS. It is contained in the stable release since OpenMS version 2.1, available at `www.openms.org` and `github.com/OpenMS/OpenMS/`. The OptiQuant TOPP tool itself has not been integrated into the mainline of OpenMS yet, due to its dependency on the CPLEX optimization framework, which is incompatible with the OpenMS license. The source code is available in a dedicated

Parameter	Description
mz_tol	m/z tolerance for isotopic trace matching
mz_unit	m/z unit for matching tolerance (ppm or Da)
rt_tol	RT tolerance for isotopic trace matching (sec)
charge_low	the lowest charge state to consider
charge_high	the highest charge state to consider
min_averagine_score	minimum averagine similarity score a hypothesis must achieve in order to be considered in the optimization
max_nr_traces	consider up to this many isotopic traces
require_n_out_of_first_m	do not consider consensus feature hypotheses with fewer than n out of the first m isotopic traces
min_nr_traces_per_map	ignore subfeatures with fewer than this many detected isotopic traces
quantify_top	in the final intensity calculation, consider this this many isotopic traces for quantification
trace_preference	['intensity', 'similarity']; if 'intensity', the <quantify_top>most intense positions of the isotope intensity distribution are selected for quantification; if 'similarity', traces whose intensity profile across maps agree better with each other are selected instead
keep_unassembled_traces	['all', 'none', 'identified']; whether or not to include unassembled traces in the final result; 'identified' includes only unassembled traces with matching peptide identification
adaptive_iso_mass_diff	whether or not to re-estimate the isotopic mass difference while collecting isotopic traces for a hypothesis
require_monoiso	whether or not to require the monoisotopic trace to be present in all maps in order for a hypothesis to be considered in the optimization
use_ids	if a mass trace has identifications attached, generate only hypotheses for the charge state(s) found in these identifications when generating hypotheses for this monoisotopic mass trace
solver_time_limit	CPLEX time limit (sec) for solving the optimization for a single hypothesis cluster
score:size_exp	exponent of the size component in the hypothesis scoring function
score:int_exp	exponent of the intensity component in the hypothesis scoring function
score:int_weight	weighting factor of the intensity component in the hypothesis scoring function
score:mz_exp	exponent of the m/z component in the hypothesis scoring function
score:mz_weight	weighting factor of the m/z component in the hypothesis scoring function
score:rt_exp	exponent of the RT component in the hypothesis scoring function
score:rt_weight	weighting factor of the RT component in the hypothesis scoring function

Table 4.2: Parameters of the OptiQuant TOPP tool.

branch at `github.com/hannesveit/OpenMS/tree/OptiQuant`. This thesis refers to revision 562ab37d9b5fbe6fea4ef7c07672fe684defa51e.

4.4 Benchmarks

We have evaluated the performance of our novel approach on various datasets and compared it to popular state-of-the-art solutions for label-free quantification of proteomics data, namely against the traditional OpenMS LFQ workflow using FeatureFinderCentroided (FFC)[19], the same workflow using the more recent FeatureFinderMultiplex[106] (FFMPX), and MaxQuant (MaxLFQ)[74,75]. All workflows were benchmarked across four different datasets, three of which were synthetic (simulated), and one of which consisted of real experimental data from the iPRG 2015 study[107].

4.4.1 Datasets

Synthetic Data

We have generated three synthetic LFQ benchmark datasets using MSSimulator[108]. Two of them are based on the same ground truth of proteins and differ only in simulated signal quality. These datasets contain peptide features resulting from *in-silico* tryptic digestion and ESI-LC-MS simulation for a set of 200 randomly selected human protein sequences. Protein abundances were drawn from a log-normal distribution with $\mu = 14$ and $\sigma = 4$. Minimum length for tryptic peptides was set to seven amino acids. The resulting ~36,000 features per run span an intensity range of eight orders of magnitude (0.5 to 1.7×10^8) with a mean feature intensity of 3.8×10^6. The simulated m/z range was $[300, 1800]$, the LC gradient had a total duration of 1 h. Each of the two datasets consists of 10 simulated runs, all based on the same ground truth of protein abundances, while retention time shifts and elution peak widths were randomly generated for each individual feature and simulation.

In order to assess the impact of signal quality on feature detection and quantification performance, we have created two different raw datasets for this ground truth: a high-quality one with high resolution (70,000), no mass error, and completely noise-free, and a low-quality one with lower resolution (20,000), simulated m/z error, and intensity noise. For the low-quality dataset, m/z error was drawn from a normal distribution with $\mu = 0$ and $\sigma = 0.001$ for each raw data peak. Detector noise was drawn from a normal distribution with $\mu = -2$ and $\sigma = 5$. This type of noise is added to the intensity of any peak that was actually simulated. In addition, we have generated white noise at any potential peak position (regardless of the presence of a true simulated peak) with intensities drawn from a normal distribution with $\mu = 0$ and $\sigma = 10$.

Since we know the ground truth of these datasets, we can compute exact values for precision and recall of feature detection and compare it across workflows. In addition, these synthetic datasets allow us to assess quantification accuracy, the proximity of the quantified feature intensity values to their true corresponding signal intensities. While protein intensities are log-normally distributed within runs, they have constant intensity across runs. Thus, in an ideal result, any pair of corresponding features from different maps would have an intensity ratio of one, and the standard deviation of subfeature intensities would equal zero for each consensus feature.

However, using only features of constant abundance, we cannot make any statements about the effects of differential abundance on quantification accuracy. The accuracy of relative quantification (ratios) across multiple runs with high dynamic range is a crucial aspect of quantification performance. In order to investigate it, we have created a third synthetic dataset, based on the low-quality dataset described above, containing additional spike-ins (proteins from *Escherichica coli*) in varying concentrations. 50 *E. coli* proteins were randomly selected and added to the existing ground truth (human protein background, constant abundance). Abundances of *E. coli* proteins were constant within and differential across the 10 runs (10^4 — $10^{8.5}$ with 0.5 increment of the exponent). The resulting maps contained ~42,000 features each.

Figure 4.4 gives an impression of the complexity of these datasets. With 42,000 (36,000) features distributed across a 1-h LC gradient, the degree of overlap between features is relatively high. Figure 4.5 shows the low-quality and the high-quality version of the simulated signal for two features in 3D view. Both features were well above the noise threshold, but we can see the lower-abundant isotopic mass traces approach the limit of detection.

iPRG 2015 Challenge

The iPRG 2015 challenge label-free dataset consists of 12 raw LC-MS maps corresponding to four samples acquired in three replicates. Each sample contained a constant background of 200 ng tryptic yeast digest and was additionally spiked with different amounts of six protein digests, as summarized in Table 4.3. The identity of the spike-in proteins was not known to the study participants. Their sequences were added at random locations to the sequence database of yeast proteins provided by the organizers, with background proteins identifiers for camouflage.

LC-MS data was acquired using a Thermo Scientific Easy-nLC 1000 system with a 110-min gradient coupled to a Thermo Scientific Q-Exactive mass spectrometer. Each MS survey scan was followed by 10 MS/MS scans for the most intense precursors using higher energy collision dissociation (HCD). Spectra were acquired in profile mode at resolution 70,000 for MS and 17,500 for MS/MS spectra. The MS scan range was set to 300-1650 m/z.

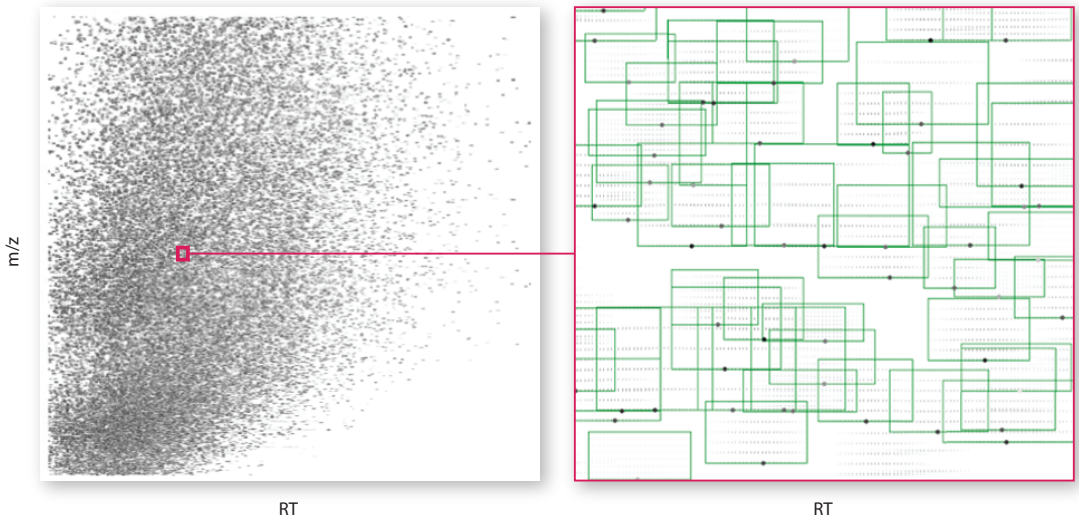

Figure 4.4: Left: overview of a single simulated raw map in 2D view (grayscale indicates log intensity). Right: Small extracted region of raw data with overlaid ground truth features. Large dots indicate feature positions (chromatographic apex of monoisotopic mass trace), green bounding boxes contain all isotopic mass traces corresponding to a feature.

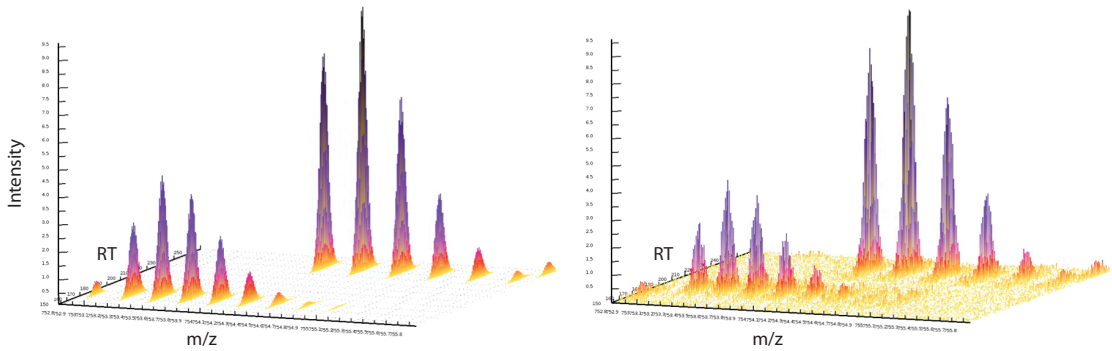

Figure 4.5: 3D visualization of two raw features from the simulated human background. Left: version without noise, high resolution. Right: version with artificial noise, low resolution.

		Samples			
	Protein	1	2	3	4
A	Ovalbumin	65	55	15	2
B	Myoglobin	55	15	2	65
C	Phosphorylase b	15	2	65	55
D	Beta-Galactosidase	2	65	55	15
E	Bovine Serum Albumin	11	0.6	10	500
F	Carbonic Anhydrase	10	500	11	0.6

Table 4.3: Concentrations of the six spike-in proteins (fmol/μg) for each of the four samples prepared for the iPRG 2015 study.[107]

Data was provided to the participants as unpicked raw LC-MS/MS data together with the aforementioned sequence database containing background and spike-in proteins. In addition, the study organizers have included their own identification results that participants could choose to use or disregard. All runs were searched against the provided target-decoy database using the search engines OMSSA[48], MS-GF+[50], and Comet[51]. Search results were individually validated on PSM-level using PeptideProphet[59], iProphet[56] was used to combine the results of the three search engines. Results were originally provided in pepXML format and converted to the OpenMS idXML format for our purposes. The final PSMs handed out to the participants were unfiltered, but annotated by the iProphet probability score, which is equivalent to (1 - posterior error probability). For detailed information on the study design, data acquisition, and search engine parameters, see Choi et al.[107].

4.4.2 Workflows

The quantification performance of the OptiQuant workflow was compared against the performance of three state-of-the-art solutions for label-free quantification (LFQ): the traditional OpenMS LFQ workflow using FeatureFinderCentroided (FFC)[19], a variant using the more recently developed FeatureFinderMultiplex (FFMPX) algorithm[106], and the popular MaxQuant[74] (with MaxLFQ[75] enabled). Figure 4.6 illustrates the TOPPAS workflows for label-free quantification using OptiQuant, FeatureFinderCentroided, and FeatureFinderMultiplex. The latter two workflows are identical except for the feature detection algorithm. For MaxQuant, we do not provide an illustration of the workflow, as MaxQuant is a monolithic GUI application rather than a workflow-driven data analysis tool. It does not offer the modularity and flexibility of OpenMS workflows and supports only a limited set of standard analysis types foreseen by its developers. For more information on the MaxQuant LFQ data analysis strategy and employed algorithms, see Cox and Mann[74] and Cox et al.[75].

Parameter settings for each workflow were adjusted manually for each benchmark dataset. We tried to optimize for the best “overall” result by considering the impact of parameter changes

(and combinations thereof) on all benchmark metrics simultaneously, changing parameters only when this led to a clear improvement and keeping the default values otherwise. Note that a full systematic grid search across the parameter space of all tools in a workflow is computationally infeasible. We tried to be as fair as possible and optimized the parameters of each workflow to the best of our abilities, but it cannot be ruled out that a chosen combination of parameters was suboptimal. The parameter settings for each workflow and benchmark dataset are listed in Supplementary Table A.1 in the Appendix.

4.4.3 Statistics

Beyond the processing workflows described above, the iPRG 2015[107] benchmark involves a downstream statistical analysis, which is – as the authors of the original study rightly emphasize – equally relevant to the overall performance of the solution. A key insight of the original study was that there is no single quantification tool or statistical approach or even combination thereof that is clearly superior to other methods. In the study, different participants employing the exact same combination of computational tools have produced vastly different results. Certain trends could be observed, for instance that spectral counting systematically underestimated fold changes, or that correction for multiple testing is important while the use of fixed fold change cutoffs does not work well. However, among the well-performing solutions, the combinations of data processing tools and statistical approaches was still rather diverse. The exact parameter settings and software versions of all involved computational tools and the precise sequence of data analysis steps have shown to have a substantial impact on overall performance.[107]

There are multiple ways to infer protein-level log fold changes and corresponding uncertainties. For instance, one way is to first summarize peptide-level intensities to protein intensities, and then perform statistical testing for difference of the mean protein intensity on the three replicates for each condition. The authors have included an example analysis protocol of this type in their publication[107]. Another idea is to test on the feature-level and then combine the feature-level p-values for each protein using a meta-analysis approach such as Fisher's method[109]. Yet another approach has worked best in our hands in an initial comparison of these strategies (data not shown) and has thus become the method of choice for this analysis:

Motivated by the common issues[110] of the t-test on small sample sizes (here, n=3), instead of testing on two groups of triplicates, our approach uses the ratios of all detected features of a protein from all replicates for the protein-level test, without prior summarization. In order to compute the p-value for differential expression of a protein P between conditions i and j, we first transform feature intensities to ratios by dividing the intensities of P's features in both conditions by the mean intensities across replicates in one of the two conditions (without loss of generality). Now, unlike feature intensities, all these ratios are on a comparable scale. Thus, we can perform a test for differential expression where one group contains the ratios for

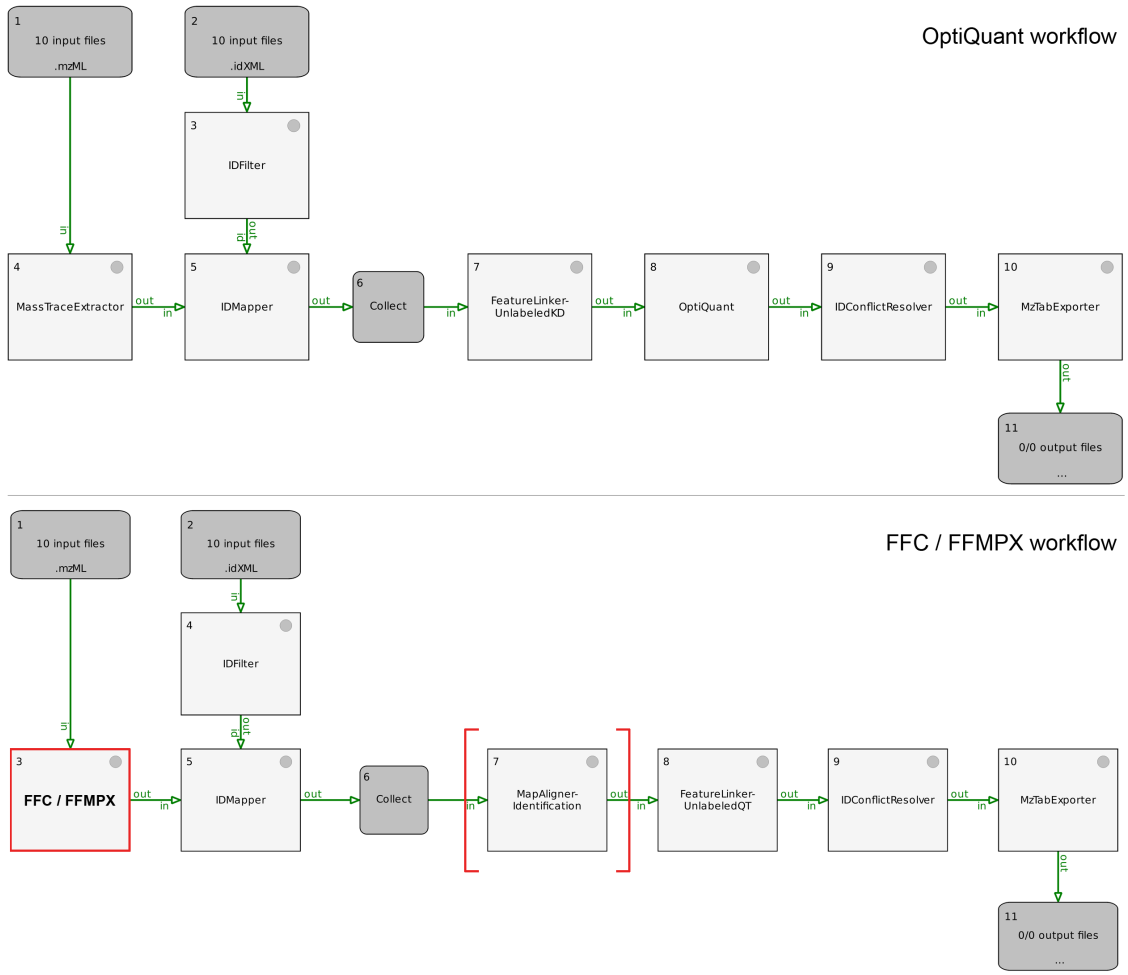

Figure 4.6: Top: The OptiQuant TOPPAS workflow applied to all benchmark datasets. Input files containing picked MS data in mzML format (node 1) are first processed by MassTraceExtractor. In a parallel branch, input files containing the corresponding identification results (node 2; generated by MSSimulator for synthetic data, provided by the organizers for iPRG data) are filtered to extract only identifications with a maximum charge of 5 (a handful of features obtained a higher charge state in the simulation) and to filter by iProphet score for the iPRG 2015 dataset. Subsequently, the peptide identifications in idXML format are mapped to the quantified mass traces in featureXML format using IDMapper. The *Collect* node waits for all preceding nodes to finish (so far, all input files have been processed sequentially), and then calls FeatureLinkerUnlabeledKD on the entire list of the annotated featureXML files. The result is a consensusXML file containing the linked consensus mass traces, serving as input to OptiQuant, which then performs the consensus feature assembly and outputs a consensusXML file containing the final linked consensus features for the entire dataset. Before exporting to mzTab format for downstream data analysis, identification conflicts within consensus features are resolved using IDConflictResolver. **Bottom**: Workflows using FeatureFinderCentroided or FeatureFinderMultiplex. The main difference to the OptiQuant approach lies in the fact taht these workflows perform run-wise feature detection as a first step (feature detection node highlighted in red), whereas OptiQuant only quantifies mass traces at this stage and delays assembly until after mass trace linking. The MapAlignerIdentification tool is in brackets because it was only used for analyzing the iPRG dataset, as we did not simulate systematic retention time shifts in the synthetic data (only random, feature-specific shifts) and thus do not benefit from transforming retention times prior to feature linking.

condition i and the other one contains the ratios for condition j[2]. Since sample sizes can be different due to missing values in one condition, we have used Welch's t-test for unequal sample size and variance. P-values were corrected for multiple testing using Benjamini-Hochberg FDR correction[111].

The main advantage of this method is that it significantly increases the sample size of the statistical test. Moreover, it circumvents protein-level intensity summarization altogether, which is a delicate endeavour due to varying ionization efficiencies and other effects making it infeasible to compare the signals of different analytes in the general case (see Section 2.2.3).

4.4.4 Results

We have assessed quantification performance of all investigated workflows in terms of sensitivity, reproducibility, and accuracy of quantification. Performance assessment on the synthetic datasets was limited to the OptiQuant, FeatureFinderCentroided, and FeatureFinderMultiplex workflows[3]. In addition to these key metrics, we present the results of our post-mortem reanalysis of the iPRG 2015 challenge data analysis task and compare them to the results submitted by the study participants.

Reproducibility

In the synthetic datasets, an ideal feature detection algorithm would detect the complete (same) set of peptides across all maps, with constant abundance for human background features and log-linear differential abundance for *E. coli* spike-in features. In the low-quality dataset, perfect quantification is clearly impossible as random noise was added to each simulated raw data peak, thus affecting quantification, and some of the signals fall below the limit of detection (LOD), resulting in missing data. For the noise-free high-resolution dataset, quantification is still challenging due to the sheer complexity of the simulated dataset: with ~36,000 simulated peptide features per run (42,000 for the spike-in dataset) distributed over a 1-h LC gradient, the degree of overlap between signals is relatively high. With finite instrument resolution, overlapping signals in the raw data can lead to inaccuracies in peak picking, which in turn translate to distorted or missing peaks, mass traces, and features. The amount of features found consistently across all (or a subset of $n > 1$) runs is thus a fraction of the total number of detected features. The higher these numbers, the more reproducible (and more likely to be sensitive, see Section 4.4.4) the feature detection is. Technical variation aside, the same expectations hold for the experimental dataset: In theory, all measured samples should contain

[2] One of those groups will have a mean log ratio of zero, as we divided by its mean intensity earlier.

[3] All attempts to import the simulated raw data into MaxQuant have failed due to its essentially abandoned support for free mass spectrometry data formats. The outdated mzXML format is the only non-proprietary option available, but the parser is nonoperational. We were unable to reverse-engineer a dialect of mzXML that would not cause MaxQuant to crash during loading (tested on six different MaxQuant versions between 1.2.2.5 and 1.5.8.0).

the same set of peptides, in constant amounts for background features, and in the respective differential amounts for spike-ins.

Figure 4.7 illustrates reproducibility of feature detection on simulated and experimental data. On the synthetic datasets, OptiQuant considerably outperformed FeatureFinderCentroided and FeatureFinderMultiplex for this metric: In the high-quality dataset with constant abundance, over 300,000 (30,000 per map) of the features detected by OptiQuant were reproducible across all 10 runs. Thus, OptiQuant detected ~40% more fully reproducible features than the other tools. In the low-quality variant of the dataset, OptiQuant was still able to quantify over 160,000 features, corresponding to a 24% improvement over FFC and 37% over FFMPX. In the dataset containing additional *E. coli* spike-ins, detection rates for the human background peptides were lower due to suppression by spike-in features.

On the experimental dataset, results look slightly different. Here, OptiQuant achieved the highest number of spike-in features detected across all maps, and a close to second-best result for the number of background features detected in all runs, but detected fewer of the less reproducible features than its competitors.

Precision and Recall

Knowing the exact ground truth of the synthetic datasets allows us to compute exact values for precision and recall of feature detection for each simulated map. The results of this analysis are depicted in Figure 4.8. All investigated tools showed a comparable and very high precision (>0.99, with one minor outlier for FFC on the noisy dataset when including irreproducible features). Recall, however, was significantly higher for OptiQuant in all investigated scenarios. Considering only fully reproducible features, OptiQuant achieved a recall of 0.70 on the noise-free dataset, whereas FFC and FFMPX achieved ~0.55. In the presence of noise and at low resolution, OptiQuant still achieved a recall of 0.41, compared to 0.32 for FFC and 0.29 for FFMPX.

Quantification Accuracy

So far, we have only compared numbers of detected features for each workflow while ignoring the actual quantification of analyte abundance. An equally critical measure of quantification performance is *accuracy*, denoting the proximity of the quantified feature intensity values to their true corresponding signal intensities. Even for simulated datasets (where the ground truth of raw data is available), accuracy can only be determined for relative quantification, since there is no single right way to quantify the absolute signal of an analyte (algorithms vary in the way they summarize peak intensities to mass trace intensities, which mass traces to include in the quantification, etc.).

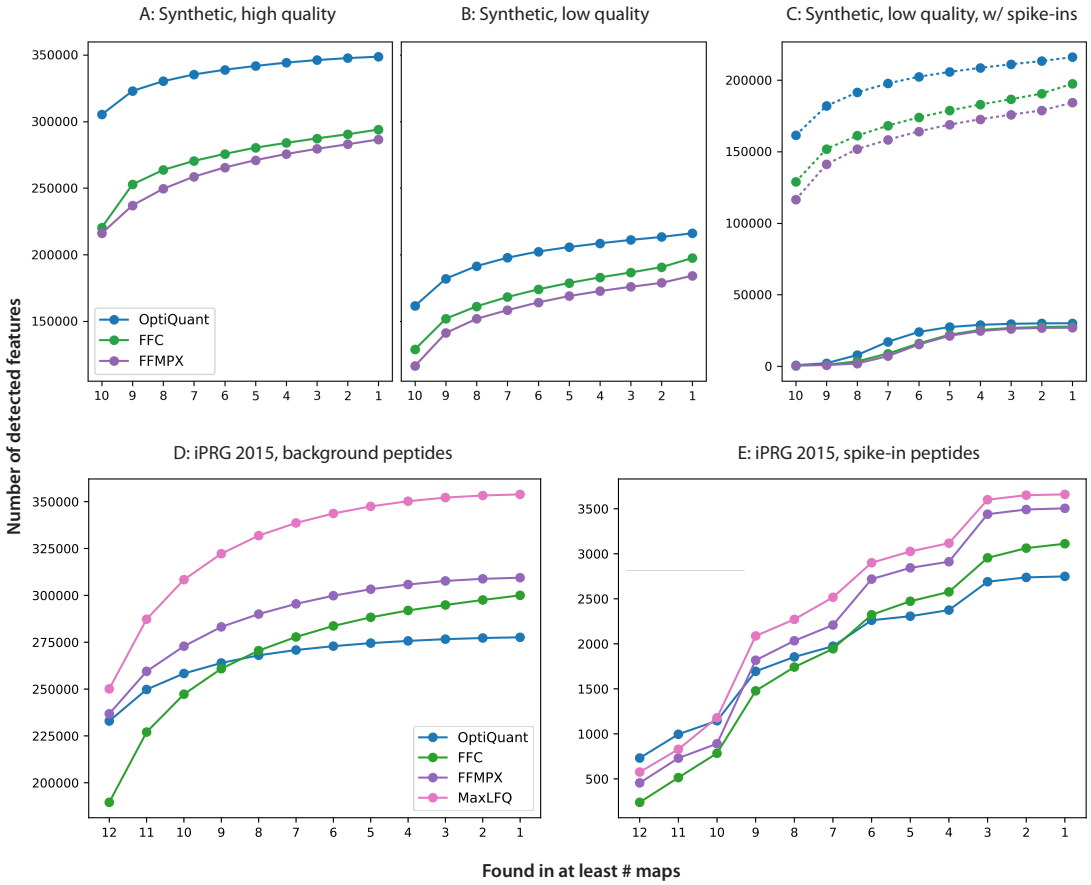

Figure 4.7: Comparison of reproducibility in constant background quantification on synthetic and experimental data. The plots show the total number of detected features as a function of a reproducibility threshold, the minimum number of maps these features were detected in. Hence, the leftmost datapoint indicates the total number of detected features when considering only those found in all 10 out of 10 maps; the next datapoint to the right represents the total number of detected features when additionally considering all those features found in only 9 out of 10 maps. The rightmost datapoint thus shows the total number of features detected in any map, regardless of reproducibility. A,B: Synthetic, constant background only; C: Synthetic, human background (solid line) plus *E. coli* spike-ins (dashed line) detected in the same raw data. D,E: separate plots for background and differential peptides in the iPRG dataset.

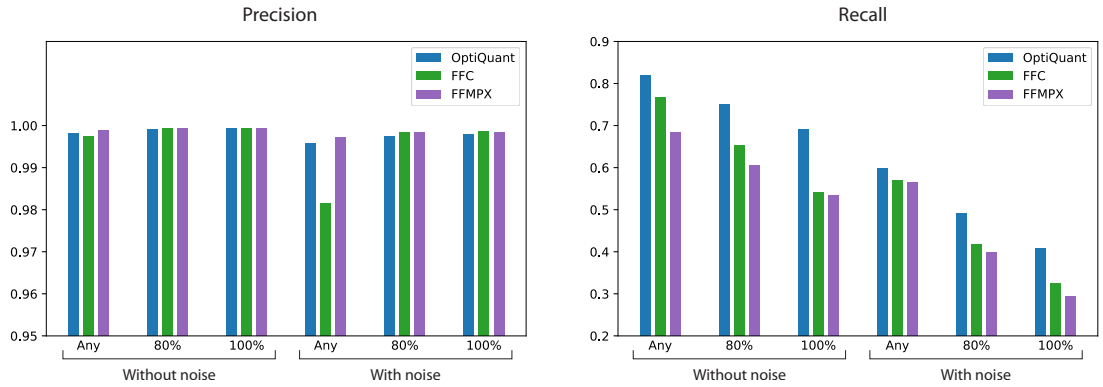

Figure 4.8: Precision and recall achieved by the investigated tools on the two simulated datasets containing only human peptides for varying reproducibility thresholds (considering only features found in at least 0%, 80%, 100% of the maps).

All investigated benchmark datasets consist of replicates with identical amounts of background peptides across runs. Thus, we can make an assessment of quantification accuracy by considering the standard deviations (STD) of log intensities of corresponding features across maps – ideally, they would all equal zero. FFC produced the most accurate quantification results with a median STD of 0.087 on high-quality and 0.089 on low-quality data. OptiQuant was a close second with a median STD of 0.092 on high-quality data and 0.091 on low-quality data. FFMPX was least accurate and achieved a median STD of 0.116 on high-quality and 0.113 on low-quality data. On the iPRG 2015 dataset, OptiQuant and FFC lead the field with a median STD of 0.073, followed by MaxLFQ (0.085) and FFMPX (0.11). Thus, in terms of overall accuracy of constant background quantification, OptiQuant and FFC performed best and approximately on par across all benchmark datasets. On the simulated datasets, their performance was about 20% better than FFMPX's, on the iPRG data they were 15% more accurate than MaxLFQ and 33% better than FFMPX.

In order to further investigate the unexpected finding that overall quantification accuracy was not worse, but instead even slightly better on the low-quality synthetic dataset, we have looked at the intensity dependence of the variation. Figure 4.9 illustrates the results of this analysis: On the high-quality dataset with high resolution, no m/z error, and no intensity noise, the average standard deviation is higher for consensus features at the lower end of the intensity distribution than on the low-quality data. These low-intensity features could not be quantified in the low-quality data (and hence do not contribute to additional complexity) since their intensity was below the noise threshold.

Having compared quantification accuracy on peptides with constant abundance across runs, let us consider the (much more interesting) case involving differentially abundant features. To this end, we have investigated quantification accuracy on the third simulated dataset containing

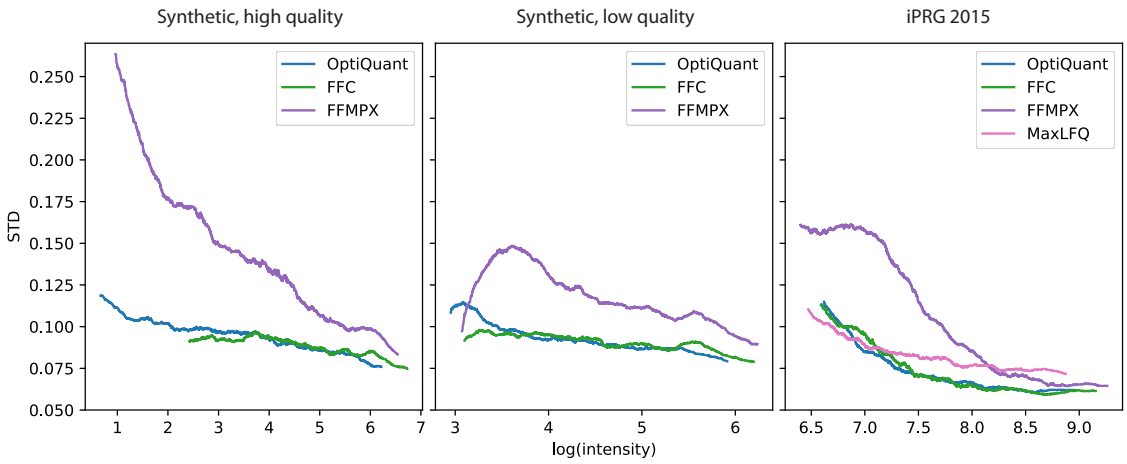

Figure 4.9: Standard deviation of log intensities for corresponding features as a function of median log intensity, smoothed using an unweighted averaging window of 1,000 data points.

constant human background and an additional log-linear concentration series of *E. coli* peptides in the same raw data. Figure 4.10 shows a scatter plot of quantified ratios against true ratios for all quantified *E. coli* peptides, together with the regression line of an ordinary least squares (OLS) fit of a linear model. Map number 10 (the one with the highest simulated concentration of *E. coli* peptides) served as the reference. Each of the other maps corresponds to one of the discrete true log ratios on the x-axis. We can make several observations here:

Firstly, none of the algorithms were able to correctly quantify *E. coli* signals in the map with lowest *E. coli* concentration, and only very few features were detected in the map with second lowest concentration. Secondly, besides the bulk of quantified ratios concentrated around the true ratio, there is a considerable amount of outliers above the regression line, corresponding to grossly underestimated fold changes. A closer look at the data revealed that these are actually the result of linking errors, where the signal of a human background peptide was assigned to an *E. coli* consensus feature. This explains the bimodality observed for FFC and FFMPX: the distribution of outlier ratios (which looks similar for each map) corresponds to the intensity distribution of human background features (which is identical for each map). This bimodality cannot be observed for the OptiQuant workflow, where linking errors occur on mass trace level rather than on feature level. Thus, instead of linking entire features to the wrong consensus, here we misassign only single mass traces. Because mass traces span a much wider intensity range than the features themselves (due to the high dynamic range of isotopic mass trace intensities), the linking error population is much more widely distributed here. In order to mitigate the effect of linking errors on the regression analysis, we have removed outliers in each map (upper/lower quartile +/- 1.5 × interquartile range) prior to linear regression. Overall, OptiQuant's and FFC's quantification results exhibit near perfect

Data	Workflow	R^2	Slope	Intersect
All data	OptiQuant	0.857	0.895	-0.127
	FFC	**0.895**	0.934	**-0.085**
	FFMPX	0.799	**0.958**	-0.102
Outliers removed	OptiQuant	**0.974**	0.987	-0.016
	FFC	0.972	**1.001**	**-0.004**
	FFMPX	0.957	1.092	0.062

Table 4.4: Summary of the linear regression results on the *E. coli* peptide concentration series for all investigated tools with and without filtering outliers. Best values per comparison are printed in boldface (highest R^2; slope closest to 1; intersect closest to 0)

linearity across the investigated ratio range. Table 4.4 contains a summary of the results of the linear model fit (with and without filtering outliers). After filtering, OptiQuant achieved the highest coefficient of determination, whereas FFC produced the best slope estimate. FFMPX performed significantly worse for both metrics.

Figure 4.11 illustrates the overall root-mean-square error (RMSE) of the quantified log ratios as a function of the true log ratio for the synthetic dataset with differential spike-ins. The extraordinarily high quantification error on the lower end of the concentration series is due to the very low numbers of detected *E. coli* features in these maps and the fact that they are outnumbered by false-positives resulting from linking errors. Removing outliers for the first two maps with lowest concentration did not have much of an effect on the quantification error, since the median of these distributions is actually closer to the average ratio of the linking error population than to the true ratio. For the higher concentrations, however, filtering outliers improves the RMSE considerably. In accordance with the linear regression results, FFC and OptiQuant were essentially on par in this comparison (FFC having a slight edge), while FFMPX was placed a distant third. The best average RMSE (ignoring the first two maps with too little data) was achieved by FFC (0.138), followed by OptiQuant (0.149). FFMPX's error on these maps was almost twice as high (0.269).

Last, but not least, we have compared quantification accuracy for the differential spike-ins in the iPRG dataset on the protein level. Figure 4.12 shows the log-log plot of all quantified protein ratios against the true ratios, together with the overall distribution of quantified ratio errors, for each of the investigated workflows. The best total RMSE of quantified protein log ratios was achieved by OptiQuant (0.71), followed by MaxLFQ (0.85), FFC (0.89), and the distant fourth FFMPX (1.68).

iPRG 2015 Challenge

The challenge task was to provide a list of differentially abundant proteins found in the dataset, together with log fold changes and a characterization of the uncertainty of the results, for each pairwise sample comparison. There were four samples (with three replicates each), hence six

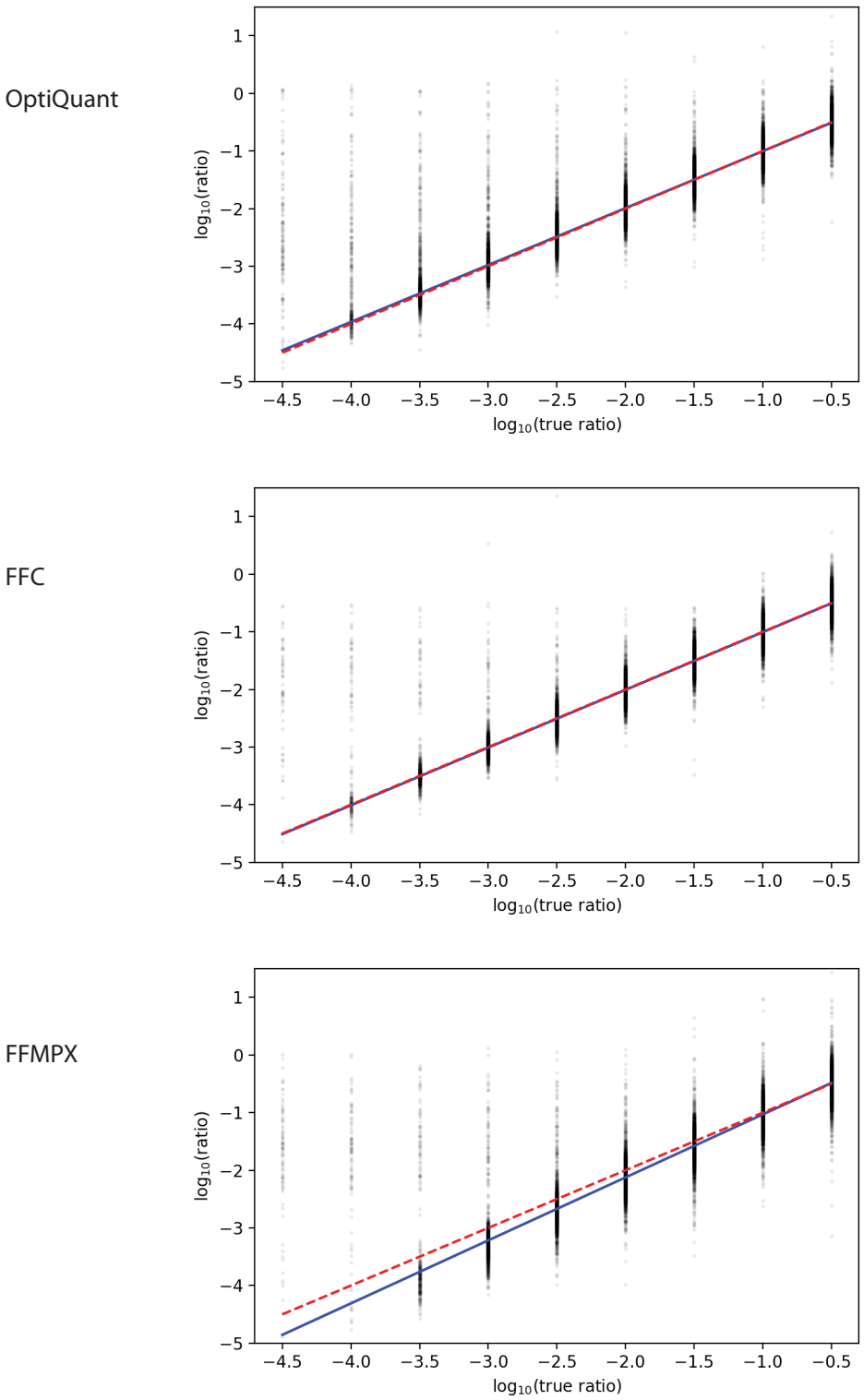

Figure 4.10: Scatter plot of quantified log ratios against true log ratios for all features with *E. coli* identification in the synthetic dataset with differential spike-ins. The map with the highest concentration of *E. coli* peptides served as reference for computing ratios. Each of the other maps corresponds to one of the nine discrete log ratios on the x-axis. The blue line depicts the results of the linear regression after removing outliers for each map. The red dashed line indicates perfect quantification.

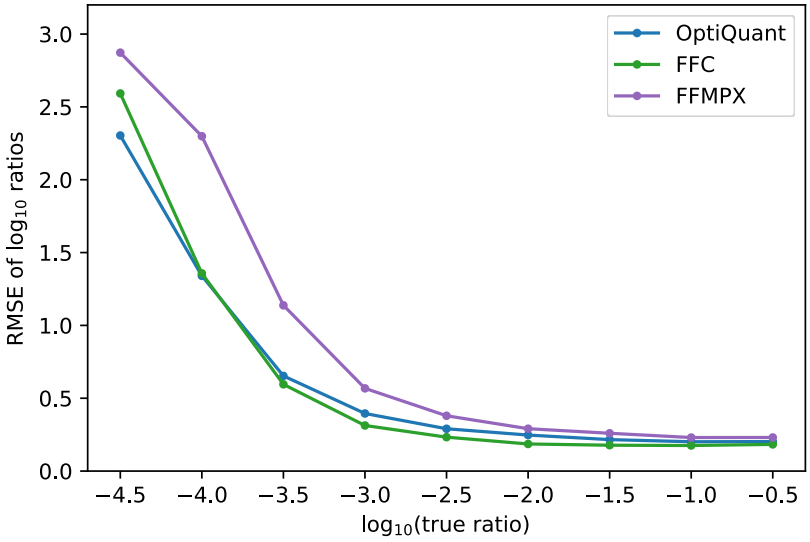

Figure 4.11: RMSE of quantified log intensities.

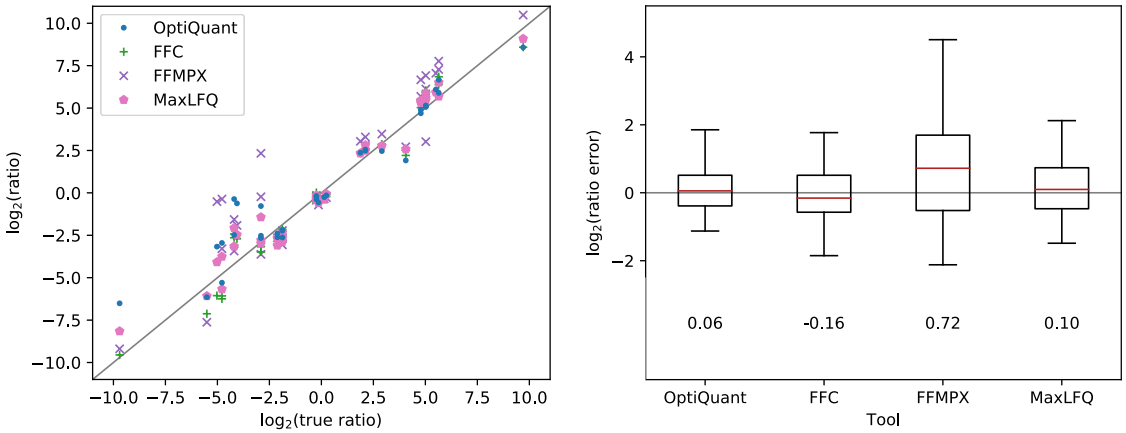

Figure 4.12: Protein-level quantification accuracy on the iPRG dataset. Left: scatter plot of quantified protein ratios vs true protein ratios. Right: corresponding distributions of quantification errors. Numeric values shown below the boxplots denote the median.

pair-wise comparisons. Each sample contained certain differential amounts for each of the six spike-in peptides, whereas the background concentrations remained constant. Thus, a perfect estimator would find 36 true positives (all six spike-ins across all six pairwise comparisons) and zero false positives (background proteins falsely classified as differential). Interestingly, in contrast to our findings on the synthetic datasets, OptiQuant was the least sensitive workflow on background proteins in this comparison: After filtering out features with decoy identifications, aggregating by sequence and charge (thus combining split features that quantify the same signal), and removing single-peptide-hit proteins, OptiQuant was able to quantify 2,573, FFC 2,809, FFMPX 2,711, and MaxQuant 3,175 proteins.

In the original study results[107], the 48 correct submissions were first grouped into three different groups based on the positive predictive value (PPV; also known as precision) of the classification into differential / non-differential. 19 submissions had a PPV $>= 0.7$, 10 submissions had a PPV between 0.2 and 0.7, and 19 submissions had a PPV of < 0.2. Within these groups, results were further ranked by the number of true positives, and ties were broken by the number of false positives. In this ranking of 52 submissions (48 original plus our four), the results produced by our workflows would have been placed as shown in Table 4.5. According to the ranking criteria of this competition, MaxQuant would have achieved the highest rank (7^{th}), followed by OptiQuant (8^{th}), FFC (11^{th}), and FFMPX (15^{th}). Out of the four workflows compared in our reanalysis, OptiQuant was the only one achieving a perfect PPV of 1.0 (28 TP / 0 FP).

Rank (1 - 52)	Workflow	PPV	TP	FP
7	MaxQuant	0.88	29	4
8	OptiQuant	1.00	28	0
11	FFC	0.96	26	1
15	FFMPX	0.92	24	2

Table 4.5: Ranking of our results within the participant submissions of the original iPRG 2015 study, including positive predictive value (PPV), number of true positives (TP), and number of false positives (FP).

4.5 Discussion

We have presented OptiQuant, a novel tool for label-free quantification of LC-MS proteomics data using mixed-integer programming for globally optimal consensus feature assembly, and demonstrated its quantification performance in a comparison with three state-of-the art LFQ solutions across synthetic and experimental benchmark datasets. The results of these first benchmark comparisons are very encouraging: On all three synthetic datasets, OptiQuant clearly outperformed its competitors in terms of sensitivity and reproducibility by a large margin, while achieving an overall quantification accuracy comparable to FeatureFinderCentroided's

(FFC) and considerably better than FeatureFinderMultiplex's (FFMPX). This was true for both high-quality and low-quality simulated data, for background features with constant abundance, as well as for differential spike-in features. On the experimental iPRG 2015 challenge dataset, OptiQuant showed lower sensitivity than the other three investigated workflows, but achieved the best overall accuracy on both background features and spike-ins. In terms of the actual challenge results, OptiQuant was the only workflow in our comparison producing a result with perfect PPV of 1.0 (28 TP / 0 FP). MaxQuant detected 29 TPs with 4 FPs. In the ranking of 48 submitted results from the participants of the orginal study, our MaxQuant results would have been ranked 7th, the OptiQuant results 8th. It shall be noted, however, that these ranking criteria were somewhat arbitrary. Whether to prefer higher numbers of true positives or lower numbers of false positives depends, to a significant degree, on the application. In some scenarios (for instance, when positive observations will be subject to further very expensive testing), precision can be of higher importance than sensitivity. If we rank submissions by PPV (ties broken by number of TP and FP), the OptiQuant workflow is ranked third (two submissions also achieved a PPV of 1.00 but found 30 TPs instead of OptiQuant's 28), FFC is ranked 10th, FFMPX is ranked 12th, and MaxQuant is ranked 14th.

An interesting and unexpected finding on the synthetic datasets was that overall quantification accuracy on the low-quality data was actually not worse, but even slightly better than on the high-quality data. This may seem counterintuitive at first, but can be explained in light of the corresponding results for sensitivity (Figure 4.7) and intensity-dependence of quantification accuracy (Figure 4.9): All investigated workflows showed significantly lower sensitivity on low-quality data. Thus, the main impact of the decrease in data quality (additional intensity noise, mass error, and lower resolution) was not on quantification accuracy for detectable features, but on the sensitivity of feature detection. Figure 4.7 indicates that a large portion of the total intensity variance among corresponding features is actually caused by low-intensity consensus features. These features are simply not detectable in the low-quality dataset as they fall below the noise level, and hence they cannot have a negative impact on the overall quantification accuracy there. In other words, the low-quality dataset is – in practical terms – less complex and has a lower dynamic range, even though it is based on the exact same theoretical ground truth.

Another interesting finding was that FFMPX showed a significant systematic bias towards over-estimation of fold changes that could not be observed for any of the other workflows. This is clearly indicated by the linear regression results on the synthetic spike-in concentration series (Figure 4.10 and Table 4.4) and by the distribution of quantification errors on experimental spike-ins in the iPRG dataset (Figure 4.12). In accordance with these results, Figure 4.11 shows that filtering outliers (primarily caused by linking errors) has not had that much of a positive effect on the quantification accuracy of FFMPX, while it considerably increased accuracy for OptiQuant and FFC. A possible explanation for this could lie in the fact that FFMPX, according

to our understanding of the algorithm, computes the overall intensity of a mass trace simply by summing up all peak intensities. A more precise way of estimating the signal intensity would be to compute the *area* under the chromatographic peak. Especially in the presence of missing peaks, this circumstance could potentially explain the effect. However, this remains mere speculation at this point, and needs to be investigated in more detail. In retrospect, our decision to prefer FFMPX over FFC as the feature detection tool of choice in the LFQProfiler workflow (see Section 3.5.2) seems questionable. When faced with this decision, FFMPX had shown to be slightly more sensitive than FFC at comparable accuracy in an ad-hoc benchmark (data not shown). This benchmark, however, involved only the quantification accuracy on features of constant abundance and did not involve any differential features, hence the effect was not visible. This underlines, once more, the importance of a diverse set of benchmark metrics and datasets.

While these initial results are very promising, there is room for improvement. The bar for an approach like OptiQuant is quite high, as the considerably increased problem complexity and the resulting CPU and memory requirements must be justified. Using OptiQuant's current implementation, we were unable to achieve results that are objectively better than MaxQuant's in the iPRG benchmark, for instance. OptiQuant is currently a proof-of-concept implementation that needs further improvement in several areas before it can be used in production. A big current hurdle is the workflow's large number of parameters, that still need too much manual adjustment by the user. FFC, FFMPX, and MaxQuant have shown to be more robust to parameter changes, and seemed to work pretty well with default parameters. With the OptiQuant workflow, however, the quality of the results using default parameters for each tool often resulted in unacceptable performance. In particular the mass trace detection and the OptiQuant assembly algorithm itself expose a significant number of parameters to the user that really must be tuned in order to achieve good performance. Thus, a practical requirement for OptiQuant's production use would be the reduction of parameters exposed to the user. Ideally, OptiQuant should be able to estimate non-intuitive parameters directly from the data.

Our modifications to MassTraceExtractor (pre-filtering of noise peaks based on presence or absence of isotopic peaks) have slightly improved the robustness of the mass trace detection part, but it was still rather cumbersome to find a combination of parameters that work well together, and it seems to be very dependent on the particular dataset at hand (complexity, resolution, noise-level, etc.). A tradeoff has to be made between precision and recall of mass trace detection, but both are actually critical to the quality of the downstream OptiQuant consensus feature assembly. Too many missing traces lead to inaccurate quantification or missing data, whereas false-positive traces increase the likelihood of linking errors. Since it is such a critical step in the OptiQuant workflow, it might be worth the effort to try and improve the mass trace detection algorithm. A potential idea would be to modify it in such a way that it takes isotopic traces into account while collecting the peaks for a mass trace hypothesis.

In our experience, fine-tuning the parameters that regulate the conditions under which trace extension stops was critical to performance. We speculate that it might be much easier to estimate whether to stop or continue mass trace extension if corresponding isotopic traces are considered simultaneously.

Processing speed and memory consumption are further aspects that should be addressed. As mentioned before, the increased problem complexity is to a certain degree inherent in OptiQuant's fundamental idea to perform consensus feature assembly using the quantified mass traces from all maps at once. Run-wise mass trace extraction and run-wise feature detection are comparable in terms of runtime (seconds or few minutes per map). Linking all unassembled mass traces (hundreds of thousands per map) will always be slower than linking features (tens of thousands). But the additional amount of CPU hours used by CPLEX for solving the optimization problem for each connected component is currently very hard to predict and varied wildly for different datasets and parameter settings in our initial tests. For slightly different parameter settings on the iPRG dataset, for example, OptiQuant in one setting was using all 24 CPU cores and 55GB of RAM for more than 20 minutes, and in another it was using less than one core on average for about five minutes and a total of 12 GB of RAM for the entire optimization. Due to the high degree of sophistication of modern MIP solvers such as CPLEX, the employed pre-solving and optimization strategies and the resulting runtime and memory requirements of the optimization are impossible to predict a priori. But it can be said in general that the average complexity of these optimization problems in our case is sensitive to the degree of overlap between hypotheses, which in turn is very sensitive to a variety of user parameters (maximum number of isotopic traces considered, averagine similarity threshold, m/z and RT tolerance, allowed number of missing traces, etc.). Finding more reliable ways of filtering out unlikely hypotheses prior to optimization will help reduce the average size of clusters of conflicting hypotheses, and thus help control the complexity of the corresponding optimization problems. In order for OptiQuant to be reliably applicable to hundreds or thousands of label-free runs on machines with, say, less than 512 GB of memory, additional mitigations of the overall runtime and memory complexity will be necessary. One ambitious idea would be to slice the dataset in the RT or m/z dimension (or both) across all runs, and then to generate hypotheses and select optimal features on each slice individually. The overall complexity would still be linear in the number of input maps, but with a much smaller constant factor. If we want to preserve the optimality of the result, however, such an approach would require an additional (non-trivial) step to combine sub-optimal results near slice boundaries to globally optimal results.

Regardless of the current practical limitations of the proof-of-concept implementation, OptiQuant is a valuable general framework for prototyping similar solutions, free for anyone to use and adapt under a permissive three-clause BSD license. At the very heart of the approach lies the hypothesis scoring function, which ultimately determines which hypotheses will be

selected and which ones will be abandoned. This scoring function was essentially formulated ad-hoc, inspired by our own experience and expectations of what true features should look like. In other words, there is no mathematical or probabilistic justification for this particular scoring function. Systematic experiments on a diverse set of datasets might lead to the discovery of better scoring criteria and thus help further improve performance of the overall approach.

Chapter 5

Forensic Applications of Mass-Spectrometry Based Proteomics

Adapted with permission from

Mass Spectrometry-Based Proteomics Reveals Organ-Specific Expression Patterns to Be Used as Forensic Evidence

Sascha Dammeier+, Sven Nahnsen+, Johannes Veit+, Frank Wehner, Marius Ueffing, and Oliver Kohlbacher

J Proteome Res. 15(1):182-92. (2016)

+ These authors contributed equally.

5.1 Introduction

Forensic medicine is an important part of jurisprudence that involves medical and analytical knowledge according to legal aspects. A forensic examiner is preferably consulted to pass an expert opinion about medico-legal or ethical issues in cases involving death, drug abuse, rape, paternity tests, etc. Apart from classical examination methods like blood analyses, psychiatric interviews and autopsies, forensic medicine has been revolutionized by modern molecular biology techniques. In particular, the analysis of DNA via the application of polymerase chain reaction (PCR) and therefore the use of traces of DNA as evidence has become a routinely used method in legal medicine[113,114]. Apart from genomic technologies, modern analytical techniques like liquid chromatography (LC) coupled to high-performance mass spectrometry (MS) have already been established in forensic medicine, e.g., for the identification and

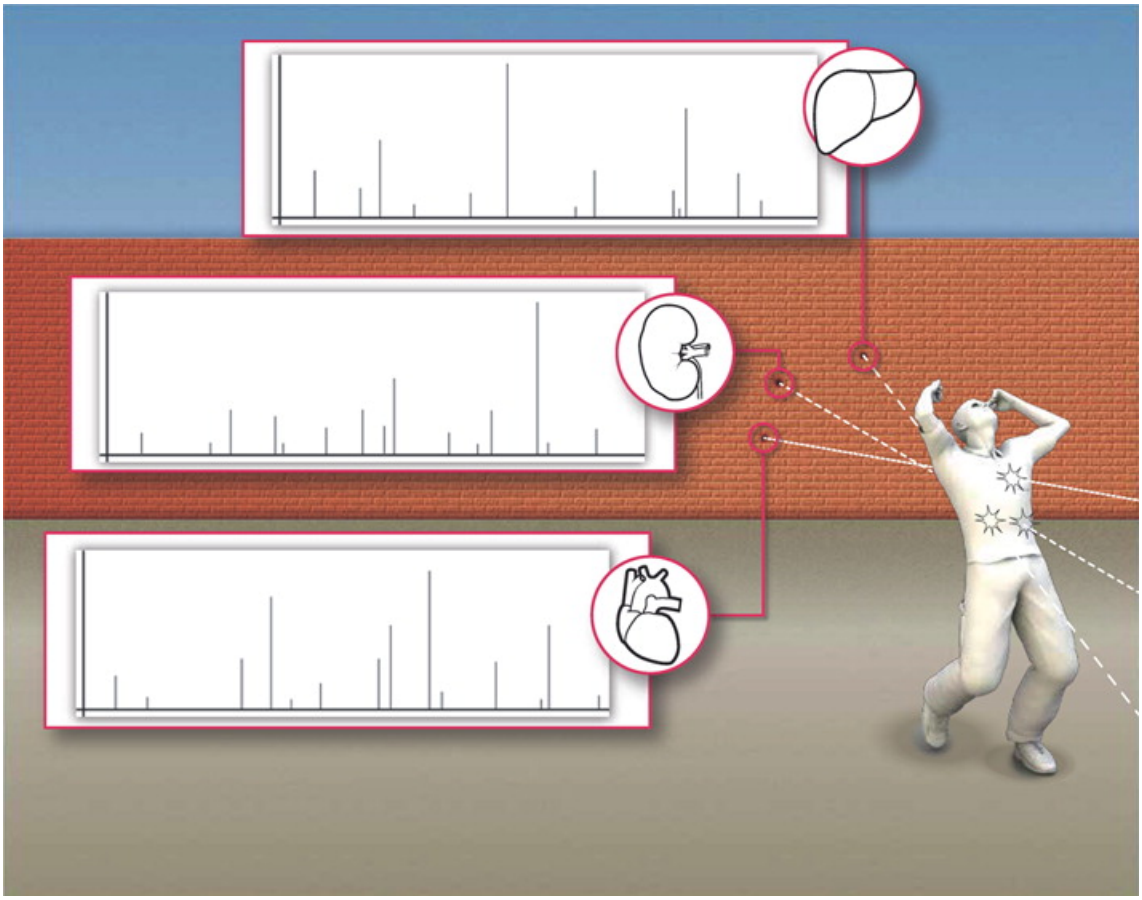

Figure 5.1: Visual Abstract. Mass spectrometry-based proteomics reveals organ-specific expression patterns to be used as forensic evidence. Reprinted from Dammeier et al.[112]

quantification of illegal or toxic substances in body fluids[115]. One of the most prominent analyses reported so far is the determination of recombinant forms of erythropoietin, a major performance-enhancing drug in sports, within the scope of doping tests[116].

In the case of shootings with fatal outcome, a standard task for the forensic practitioner is the examination of bodies with respect to the impact of projectiles. During postmortem examination regarding the cause of death, it is a special task to reconstruct the crime via analysis of bullet channels. Furthermore, bullets are routinely matched to certain weapons by standard ballistic and technical analyses[117]. The question of which projectile has hit which subject can be solved by modern PCR analysis of traces of DNA remaining on the surface of the projectile[118]. However, it is far more challenging to determine which (vital) organs have been penetrated by a certain bullet just from the projectile as single evidence, especially in cases of multiple hits and more than one shooter. This is of particular importance in countries with moderate gun control. For instance, in the US statistically one murder took place every 37 minutes in 2013, however, the clearance of murder and non-negligent manslaughter cases was just 64.1% in the same year[119].

In the case of multiple hits and/or shooters, the examination of nucleic acids is rather inadequate, since RNA is degraded rapidly in most cases, and DNA cannot be used to identify organ-specific gene expression. However, with regard to RNA, and in particular mRNA, which should also represent tissue-specific profiles as it is translated into proteins, some applications concerning forensic science have been established (e.g. to determine the age of biological samples like bloodstains and hairs[120,121]). These methods seem to work well although with certain limitations. Here, age determination is based on the complex, non-linear degradation kinetics of particular RNA species that are both generally detectable by RT-PCR and undergo different degradation kinetics, which allows a relative semi-quantification and, finally, age-determination. Thus, in case of bullets and biological debris, an RNA-based method to determine organ specificity would mean tedious work to find characteristic mRNA pairs that exhibit well characterized kinetics as well as organ-specificity. Therefore forensic scientists have focused research on easier methods like cytological detection and immunochemistry. However, those methods have exhibited limitations with regard to the stability or integrity of the biological material remaining on the piece of evidence[122,123]. Furthermore, analyses based on the immunological identification of tissues via antibody epitopes might not be sufficient to answer the central question of which projectile is likely to have caused the lethal impact.

Proteome analysis combined with bioinformatics tools allows a comprehensive description of a large set of expressed proteins as well as their interactions and modifications in the context of a biological process. High-resolution MS coupled with high-performance LC has become the analytical technology of choice in many proteomic studies. LC-MS experiments allow the identification and quantification of thousands of proteins in a single instrument run[124]. Inspired by its analytical power with regard to sensitivity and selectivity, we set out to examine

the remnants of protein material on the surface of bullets in order to identify organ-specific protein patterns. The proposed forensic workflow is visualized in Figure 5.2.

The identification of tissue-specific protein markers is one way to find corresponding organs. Mass spectrometry-based proteomics has previously been shown to be a powerful technology in assigning patterns of protein expression and protein modification to the tissue of origin[125]; however, those have never been exploited for forensic application[126]. In a first attempt, protein signatures that allow unambiguous determination of tissues/organs in the circumstances of crime need to be identified experimentally. Once such signatures are established, they can serve as organ-specific markers and would be used to make organ assignments to projectiles, and probably to other types of penetrating weapons.

Organic material remaining attached to bullets was therefore analyzed, and LC-MS/MS raw data as well as the corresponding protein identifications thereof were used to correlate individual organs to protein expression patterns. To get started we designed a very artificial experimental setting using bovine organs and manual penetration being aware that the findings needed to be refined on different, more realistic validation levels in the course of our studies (e.g., testing high-speed penetration, dealing with contamination issues, and translation into forensic practice). The identified correlations were highly significant and reproducible. Moreover, experiments have been performed to demonstrate the robustness and transferability of the experiments to shootings and human material. Using evidence of a real murder case, and combining our experimental, molecular findings with results from classical trajectory analysis, the method generated highly valuable additional information qualifying it as a method of choice for unclear forensic cases.

5.2 Materials and Methods

5.2.1 Sample Taking

For the main data set, different metal projectiles, namely (a) 7.62 mm bore (Tokarer), (b) blunt, plain metal pins, 5.0 mm bore, (c) precision bullets, 4.55 mm bore (H&N Sport), (d) 6.35 mm bore (Sellier/Bellot), and (e) 6.35 mm bore (Geco/Dynamit Nobel), were used to penetrate a selection of four bovine inner organs (kidney, lung, liver, and heart; received directly and fresh from the abattoir) and one ubiquitous control organ (skeletal muscle: received from the same animals) by manual force for a total contact time of approximately 2 seconds. The air-dried projectiles were kept in reagent vials until further processing. For each of the five organs, the penetration by the five different bullets was repeated on three biological replicates (four in the case of lung). One of the liver runs was discarded for technical reasons. Hence, the main data set comprises 3 replicates $\times$ 5 organs $\times$ 5 bullets + 5 additional lung samples – 1 bad sample = 79 samples.

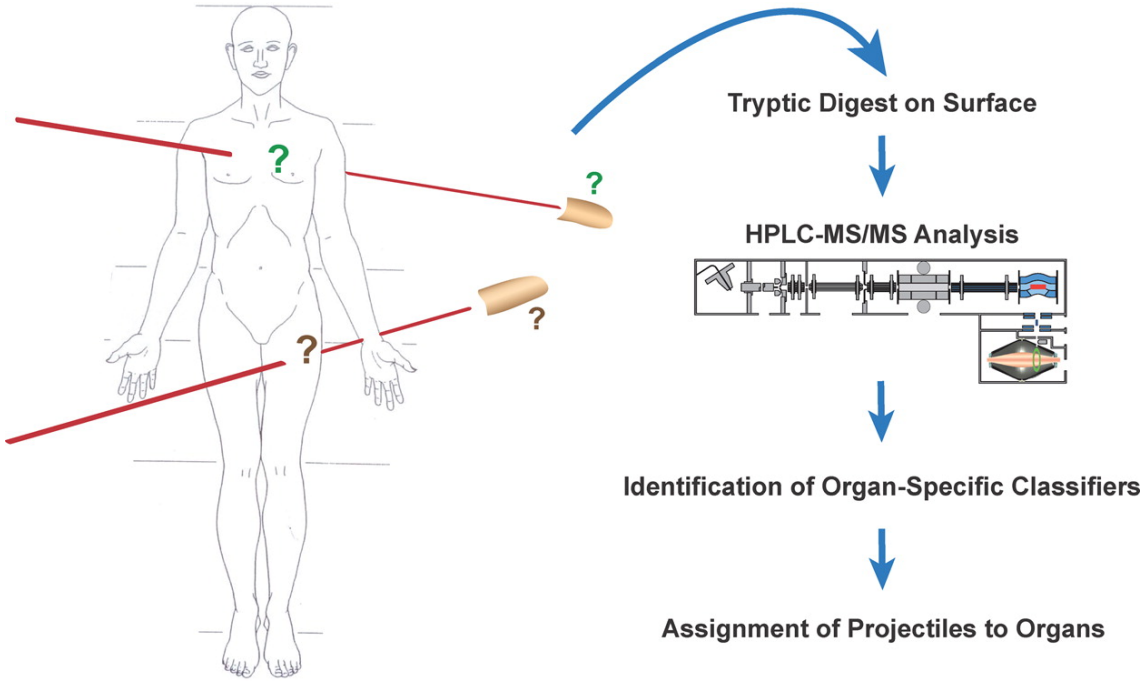

Figure 5.2: Schematic visualization of the overall workflow for the identification of organ-specific proteins to be used as forensic evidence. Projectiles are processed by tryptic proteolysis on the surface. The resulting peptides are analyzed by liquid chromatography and mass spectrometry followed by bioinformatic and statistical analysis to identify organ-specific discriminating proteins. Reprinted from Dammeier et al.[112]

In addition, shooting experiments using selected bovine organs embedded in gelatin blocks were performed as described earlier[123]. Due to the additional costs and effort, we restricted ourselves to four of these samples (kidney, liver, lung, and heart) for a proof of concept.

Bullets derived from a homicide were received blinded from the coroner in small plastic bags according to standard procedures used to store court exhibits. The bullet that was taken as a blood contamination control was covered with two drops of fresh blood derived from a healthy male volunteer and treated like the samples of the bovine experiments.

5.2.2 Proteomics Sample Preparation

In order to obtain peptide samples that are suited for mass spectrometric analysis, the bullets were covered with 50 mM ammonium bicarbonate solution containing 10% RapiGest SF Surfactant (Waters, UK). Subsequently, dithiothreitol was added (3 mM final concentration) and they were incubated for 15 min at 60 °C followed by the addition of iodacetamide (10 mM final concentration) and incubation for 30 min at room temperature in the dark. Finally, the bullet sample was incubated in its supernatant liquid with trypsin (Sigma Aldrich, Germany) for a minimum of 14 h at 37 °C to perform limited proteolysis. The reaction was stopped by applying trifluoroacetic acid to a final concentration of 5%. The samples were centrifuged (5 min at $16,000 \times g$), and the supernatant was recovered and processed using StageTips (Thermo Fisher Scientific, Germany) according to the manufacturer's protocol. The resulting peptide solution was lyophilized and stored at -20 °C until analysis.

5.2.3 Mass Spectrometry

LC-MS/MS analysis was performed on a NanoRSLC3000 HPLC system (Dionex) coupled to a LTQ OrbitrapXL mass spectrometer (Thermo Fisher Scientific) by a nano spray ion source. Tryptic peptide mixtures were automatically injected and loaded at a flow rate of $6\,\mu L\,min^{-1}$ in 98% solvent C (0.1% trifluoroacetic acid in HPLC-grade water) and 2% solvent B (80% actetonitrile and 0.08% formic acid in HPLC-grade water) onto a nano trap column (75 µm i.d. × 2 cm, packed with Acclaim PepMap100 C18, 3 µm, 100 Å; Dionex). After 5 minutes, peptides were eluted and separated on the analytical column (75 µm i.d. × 25 cm, Acclaim PepMap RSLC C18, 2 µm, 100 Å; Dionex) by a linear gradient from 2% to 35% of solvent B in solvent A (2% acetonitrile and 0.1% formic acid in HPLC-grade water) at a flow rate of $300\,nL\,min^{-1}$ over 150 minutes. Remaining peptides were eluted by a short gradient from 35% to 95% solvent B in 5 minutes. The eluted peptides were analyzed using a LTQ Orbitrap XL mass spectrometer. From the mass spectrometry pre-scan at a resolution of 60,000 with a m/z range of 300 – 1,500, the 10 most intense peptide ions were selected for fragment analysis in the linear ion trap if they exceeded an intensity of at least 200 counts and if they were at least doubly charged. The normalized collision energy for collision-induced dissociation was set to a value of 35, and the

resulting fragments were detected with normal resolution in the linear ion trap. The lock mass option was activated and set to a background signal at m/z 445.1200215. Every ion selected for fragmentation was excluded for 20 seconds by dynamic exclusion.

5.2.4 Data Processing

For qualitative protein identification the raw data were analyzed using Mascot (Matrix Science, UK; version 2.4.1) and Scaffold (version 3.6.5, Proteome Software Inc., USA). Tandem mass spectra were extracted, charge state deconvoluted and deisotoped by `extract_msn.exe` version 5.0. All MS/MS samples were analyzed using Mascot. Mascot was set up to search the Uniprot database (selected for *Bos taurus*, extracted on 2013-04-23, 31462 entries) assuming the digestion enzyme trypsin. Mascot was searched with a fragment ion mass tolerance of 1 Da and a parent ion tolerance of 10 ppm. Carbamidomethyl of cysteine was specified in Mascot as a fixed modification. Deamidation of asparagine and glutamine and oxidation of methionine were specified in Mascot as variable modifications. Scaffold was used to validate MS/MS based peptide and protein identifications. Peptide identifications were accepted if they could be established at greater than 80.0% probability by the Peptide Prophet algorithm[127] with Scaffold delta-mass correction. All filtered peptides are than subjected to protein inference using the Protein Prophet model. Protein identifications were accepted if they could be established at greater than 99.0% Protein Prophet probability and contained at least 2 identified peptides. Protein probabilities were assigned by the Protein Prophet algorithm[14]. Finally, the probability-filtered peptides and proteins were subjected to peptide and protein FDR calculation using the probabilistic approach as implemented in the Trans Proteomic Pipeline[57], resulting in a final protein FDR of 0.1% (and 2.4% on the peptide level). Proteins that contained similar peptides and could not be differentiated based on MS/MS analysis alone were grouped to satisfy the principles of parsimony.

After data conversion with Proteowizard[15,128], the quantitative data was processed using OpenMS/TOPP in version 1.11[11,12] for peak picking, feature detection, map alignment, feature linking, and intensity normalization. An overview of the processing workflow implemented in TOPPAS is depicted in Figure 5.3. Finally, the statistical software R, in version 2.15.1, was used for principle component analysis (PCA) and subsequent statistical assessment. Classification was performed using python and the scikit-learn package, version 0.14.1[129,130].

5.2.5 Data Deposition

The mass spectrometry proteomics data have been deposited to the ProteomeXchange Consortium[131] via the PRIDE partner repository[132] with the dataset identifier PXD002193 and DOI `http://dx.doi.org/10.6019/PXD002193`.

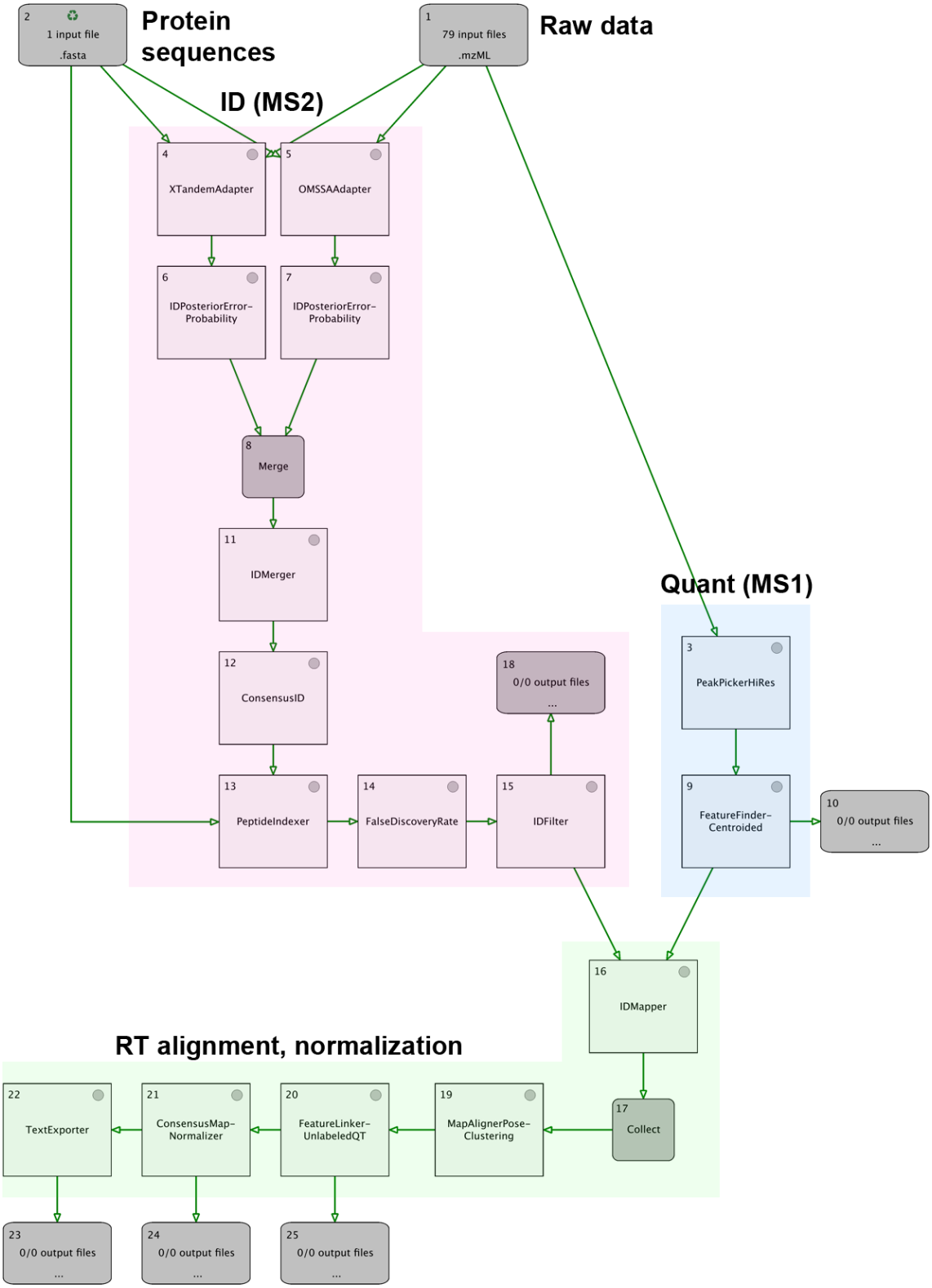

Figure 5.3: TOPPAS workflow for labelfree quantification used in the forensics study. Reprinted from Dammeier et al.[112]

5.2.6 Finding Marker Proteins

As a first step towards finding potential markers among the identified proteins, we sorted identifications by their absolute Pearson correlation coefficient with each of the five investigated organs. With a total of $n = 79$ samples, the correlation between a protein identification p_i and organ o_j is computed as the Pearson correlation coefficient between the two binary vectors P_i and O_j, where $P_i = (p_{i1}, \ldots, p_{in}) : p_{ik} = 1$ if protein i was identified in sample k, else 0, and $O_j = (o_{j1}, \ldots, o_{jn}) : o_{jk} = 1$ if sample k originates from organ j, else 0.

In a first attempt to find potential tissue markers, we searched for protein expression profiles correlating with organ assignment of the samples. We constructed expression profile vectors P_i for each identified protein i, describing in which sample protein i was identified. $P_i = (p_{i1}, \ldots, p_{in}), p_{ik} \in \{0, 1\}$ is a binary vector where $p_{ik} = 1$ if protein i was identified in sample k. Similarly, we define sample-organ assignment vectors $O_j = (o_{j1}, \ldots, o_{jn}), o_{jk} \in \{0, 1\}$, where $o_{jk} = 1$ if sample k stems from organ j and 0 otherwise.

For proteins expressed in specific organs only, the expression profile of the protein across all samples should coincide with the organ assignment vector of the respective organ: proteins should be identified exactly in those samples derived from the respective tissue. Obviously, incorrect or incomplete identifications (false positives, false negatives) as well as simultaneous expression in different tissues complicate this. We thus used Pearson correlation to rank similarities between expression profiles and organ assignments. Proteins relevant as markers for a specific organ should have a high correlation with their respective organ assignment vectors.

5.2.7 Classification

We evaluated the classification performance on our bovine data set using different machine learning algorithms (support vector machines (SVM)[133], random forests[134], Gaussian naïve Bayes, and multinomial naïve Bayes[135]) implemented in scikit-learn[129,130]. We compared prediction performance when using only qualitative attributes (protein identified / not identified) and when using a combination of qualitative and quantitative attributes (MS feature intensities). Hence, samples were represented as attribute vectors composed of binary qualitative attributes and [0,1]-scaled real-valued quantitative attributes of the detected peptide feature intensities. In order to prevent overfitting of the model to noisy features, we reduced the complexity by removing quantitative attributes that could not be identified as well as those that were quantified in less than ten out of 79 samples.

To assess classification performance, we performed 5-fold cross-validation by stratified random sub-sampling of a training set consisting of 80% of the samples and a test set containing the remaining 20%. Accuracy on the training set as well as the classification performance of the test set was reported. To reduce the complexity of the model and to investigate how well classification works when only a few marker proteins are considered, we tried selecting

only the top $N \in \{1, 3, 5, 10, 20, 100\}$ proteins with the highest Pearson correlation coefficients for each of the five investigated organs respectively, and used only those N attributes for classification. To prevent overfitting of the model, attribute selection as well as the selection of model parameters was performed only on the training data set, which is independent of the test data set used for performance evaluation. This was done using a nested 5-fold cross-validation, where the training set is again split into an inner training set and the remaining samples are used as an inner validation set. Attributes and model parameters yielding the best average performance on the validation set were used to actually train the model on the entire outer training set and finally make predictions on the outer test set, on which the performance was evaluated. The entire procedure was repeated 1,000 times. We report the average classification accuracy (percentage of correctly classified instances) on the test data set together with the standard error of the mean (SEM).

5.2.8 Forensic Use Case

Evidence of a recent case of murder (five projectiles filed as court exhibit numbers “1-3”, “1-12”, “1-13”, “1-14”, and “1-17”) was received from the district public prosecution authority (Staatsanwaltschaft Tübingen, AZ: NN/2012 (06.03.2012)) in blinded fashion. As usual for homicide cases, a standard autopsy of the body was performed by the forensic examiner, and results regarding the bullet channels were shared after the proteomic analysis had been performed, and after an organ assignment had been proposed.

5.3 Results

5.3.1 Proteomic Analysis of Organic Debris on Bullets Allows Discrimination of Penetrated Organs

In total, we analyzed 79 bovine tissue samples that originated from five distinct organs, i.e., heart, kidney, liver, muscle and lung, using the workflow as described in Section 5.2 in detail. In brief, to simulate the hit of a projectile, metal pieces were manually pressed through bovine tissues. This experimental setting was used to obtain sufficient amount of data for a comprehensive statistical analysis. Biological material was treated with trypsin directly on the surface of the metal pieces, and the resulting peptides from each sample were measured in separate three-hour LC-MS/MS runs, resulting in 79 raw datasets. We also included several controls for assessing the influence of protein degradation over time (at least one metal piece per organ was kept at ambient temperature for a minimum of three weeks before being processed), and the influence of the metal composition (use of different common bullet types, i.e., made out of different metal alloys, with the identical organic material). After processing the raw data, we detected an average of approximately 14,000 peptide features per sample. By mapping

identified MS/MS spectra to features we identified a total of 1,756 proteins. The measurements exhibited decent reproducibility as demonstrated by a computation of the overlap of protein identifications across all replicates for each organ. On average, 30% of the proteins identified in any of the replicates of an organ could be identified in at least half of the replicates of that organ (heart 33%, kidney 22%, liver 40%, lung 20%, muscle 35%).

Furthermore, we investigated the influence of different metal compositions that are commonly used as bullet bodies. To this end, bovine liver was penetrated with four different bullet types, and the routine digestion protocol was performed. A comparison of the protein identifications revealed a reasonable overlap between the four samples: 50% (478 out of 955) of the proteins identified in any of the samples could be reproducibly identified across all four samples, 66% (626 out of 955) in at least three out of four. Hence the protocol can be assumed to be robust also with respect to different metal alloys of projectiles.

Principle component analysis (PCA) was used to assess the degree of similarity between the different protein profiles. A clustering of peptide feature intensities was performed to visualize the structure of the quantitative data. Using all features detected in any map and aligned from map to map, we were able to construct a PCA input matrix of 79 samples and 282,640 peptide feature intensities in total. Figure 5.4 shows the scores plot of the PCA. PCA clustering revealed strong separation power using the first three components, which by themselves account for almost 60% of the total variance in the data set. The resulting clusters allow for a clear visual separation of all muscle samples and almost all heart samples (four samples show overlap with lung profiles). While the distribution of samples for lung, liver and kidney show more overlap than heart and muscle, the centers of the respective clusters are clearly drawn apart.

Subsequently, we evaluated the classification performance on this data set using support vector machines (SVM), random forests, Gaussian naïve Bayes, and multinomial naïve Bayes and compared their performances on sets of qualitative and quantitative attributes. All investigated classifiers achieved an excellent performance of well over 90% correctly classified instances on this data set. Notably, the simple multinomial naïve Bayes classifier achieved a surprisingly good performance of more than 99% correctly classified instances even when only the top three discriminating proteins per organ (hence 15 in total) were used as attributes, outperforming all other classifiers in the majority of cases. A comprehensive summary of the performance evaluation results can be found in Table 5.1.

Classification performance of the multinomial naïve Bayes classifier using only the top 1, 3, and 5 highest-correlated attributes is illustrated in Figure 5.5. We compared the performance of our approach when using only qualitative attributes (“ID only”) and when considering both qualitative and quantitative attributes (“ID + quant”). In the “ID only” attribute selection process, the top N highest-correlated protein identifications for each of the 5 investigated organs, hence 5 N in total, were selected. For “ID + quant”, we additionally selected the top N highest-correlated quantitative attributes (MS feature intensities), hence 10 N attributes in

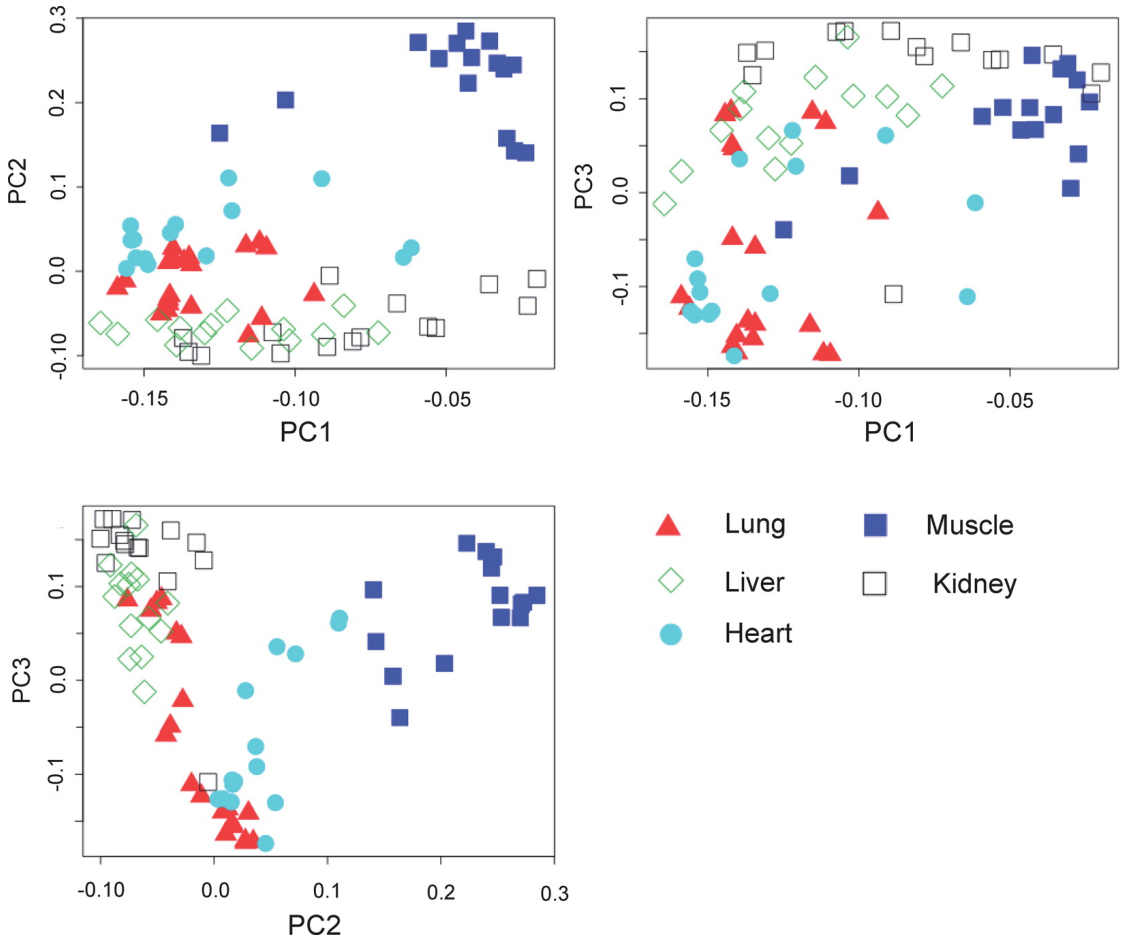

Figure 5.4: Principal component analysis of protein profiles from 79 bullet surfaces derived from penetration experiments on five different solid bovine organs. The first three components are visualized and exhibit a clustering of organ-specific profiles. Organ assignments are represented by colors, as indicated. Reprinted from Dammeier et al.[112]

Attributes		SVM		Random Forest		Gaussian NB		Multinomial NB	
Number	Kind	Test set	Train set	Test set	Train set	Test set	Train set	Test set	Train set
Top 1	Quant	86.91% (±0.25%)	96.25% (±0.25%)	86.63% (±0.27%)	95.19% (±0.27%)	**89.76% (±0.24%)**	95.22% (±0.24%)	87.13% (±0.24%)	94.61% (±0.24%)
	ID	90.46% (±0.20%)	97.04% (±0.20%)	91.44% (±0.20%)	97.13% (±0.20%)	90.46% (±0.20%)	96.72% (±0.20%)	**94.72% (±0.16%)**	97.35% (±0.16%)
	Both	96.28% (±0.15%)	98.96% (±0.15%)	94.21% (±0.19%)	98.45% (±0.19%)	95.91% (±0.16%)	98.31% (±0.16%)	**98.60% (±0.09%)**	99.90% (±0.09%)
Top 3	Quant	91.16% (±0.20%)	96.85% (±0.20%)	91.33% (±0.21%)	96.97% (±0.21%)	94.58% (±0.17%)	96.06% (±0.17%)	**95.97% (±0.14%)**	98.24% (±0.14%)
	ID	97.53% (±0.11%)	99.07% (±0.11%)	95.55% (±0.17%)	99.01% (±0.17%)	98.02% (±0.11%)	98.81% (±0.11%)	**99.11% (±0.09%)**	99.90% (±0.09%)
	Both	97.95% (±0.10%)	99.28% (±0.10%)	96.00% (±0.15%)	98.76% (±0.15%)	98.84% (±0.10%)	99.06% (±0.10%)	**99.81% (±0.03%)**	99.97% (±0.03%)
Top 5	Quant	92.18% (±0.18%)	97.57% (±0.18%)	91.62% (±0.20%)	97.20% (±0.20%)	95.00% (±0.17%)	95.92% (±0.17%)	**97.71% (±0.10%)**	99.30% (±0.10%)
	ID	98.18% (±0.10%)	99.25% (±0.10%)	96.62% (±0.16%)	99.04% (±0.16%)	99.01% (±0.09%)	99.16% (±0.09%)	**99.11% (±0.07%)**	99.95% (±0.07%)
	Both	98.49% (±0.09%)	99.34% (±0.09%)	96.31% (±0.14%)	98.75% (±0.14%)	99.21% (±0.08%)	99.30% (±0.08%)	**99.86% (±0.03%)**	99.97% (±0.03%)
Top 10	Quant	92.81% (±0.18%)	97.81% (±0.18%)	92.99% (±0.19%)	97.79% (±0.19%)	95.48% (±0.16%)	96.56% (±0.16%)	**98.41% (±0.09%)**	99.67% (±0.09%)
	ID	98.45% (±0.09%)	99.36% (±0.09%)	97.88% (±0.13%)	99.12% (±0.13%)	**99.46% (±0.06%)**	99.29% (±0.06%)	98.71% (±0.08%)	99.84% (±0.08%)
	Both	98.69% (±0.08%)	99.53% (±0.08%)	96.60% (±0.14%)	98.81% (±0.14%)	**99.63% (±0.05%)**	99.45% (±0.05%)	99.55% (±0.05%)	99.87% (±0.05%)
Top 20	Quant	93.12% (±0.17%)	98.01% (±0.17%)	94.29% (±0.17%)	98.15% (±0.17%)	97.24% (±0.13%)	98.63% (±0.13%)	**98.53% (±0.09%)**	99.79% (±0.09%)
	ID	98.37% (±0.09%)	99.47% (±0.09%)	98.16% (±0.10%)	99.06% (±0.10%)	98.74% (±0.08%)	98.97% (±0.08%)	**99.02% (±0.07%)**	99.46% (±0.07%)
	Both	98.62% (±0.08%)	99.57% (±0.08%)	96.84% (±0.14%)	98.91% (±0.14%)	98.78% (±0.08%)	99.07% (±0.08%)	**99.21% (±0.07%)**	99.77% (±0.07%)
Top 100	Quant	93.31% (±0.18%)	98.00% (±0.18%)	95.36% (±0.15%)	98.11% (±0.15%)	**99.85% (±0.03%)**	99.97% (±0.03%)	99.22% (±0.07%)	99.90% (±0.07%)
	ID	99.84% (±0.03%)	99.95% (±0.03%)	99.19% (±0.07%)	99.71% (±0.07%)	**99.88% (±0.03%)**	99.82% (±0.03%)	99.12% (±0.07%)	99.54% (±0.07%)
	Both	**99.94% (±0.02%)**	99.97% (±0.02%)	97.36% (±0.13%)	99.01% (±0.13%)	99.89% (±0.03%)	99.86% (±0.03%)	99.61% (±0.05%)	99.88% (±0.05%)
All (25,872)	Quant	96.55% (±0.14%)	95.89% (±0.14%)	97.28% (±0.12%)	97.52% (±0.12%)	**98.78% (±0.08%)**	98.36% (±0.08%)	96.22% (±0.15%)	95.95% (±0.15%)
All (1,756)	ID	**99.98% (±0.01%)**	99.94% (±0.01%)	99.41% (±0.06%)	99.80% (±0.06%)	99.96% (±0.02%)	99.87% (±0.02%)	98.55% (±0.09%)	98.78% (±0.09%)
All (27,628)	Both	99.72% (±0.04%)	99.13% (±0.04%)	97.78% (±0.12%)	98.09% (±0.12%)	**99.84% (±0.03%)**	99.35% (±0.03%)	98.14% (±0.11%)	97.91% (±0.11%)

Table 5.1: Average classification accuracies for different numbers and types of selected attributes using various classifiers. Adapted from Dammeier et al.[112]

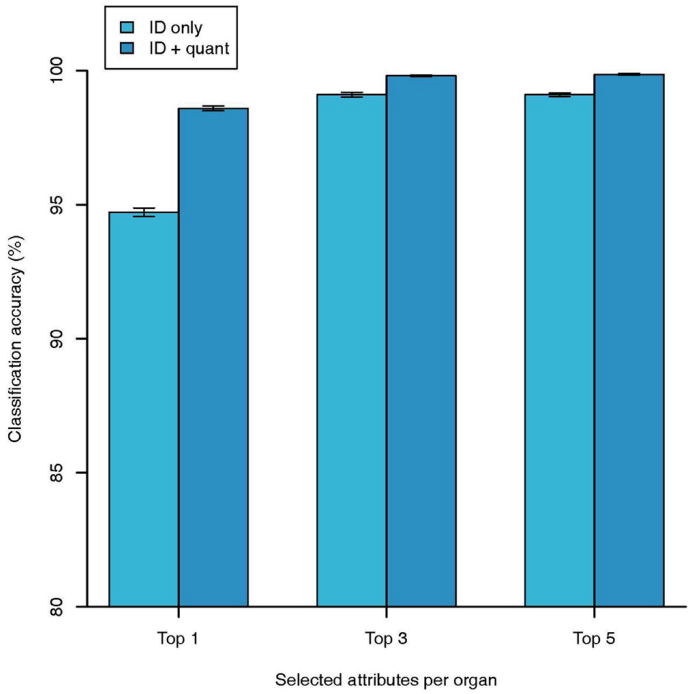

Figure 5.5: Average classification accuracy on the test set for different numbers and types of selected attributes using a multinomial naïve Bayes classifier. Even if only the presence or absence of only a single marker protein per organ is used as a criterion, we obtain almost 95% correct organ assignments on the test set. Adding quantitative information significantly improves the performance even further. No significant improvement is observed after adding more than the top three attributes per organ. Error bars represent the SEM. Reprinted from Dammeier et al.[112]

total were used for classification. Even if only the presence or absence of only a single marker protein for each organ is considered, we already achieve a classification accuracy of 94.7% (±0.16%). If we add the quantitative information, we see a significant improvement to 98.6% (±0.09%). Using the top three attributes, we reach 99.1% (±0.09%) with qualitative attributes and even 99.8% (±0.03%) when the quantitative information is added. Selecting the top five or more attributes does not significantly improve the classification performance any further. Table 5.2 shows the confusion matrix for the classification results using the multinomial naïve Bayes classifier on the top 3 selected ID and quantitative features per organ.

Discriminating proteins were ranked according to their Pearson correlation factors. The top-ranked proteins for the entire dataset, together with their Pearson correlation coefficients as well as p-values for organ specificity computed using Fisher's exact test, are listed in Table 5.3.

True organ \ Predicted organ	Kidney	Lung	Liver	Muscle	Heart
Kidney	3000	0	0	0	0
Lung	0	3999	0	0	**1**
Liver	0	0	3000	0	0
Muscle	0	0	0	3000	0
Heart	0	0	**29**	0	2971

Table 5.2: Confusion matrix for classification results on the test set using the multinomial naïve Bayes classifier on the top 3 selected ID and quantitative features per organ. Numbers are the sum over all 1000 randomized repetitions of the stratified 5-fold cross-validation. Hence, each row sum equals 1000 times the number of samples originating from the corresponding organ per repetition. Numbers on the main diagonal indicate correct classifications, whereas all other numbers represent misclassifications. Adapted from Dammeier et al.[112]

5.3.2 Projectiles of Shooting Experiments Adsorb Sufficient Amounts of Protein

As the experiments using isolated bovine organs in combination with manual bullet penetration were artificial with regard to a real shooting scenario we sought to address the crucial question whether manual penetration is comparable to high-speed penetration from a proteomics perspective.

In a real shooting, bullets of common handguns would hit the body at a velocity of up to 500 m/s. Thus the projectile has a high amount of kinetic energy that is mainly needed to perform work at the target, i.e. to penetrate and to cavitate. The ideal effect of a shot would be instant incapacitation due to the hit of a vital structure, preferably heart or brain. Although the kinetic energy of a non-deforming projectile is proportional to the volume displaced by cavitation this rule applies ideally for a non-elastic medium[137]. Tissue, however, is highly elastic so that a lot of kinetic energy is applied to elastic displacement. More energy is lost to the deformation of the bullet. Thus besides the energy spent for the rupture of the tissue (by hydrodynamic pressure) only a minor portion of energy is lost to friction or in consequence to heat. Therefore the extent of heat development is far insufficient to carbonize proteins that are exposed to the surface of the penetrating bullet. Otherwise it would not have been possible to identify tissue-specific proteins by immunodetection on the surface of projectiles in shooting experiments earlier[123]. Another issue is probably the contact time of projectile and penetrated tissue. Due to the high velocity this is rather short for each organ, which equals to a specific "protein pool" that is crossed.

Heart				
Protein Name	**Accession (bovine)**	**Accession (human)**	*R*	*p*
Myosin binding protein C, cardiac-type	Q0VD56_BOVIN	MYPC3_HUMAN	0.92	6.99e-10
Glycogen phosphorylase, brain form	PYGB_BOVIN	PYGB_HUMAN	0.83	5.05e-09
Troponin I, cardiac muscle	TNNI3_BOVIN	TNNI3_HUMAN	0.79	3.66e-06
Myosin light chain 3	MYL3_BOVIN	MYL3_HUMAN	0.76	1.60e-07
Pyruvate dehydrogenase E1 subunit beta	ODPB_BOVIN	ODPB_HUMAN	0.75	6.24e-06
Calpastatin	Q9XSX1_BOVIN	ICAL_HUMAN	0.74	4.27e-05
NADH dehydrogenase 1 alpha subunit 4	NDUA4_BOVIN	NDUA4_HUMAN	0.74	4.27e-05
Cytochrome c oxidase subunit 4 isoform 1	COX41_BOVIN	COX41_HUMAN	0.70	2.11e-04
Kidney				
Protein Name	**Accession (bovine)**	**Accession (human)**	*R*	*p*
Calbindin	CALB1_BOVIN	CALB1_HUMAN	1	3.26e-13
Na(+)/H(+) exchange regulatory cofactor NHE-RF3	NHRF3_BOVIN	NHRF3_HUMAN	1	3.26e-13
Low-density lipoprotein receptor-related protein 2	F1N6H1_BOVIN	LRP2_HUMAN	1	3.26e-13
Villin 1	Q5E9Z3_BOVIN	VILI_HUMAN	0.96	2.12e-11
Retinyl ester hydrolase type 1	Q5MYB8_BOVIN	Q8TDZ9_HUMAN	0.92	3.13e-10
Membrane metallo-endopeptidase variant 2	E1BPL8_BOVIN	NEP_HUMAN	0.92	6.99e-10
Phosphotriesterase-related protein	PTER_BOVIN	PTER_HUMAN	0.83	2.65e-07
Plastin-1	PLSI_BOVIN	PLSI_HUMAN	0.79	3.66e-06
Liver				
Protein Name	**Accession (bovine)**	**Accession (human)**	*R*	*p*
Hydroxymethylglutaryl-CoA synthase, mitochondrial	HMCS2_BOVIN	HMCS2_HUMAN	0.96	1.41e-12
Carbamoyl-phosphate synthase, mitochondrial	F1ML89_BOVIN	CPSM_HUMAN	0.92	2.12e-11
Phenylalanine-4-hydroxylase	PH4H_BOVIN	PH4H_HUMAN	0.92	2.12e-11
3-oxo-5-beta-steroid 4-dehydrogenase	E1BBT0_BOVIN	AK1D1_HUMAN	0.92	2.12e-11
Catechol O-methyltransferase	COMT_BOVIN	COMT_HUMAN	0.92	2.12e-11
Acetyl-CoA acetyltransferase	Q17QI3_BOVIN	THIC_HUMAN	0.92	9.31e-11
Cytochrome P450 2E1	CP2E1_BOVIN	CP2E1_HUMAN	0.92	9.31e-11
Dimethylaniline monooxygenase	G5E5R0_BOVIN	FMO1_HUMAN	0.92	9.31e-11
Lung				
Protein Name	**Accession (bovine)**	**Accession (human)**	*R*	*p*
Plastin-2	F1MYX5_BOVIN	PLSL_HUMAN	0.93	1.21e-12
Cathelicidin-4	CTHL4_BOVIN	CAMP_HUMAN	0.90	2.50e-11
Tubulin beta chain	TBB5_BOVIN	TBB5_HUMAN	0.90	2.23e-11
Cysteine and glycine-rich protein 1	CSRP1_BOVIN	CSRP1_HUMAN	0.87	3.94e-10
Prostaglandin F synthase 2	PGFS2_BOVIN	AK1C1_HUMAN	0.87	3.94e-10
Myosin light polypeptide 6	MYL6_BOVIN	MYL6_HUMAN	0.84	1.46e-09
Calpain-2 catalytic subunit	CAN2_BOVIN	CAN2_HUMAN	0.83	5.05e-09
Alpha-actinin-1	ACTN1_BOVIN	ACTN1_HUMAN	0.83	5.05e-09
Muscle				
Protein Name	**Accession (bovine)**	**Accession (human)**	*R*	*p*
Bridging integrator 1	Q2KJ23_BOVIN	Q9BTH3_HUMAN	1.00	3.26e-13
Myosin-binding protein C, slow-type	A6QP89_BOVIN	MYPC1_HUMAN	1.00	3.26e-13
Fructose-1,6-bisphosphatase isozyme 2	F16P2_BOVIN	F16P2_HUMAN	1.00	3.26e-13
Myosin-binding protein C, fast-type	E1BNV1_BOVIN	MYPC2_HUMAN	1.00	3.26e-13
Troponin C, skeletal muscle (fast type)	Q148C2_BOVIN	TNNC2_HUMAN	1.00	3.26e-13
Myosin regulatory light chain 2, skeletal muscle	MLRS_BOVIN	MLRS_HUMAN	0.96	5.21e-12
Nebulin	F1MQI3_BOVIN	NEBU_HUMAN	0.92	4.43e-11
PDZ and LIM domain protein 3	PDLI3_BOVIN	PDLI3_HUMAN	0.92	4.43e-11

Table 5.3: Top discriminating protein identifications per organ. Protein identifications were ranked by absolute Pearson correlation coefficient (Corr) between presence of a protein identification and organ affiliation over all samples. In addition to the Uniprot accessions of bovine proteins, we provide accessions of one homologous human protein each as determined by executing the BLAST algorithm in protein-to-protein mode[136]. Adapted from Dammeier et al.[112]

To investigate the applicability of our approach to real-world scenarios, shooting experiments were performed using four of the isolated bovine organs, namely liver, lung, kidney and heart, which were imbedded in a gelatin matrix. The collected projectiles were processed using our established protocol, and proteomic data analysis was performed. Subsequently, the protein identification overlap was computed between samples taken from shot bullets and manually pushed bullets. For each of the four shot bullets, we calculated the average protein identification overlap with the reproducible protein IDs (those found in at least half of the replicates) for the corresponding organ from the manual penetration data set. Despite the very different nature of sample taking, we could identify a substantial number of proteins on the shot bullets that have also been reproducibly identified in the manual penetration data set: 46% (166 out of 360) for the liver sample, 43% (96 out of 222) for lung, 20% (35 out of 174) for kidney, and 17% (11 out of 64) for heart.

5.3.3 Proteomics of Shot Projectiles Allows Organ Classification

To evaluate the applicability of our classification model, which was trained on the entire data set of manually penetrated bovine organs, to the shooting scenario, we sought to predict which bullet was shot through which organ. Here, we applied the classifier and attribute combination yielding the best performance on the training set, namely the multinomial naïve Bayes classifier using the top five attributes per organ. The classification was correct for two out of the four shooting samples, namely liver and lung. The heart sample was incorrectly classified as muscle, kidney was classified as lung. As the heart represents a specialized muscle, this result is to a certain extent not completely wrong. However, it exemplified that the heart-specific marker proteins discriminating heart from muscle as determined by our model were not identified here. One reason for the misclassification of the heart sample becomes obvious: the overall number of protein identifications on this bullet was very small. Furthermore, both misclassified samples exhibited a considerably smaller protein identification overlap with the corresponding samples from the manual penetration data set. Nevertheless, these results confirmed partially that manual penetration and shooting experiments lead to a comparable outcome.

5.3.4 Application to a Case of Homicide

In order to test and validate protein signatures established from bovine organs towards forensic application, we analyzed a recent case of murder, in which the coroner was not able to fully reconstruct the shooting by analysis of the bullet channels during autopsy. For case details, refer to the file reference in the Experimental Procedures. In brief, a 63-year-old male was shot by his wife inside a car while driving. After several hits he was still able to leave the car, though followed by his assailant. In the end, the victim was hit by a shot in the head and died in the street. Although the general solution of this case of murder was clear, from

a forensic point of view, it was of specific interest which of the multiple shots was the first lethal one, i.e., inside the car or outside. To investigate this, two projectiles recovered from inside the car, filed as court exhibits numbers 1-12 and 1-13, and three projectiles found in the street, filed as court exhibits numbers 1-14, 1-17, and 1-3, were taken for proteomic analysis as described. It is noteworthy to mention that sample 1-3 was found in a puddle of blood on the asphalt, and that all court exhibits were analyzed in blinded fashion, i.e. without knowledge of details of the autopsy. The database search revealed a total of 269, 180, 33, 34, and 84 protein identifications for samples 1-12, 1-13, 1-14, 1-17, and 1-3, respectively. A machine learning-based approach was performed with the protein identification data as described for the bovine test case. Unfortunately, the outcome was not significant (data not shown), which was not surprising since the projectiles of interest had penetrated multiple organs. Therefore contamination might have been a major issue. Moreover, most likely not all of the penetrated organs were covered by the bovine experiments, and, in consequence, by our model.

The autopsy of the body revealed three full primarily abdominal penetrations and one graze of the right shoulder as well as two bullets, of which one was lodged in the brain and the other in the right thigh. Only bullets found outside the body were analyzed by our proteomic approach without knowledge of anatomical details of the autopsy. Additionally, the forensic examination exhibited that the three abdominal penetrations could be characterized further as (i) penetration of liver and right atrium, (ii) penetration of upper arm, lung and heart, and (iii) penetration of aorta and trachea. In order to reveal the correlation of bullet channels to projectiles, we followed a different strategy and tried to map the protein identifications of the projectiles to the top-ranked ten organ-discriminating proteins. The results are summarized in Table 5.4. Sample 1-12 was the only one that exhibited three of the top-ranked discriminating proteins for liver. In combination with the identification of four top heart-discriminating proteins, this indicated that 1-12 maps best to the bullet channel of penetration (i), since it was the only one affecting the liver. In addition, as sample 1-13 exhibited a number of heart-specific plus lung-specific proteins, but not any liver-discriminating proteins, we suggest that this projectile can be matched to penetration defect (ii).

Because the remaining projectiles did not exhibit any of the top-ranked organ-discriminating proteins, and due to the fact that at least sample 1-3 showed a reasonable number of 84 protein identifications, we set out to adapt our approach by taking major contaminations into account. Due to the special setting of a forensic situation contamination is generally one of the major issues. A projectile that hit a body traverses in most cases fabric, skin, connective tissue, and it certainly gets in contact with blood. Eventually it even leaves the body and may be found in dirt or dust. With regard to standard proteomic analyses blood, which is available in every organ, seems to be the most relevant general contaminant. As projectile 1-3 was found in a puddle of blood, it was indicated to subtract a general “blood-projectile-profile” that was generated by the contamination of a naïve bullet with human blood and proteomic analysis

Organ	Protein accession (human)	Correlation (bovine)	Protein ID*				
			1-12	1-13	1-14	1-17	1-3
Heart	MYPC3_HUMAN	0.92	7				
	PYGB_HUMAN	0.83		13			
	TNNI3_HUMAN	0.79		3			
	MYL3_HUMAN	0.76	6	8			
	ODPB_HUMAN	0.75	2				
	NDUA4_HUMAN	0.74	2				
Liver	HMCS2_HUMAN	0.96	6				
	CPSM_HUMAN	0.92	45				
	FMO1_HUMAN	0.92	3				
Lung	TBB5_HUMAN	0.90	3	2			
	MYL6_HUMAN	0.84	2	2			
	ACTN1_HUMAN	0.83	6	12			
Muscle	MYPC1_HUMAN	1		13			
	F16P2_HUMAN	1	7				
	TNNC2_HUMAN	1	2	2			
	MLRS_HUMAN	0.96		2			
	NEBU_HUMAN	0.92		7			
	PDLI3_HUMAN	0.92		2			

Table 5.4: Detailed protein identification analysis of court exhibits of a homicide case. The overlap of protein identifications from the homicide case (court exhibits no. 1-12, 1-13, 1-14, 1-17, 1-3) with the top ten most organ-discriminating proteins from the bovine experiments as listed in Table 5.3 is exhibited. Only proteins that were actually identified in the homicide case are shown. This information was used for manual forensic interpretation (e.g., exclusion of specific organ penetrations). Adapted from Dammeier et al.[112]

*Protein identification parameters (numbers of identified peptides are shown) were kept at minimum of 2 identified peptides/protein and at a protein identification threshold of 99%.

after storage for 3 weeks (see data in the PRIDE repository). The remaining list of 12 proteins still exhibited a prominent number of blood-specific proteins (e.g., hemoglobin subunits), yet it additionally revealed a number of proteins that are very abundant in muscle. However, most of them are also found in heart tissue (e.g., myosin isoforms 2 and 7 and creatine kinase M). Furthermore, strong evidence that projectile 1-3 can be matched rather to penetration (iii) than to the grace shot, for which the projectile was also not assigned, is derived from the fact that vimentin was identified on its surface. In the analysis of the bovine organ data, vimentin exhibited a moderate correlation with lung and heart, but an anti-correlation with liver and kidney, and hardly any correlation with muscle. In summary, we found protein signatures determining organ specificity of bullet penetration for three projectiles. These signatures correlated with the anticipated bullet channels determined by classical forensic autopsy. Moreover, a contamination with blood could be experimentally evaluated to some extent.

5.4 Discussion

We present the first study applying mass spectrometry-based proteomics to reveal organ-specific protein expression for forensic evidence. So far, forensic examinations in this context have been based preferably on a few marker proteins that were primarily predefined (e.g., by applying immunodetection[122]). The method described here extends the notion of markers to revealing protein profiles using comprehensive proteomic datasets. Previous studies have shown that proteomic technologies are able to reveal organ-specific protein expression and/or specific patterns of protein modification[125,138].

Furthermore, with respect to forensic applications, mass spectrometry-based analysis of protein debris on evidence exhibited the potential to precisely define the type of biological material (e.g., blood, saliva, or feces[139]). We used a range of bioinformatics methods to reveal a data-driven clustering of organs based on their raw MS profile and assigned protein IDs. Principle component analysis, based on peptide feature intensities alone, already revealed a significant separation of most profiles originating from different organs. This could generally be anticipated with respect to the experimental design chosen. In addition, we have shown that using both quantitative and protein identification information permits the classification of samples with an overall accuracy of over 99% for the five organs investigated. This suggests a clear conservation of the underlying organ-specific protein expression profile from the collected bullet to the mass-spectrometric data. As no significant improvement of classification accuracy was observed after adding more than the top three features this might also be a promising sign to develop a relatively inexpensive targeted assay for forensic testing. However, we have to be cautious about that because of the extremely artificial situation of the manual penetration experiments. Furthermore we were able to show that the organ-specificity could be partially replicated in shooting experiments. In doing so the level of congruence seemed to be dependent

on the type of organ (e.g., for liver it was acceptable (46%), whereas for heart it was rather poor (17%)). It could be speculated that this finding is somehow related to the different textures of the organs, which might have different effects on a rapidly penetrating bullet.

With respect to the biochemical nature of the most relevant organ-discriminatory proteins, we found either structural proteins or enzymes that could be related to the physiological role of the organ. In general, it was assumed that the identified and discriminating proteins are the outcome of different intrinsic, external, and analytical factors. Intrinsic factors are relative protein abundance, robustness against degradation, hydrophobicity or adhesive properties of the proteins. External factors are influencing variables outside the biological context (e.g., environmental factors like temperature and humidity). Especially the matrix background of forensic samples represents a major analytical challenge during analysis. Taking these factors into account, it is also of scientific interest to elaborate what kind of substantial biochemical or physiological organ properties are reflected by the most discriminating proteins as summarized in Table 5.3. Hence we sought to interpret the discriminatory protein data of heart and muscle used as an example for closely related organs, as well as of liver used as an example for a non-muscular inner organ (In the following general protein characteristics are referenced according to their UniProt annotations (`www.uniprot.org`)).

Proteins that were suitable for classifying skeletal muscle successfully exhibit the organ's special role in motility. Prominent examples are myosin-binding proteins, troponin C, myosin regulatory light chain, and also nebulin, a stabilizing, muscle-specific protein enhancing the integrity of sarcomers and membranes of myofibrils. In contrast, bridging integrator 1 or amphiphysin, a protein that exhibited a correlation factor of 1, is highly expressed in nerve terminals, probably being involved in synaptic vesicle endocytosis thus reflecting the muscles' high level of innervation. Although being ubiquitously distributed, some isoform subtypes have been reported to be very tissue specific (e.g., in brain or muscle[140]). Furthermore, since skeletal muscles consume high amounts of energy, the list of muscle-discriminating proteins is completed by enzymes that are involved in energy-generating processes (e.g., fructose-1,6-bisphosphatase isozyme 2, which plays a prominent role in glycolysis). Strikingly, the identified discriminatory protein profiles on the projectile surfaces allow for distinguishing the skeletal muscle from the heart muscle, a muscle that is both powerful and persevering. Again, as for muscle tissue, proteins directly or indirectly involved in contractility are prominent discriminators (e.g., myosin binding protein C and myosin light chain 3). However, the proteomic analysis is able to detect cardiac-specific variants of those, the most important representative of this class being troponin I, which has been discussed as a (blood) biomarker for heart-related disorders like myocardial infarction[141]. An additional set of cardiac discriminators derive from key proteins involved in energy metabolism, i.e., glycogen phosphorylase and pyruvate dehydrogenase E1. Interestingly, a subgroup of this class of top discriminators consists of mitochondrial proteins, i.e., subunit 4 of the NADH dehydrogenase alpha subcomplex and

subunit 4 of cytochrome C oxidase, both referring to the high impact of mitochondrial energy generation in the heart.

In contrast, discriminating protein signatures for liver contain specific enzymes associated with steroid metabolism, (e.g., HMG-CoA synthase and 3-oxo-5-beta-steroid 4-dehydrogenase). These proteins feature the extraordinary metabolic activity in lipid synthesis of the liver. Furthermore, they are accompanied by a prominent enzyme of fatty acid metabolism, namely, acetyl-CoA acetyltransferase. The generation of steroids and the metabolism of lipids are important central hepatic functions. Another central function of the liver is detoxification or degradation of waste molecules. These features explain the presence of various key metabolizing enzymes, including dimethylaniline monooxygenase, cytochrome P450 2E1, and catechol O-methyltransferase among the top liver-discriminating proteins. Furthermore, proteins primarily involved in the degradation of amino acids and amines complete this picture (e.g., phenylalanine-4-hydroxylase and carbamoyl-phosphate synthase, an enzymatic component of the urea cycle). It is highly remarkable that we detected neither the classical (diagnostic) liver enzymes such as the aminotransferases AATM/AATC and ALAT, nor the liver-specific alcohol dehydrogenases among the highest discriminating proteins.

Finally, the top discriminating proteins were searched in a data base that contains tissue and organ information in order to benchmark our experimental findings. Therefore we chose ProteomicsDB, a comprehensive atlas of the human proteome which has been released recently[142]. Overall, a good correlation with the organ-specific expression level could be found for heart, kidney and liver, whereas muscle proteins are not annotated as such in the database (data not shown). The comparison also indicates that the forensic applicability of precompiled organ proteomes is limited and therefore the experimental determination of specific datasets as reported here should be preferred.

As the statistical findings in the bovine experimental case exhibited good correlations between bullets and organs (see 5.3), and due to the fact that our experimental setting could function as model for shootings, it could be assumed that a similar approach would also be successful in real forensic cases. However, in the investigated homicide case it was not possible to achieve comparable accuracy solely by statistical classification in a blinded approach. Nevertheless, the knowledge of organ-discriminating protein patterns fostered the biochemical and forensic interpretation of the proteins that were actually identified in the homicide case. Thereby valuable additional information was provided for the coroner towards the assignment of specific projectiles to distinct bullet channels. Thus we postulate that our method could set new standards in forensic analysis and, consequently, in jurisdiction. Although most cases will include additional challenges, such as contamination with various organic materials at the crime scene, or protein degradation as a consequence of late recovery or inadequate storage and asservation, in the real crime case, our approach enabled the assignment of projectiles for two out of three penetrating defects of the corpse successfully (Table 5.4). This improvement of

the forensic examination, and consequently, of the reconstruction of the whole case would not have been feasible without the utilization of the proteomic data set. However, due to multiple organ penetrations, it was necessary to combine and assign the top discriminating proteins manually. A potential reason for this could be the multi-organ situation that is very difficult to address experimentally. In addition, the consideration of case-related contaminations like the obvious contact with special tissues (one of the projectiles was found in a puddle of blood) increased the chance of an accurate organ assignment. However, we assumed not to find an important influence of ubiquitous tissues like fat or skin on our analyses. In particular, skin contaminations seem not to be highly relevant since contamination by dermal or epidermal proteins, such as keratins, is a general issue in proteome research[143]. Therefore we have chosen muscle tissue as potential major contamination control in our experiments, as well as the heart would be one of the principal targets of forensic investigation.

The study presented here gives a proof of principle for the application of mass spectrometry-based proteomics during forensic examination. Further improvements of the method in combination with a more comprehensive database of forensic reference samples have the potential for almost automatically assigning even multiple organs to a sample in future applications. At the same time, we suggest that the presented forensic proteomic method would also be applicable to cases of stabbing, in which very often many forensic uncertainties remain[119]. Finally, with the availability of personalized genome sequences the proteomics approach can also be further extended to identify not only the victim, but also the organs involved in one comprehensive analysis.

Chapter 6

Conclusion

Modern high-throughput technologies in proteomics produce vast amounts of experimental data. Tools for efficient and automated data analysis have become an absolute necessity. The wide range of available techniques for quantification and identification and the variety of different instrument types give rise to a broad range of computational challenges. Computational data analysis has become a key bottleneck of the overall workflow in today's biomedical studies. In the context of this thesis, we have developed novel algorithms and tools for automated data processing and efficient analysis of high-throughput LC-MS proteomics data and demonstrated their performance in various benchmark settings. Finally, we have described an exciting application of our work in the context of a LC-MS proteomics study in the field of forensic science.

The first major contribution presented in this thesis, the development of TOPPAS, the OpenMS proteomics pipeline assistant, has paved the way for what has now become one of the key concepts of OpenMS: reproducible data analysis using established, documented, and highly configurable computational workflows. TOPPAS is a GUI-driven dedicated OpenMS/TOPP workflow engine implemented in C++/Qt. Since OpenMS version 1.9, TOPPAS is included in every installation of the software package and allows non-computer scientists and bioinformaticians alike to rapidly set up data analysis workflows for mass spectrometric data. The entire data processing workflow, including all tool parameters, is stored in a single file. This facilitates the design, customization, parameter optimization, and documentation of entire data processing workflows and provides a convenient way for researchers to share established analysis workflows with the community. For developers, TOPPAS has become an invaluable tool for rapid prototyping and testing of novel tools and workflows. Due to its tight integration with the overall OpenMS framework and build system, all TOPP tools contained in the underlying OpenMS installation are automatically registered with TOPPAS and thus instantly available as workflow nodes. This is especially useful for prototyping workflows involving TOPP tools that are still in the early development phase, hence unpublished and

unavailable in the current stable releases of OpenMS or on other workflow platforms. For batch processing of high-throughput data on more powerful computing resources, mature workflows can be run without the GUI using the ExecutePipeline TOPP tool, allowing the deployment on dedicated compute servers, where a graphical environment may be unavailable. In addition to the TOPP tools included with OpenMS, arbitrary external command line tools can be integrated into TOPPAS workflows as well. This is achieved by providing TOPPAS with an XML-based configuration file describing the external tool's command line interface. This format has eventually inspired the design of its successor, the Common Tool Descriptor (CTD) format, which has now become a central component of our approach for wrapping external tools in other workflow platforms and has been suggested as a general community standard for simililar efforts across the field.

The presented OpenMS plugin for the popular KNIME workflow platform powered by the Generic KNIME Nodes (GKN) extension takes the idea of "holistic" data analysis workflows one step further: In addition to the data processing part using OpenMS/TOPP, the entire downstream statistical analysis, from "raw" processing results up to the generation of publication-ready figures and tables, can be designed, executed, and documented in the same place. A key argument for choosing KNIME over TOPPAS is the huge variety of available KNIME nodes for downstream statistical data analysis and visualization. TOPPAS, on the other hand, has better built-in support for parallel execution of processing jobs, and is thus better suitable in high-throughput settings where processing speed is crucial. While TOPPAS can make use of an arbitrary user-defined number of CPU cores for efficient parallel processing of queued jobs on a single compute server, there are cases where efficient data analysis of ultra-high-throughput data demands the compute power of a cluster, grid, or cloud environment. The KNIME2gUSE workflow converter is the latest addition to the repertoire of available workflow technologies for OpenMS/TOPP. It allows to easily convert KNIME workflows to the gUSE workflow language, enabling the seamless deployment of workflows designed and tested in the KNIME environment on powerful grid and cloud resources.

TOPPAS, KNIME-GKN, and KNIME2gUSE span a wide range of use cases and audiences. However, our experience has shown that a significant number of biologists and other non-computer scientists are reluctant to use a powerful modular workflow system when they want to perform standard analyses that can also be achieved using dedicated standalone solutions. In order to make OpenMS algorithms and workflows available to these users, we have developed a plugin for the popular vendor software platform Proteome Discoverer (PD; Thermo Scientific), adding community nodes for label-free quantification of peptides and proteins (LFQProfiler) and protein-RNA cross-linking data analysis (RNP^{xl}) to PD's repertoire of workflow nodes. The motivation for targeting this additional platform was to combine the efficiency of OpenMS algorithms and workflows with the convenience and integrated data analysis capabilities of Proteome Discoverer. PD's workflow concept represents a tradeoff between modularity, flexi-

bility, and configurability on the one hand and usability by non-expert users on the other hand. As a result, PD workflow nodes typically perform a series of computational steps that would be separated into individual modules in the OpenMS philosophy. LFQProfiler, for instance, performs run-wise feature detection in one PD node, and the rest of the label-free quantification workflow (mapping of identifications to quantified features, retention time alignment, feature linking, protein inference) in a downstream node. Only the most important parameters of the underlying OpenMS workflows are exposed to the user. LFQProfiler is based on a modified version of an established LFQ workflow described by Weisser et al.[19], and has shown to perform at least on par with other state-of-the-art tools for label-free quantification in a performance comparison on a public benchmark dataset[10]. RNP^{xl} is based on an approach by Kramer et al.[99] and represents the first software solution to date for identification of peptide-RNA cross-links including automatic localization at amino acid resolution and localization scoring. Due to the tight integration with PD, results can be interactively inspected using a customized peptide-nucleotide cross-link spectrum viewer including custom annotations for the detected fragment ion types.

In an effort to further improve on the quantification performance of OpenMS-based label-free quantification (LFQ), we have developed OptiQuant, a novel approach to LFQ for proteomics data. One of the key challenges in LFQ lies in reliable signal detection and quantification of raw data in datasets with high sample complexity. Partially overlapping signals within the same LC-MS run can lead to skewed quantification values or prevent features from being detected altogether, leading to missing data for the respective run. Traditional methods perform run-wise feature detection and then link corresponding features across maps. Feature detection usually involves simple local heuristics to select one out of several conflicting feature hypotheses. In contrast, OptiQuant uses mixed-integer programming to compute a globally optimal solution for the feature detection problem, considering signals from all maps at once. We wanted to investigate to which extent such an approach can improve overall sensitivity and quantification accuracy. To this end, we have compared OptiQuant's results to those of three state-of-the art LFQ solutions across both synthetic and experimental datasets. The first benchmark results are very encouraging, yet perhaps not quite impressive enough to justify the increased problem complexity inherent to OptiQuant's approach: OptiQuant clearly outperformed the other tools in terms of sensitivity and reproducibility on three complex synthetic datasets, while achieving a very good overall quantification accuracy. On the experimental iPRG 2015 LFQ challenge dataset[107], OptiQuant achieved lower sensitivity but the highest quantification accuracy of all tested tools. In the ranking of 48 results submitted by the challenge participants, OptiQuant's results would have been ranked 8th – a respectable result. The set of differential proteins reported by the OptiQuant workflow achieved a perfect precision of 1.0. Out of the 48 submitted solutions, only seven achieved an equally high precision, and only two of them had a higher recall than OptiQuant. While these initial benchmark results are

promising, a number of areas could benefit from further improvement. These include usability (reducing the number of unintuitive user parameters), the reliability of mass trace detection, the central scoring function for mass trace hypotheses at the heart of the optimization function, better handling of outlier traces resulting from mass trace linking errors, as well as processing speed. Beyond the current proof-of-concept implementation, we believe that OptiQuant can serve as a valuable general prototyping framework for related approaches.

Last, but not least, we have presented our findings from an LC-MS proteomics study in the field of forensics science. Here, we have successfully used TOPPAS workflows for label-free quantification, in combination with downstream statistical analysis and machine learning, to analyze the proteomes contained in traces of organic material remaining on projectiles after perforation of vital organs. Matching bullets to victims by means of DNA analysis has become a routine task in modern forensics, but it is rather difficult to determine which projectile caused the lethal injury in cases involving multiple shooters and bullet channels. In this proof-of-concept study, we were able to demonstrate that a sufficient amount of protein can be recovered from the surface of bullets after perforation of bovine organs, in both manual penetration and actual shooting experiments. Moreover, we have shown that the penetrated organ can be determined based on the characteristic expression profile of proteins found on the bullet. This was demonstrated using multiple machine learning classifiers in a stratified nested cross-validation setting. Classifiers were trained on identification and quantification results for a total of 79 label-free LC-MS experiments corresponding to biological and technical replicates of bullet samples after perforation of bovine organs. We could show that the perforated organ can be reconstructed with very high accuracy, even when the prediction is based only on a relatively small subset of characteristic proteins. The results of this proof-of-principle study are not yet generally applicable in everyday forensic practice. More work remains to be done before our method can be applied to human samples and meet the reliability standards of legal proceedings. An obvious practical obstacle is the acquisition of a sufficiently large training dataset of human samples. Moreover, our classification approach currently supports only the prediction of a single penetrated organ for each bullet. In practice, however, bullet channels can span multiple vital organs. It remains to be investigated how mixed tissues affect classification performance, and whether it is even feasible to reliably predict combinations of penetrated organs. Nevertheless, our study has clearly shown that the general approach is viable, and thus laid important groundwork for future efforts along these lines. In conclusion, we are confident that this thesis has contributed to the advancement of the field of computational mass spectrometry – and hopeful that it will one day help solve homicide cases routinely.

Bibliography

[1] I. H. G. S. Consortium, "Initial sequencing and analysis of the human genome," *Nature*, vol. 409, pp. 860–921, Feb. 2001. 1

[2] J. C. Venter, M. D. Adams, E. W. Myers, P. W. Li, and R. J. Mural, "The sequence of the human genome," *Science*, 2001. 1

[3] R. Staden, "A strategy of DNA sequencing employing computer programs," *Nucleic Acids Research*, vol. 6, pp. 2601–2610, June 1979. 1

[4] R. D. Fleischmann, M. D. Adams, O. White, R. A. Clayton, E. F. Kirkness, A. R. Kerlavage, C. J. Bult, J. F. Tomb, B. A. Dougherty, and J. M. Merrick, "Whole-genome random sequencing and assembly of Haemophilus influenzae Rd," *Science*, vol. 269, pp. 496–512, July 1995. 1

[5] M. D. Adams, S. E. Celniker, R. A. Holt, C. A. Evans, J. D. Gocayne, P. G. Amanatides, S. E. Scherer, P. W. Li, R. A. Hoskins, R. F. Galle, R. A. George, S. E. Lewis, S. Richards, M. Ashburner, S. N. Henderson, G. G. Sutton, J. R. Wortman, M. D. Yandell, Q. Zhang, L. X. Chen, R. C. Brandon, Y.-H. C. Rogers, R. G. Blazej, M. Champe, B. D. Pfeiffer, K. H. Wan, C. Doyle, E. G. Baxter, G. Helt, C. R. Nelson, G. L. Gabor, Miklos, J. F. Abril, A. Agbayani, H.-J. An, C. Andrews-Pfannkoch, D. Baldwin, R. M. Ballew, A. Basu, J. Baxendale, L. Bayraktaroglu, E. M. Beasley, K. Y. Beeson, P. V. Benos, B. P. Berman, D. Bhandari, S. Bolshakov, D. Borkova, M. R. Botchan, J. Bouck, P. Brokstein, P. Brottier, K. C. Burtis, D. A. Busam, H. Butler, E. Cadieu, Angela Center, I. Chandra, J. M. Cherry, S. Cawley, C. Dahlke, L. B. Davenport, P. Davies, B. de Pablos, A. Delcher, Z. Deng, A. D. Mays, I. Dew, S. M. Dietz, K. Dodson, L. E. Doup, M. Downes, S. Dugan-Rocha, B. C. Dunkov, P. Dunn, K. J. Durbin, C. C. Evangelista, C. Ferraz, S. Ferriera, W. Fleischmann, C. Fosler, A. E. Gabrielian, N. S. Garg, W. M. Gelbart, K. Glasser, A. Glodek, F. Gong, J. H. Gorrell, Z. Gu, P. Guan, M. Harris, N. L. Harris, D. Harvey, T. J. Heiman, J. R. Hernandez, J. Houck, D. Hostin, K. A. Houston, T. J. Howland, M.-H. Wei, C. Ibegwam, M. Jalali, F. Kalush, G. H. Karpen, Z. Ke, J. A. Kennison, K. A. Ketchum, B. E. Kimmel, C. D. Kodira, C. Kraft, S. Kravitz, D. Kulp, Z. Lai, P. Lasko, Y. Lei, A. A. Levitsky, J. Li, Z. Li, Y. Liang, X. Lin, X. Liu, B. Mattei, T. C. McIntosh, M. P. McLeod, D. McPherson, G. Merkulov, N. V. Milshina, C. Mobarry, J. Morris, A. Moshrefi, S. M. Mount, M. Moy, B. Murphy, L. Murphy, D. M. Muzny, D. L. Nelson, D. R. Nelson, K. A. Nelson, K. Nixon, D. R. Nusskern, J. M. Pacleb, M. Palazzolo, G. S. Pittman, S. Pan, J. Pollard, V. Puri, M. G. Reese, K. Reinert, K. Remington, R. D. C. Saunders, F. Scheeler, H. Shen, B. C. Shue, I. Sidén-Kiamos, M. Simpson, M. P. Skupski, T. Smith, E. Spier, A. C. Spradling, M. Stapleton, R. Strong, E. Sun, R. Svirskas, C. Tector, R. Turner, E. Venter, A. H. Wang, X. Wang, Z.-Y. Wang, D. A. Wassarman,

G. M. Weinstock, J. Weissenbach, S. M. Williams, T. Woodage, K. C. Worley, D. Wu, S. Yang, Q. A. Yao, J. Ye, R.-F. Yeh, J. S. Zaveri, M. Zhan, G. Zhang, Q. Zhao, L. Zheng, X. H. Zheng, F. N. Zhong, W. Zhong, X. Zhou, S. Zhu, X. Zhu, H. O. Smith, R. A. Gibbs, E. W. Myers, G. M. Rubin, and J. C. Venter, "The Genome Sequence of Drosophila melanogaster," *Science*, vol. 287, pp. 2185–2195, Mar. 2000. 1

[6] "Omics - `https://omics.org/`," June 2018. 2

[7] S. P. Yadav, "The wholeness in suffix -omics, -omes, and the word om.," *Journal of Biomolecular Techniques : JBT*, vol. 18, p. 277, Dec. 2007. 2

[8] R. Chen, G. I. Mias, J. Li-Pook-Than, L. Jiang, H. Y. K. Lam, R. Chen, E. Miriami, K. J. Karczewski, M. Hariharan, F. E. Dewey, Y. Cheng, M. J. Clark, H. Im, L. Habegger, S. Balasubramanian, M. O'Huallachain, J. T. Dudley, S. Hillenmeyer, R. Haraksingh, D. Sharon, G. Euskirchen, P. Lacroute, K. Bettinger, A. P. Boyle, M. Kasowski, F. Grubert, S. Seki, M. Garcia, M. Whirl-Carrillo, M. Gallardo, M. A. Blasco, P. L. Greenberg, P. Snyder, T. E. Klein, R. B. Altman, A. J. Butte, E. A. Ashley, M. Gerstein, K. C. Nadeau, H. Tang, and M. Snyder, "Personal omics profiling reveals dynamic molecular and medical phenotypes," *Cell*, vol. 148, pp. 1293–1307, Mar. 2012. 3

[9] N. Savage, "Proteomics: High-protein research," *Nature*, vol. 527, pp. S6–7, Nov. 2015. 3

[10] C. Ramus, A. Hovasse, M. Marcellin, A.-M. Hesse, E. Mouton-Barbosa, D. Bouyssié, S. Vaca, C. Carapito, K. Chaoui, C. Bruley, J. Garin, S. Cianférani, M. Ferro, A. Van Dorssaeler, O. Burlet-Schiltz, C. Schaeffer, Y. Couté, and A. Gonzalez de Peredo, "Benchmarking quantitative label-free LC–MS data processing workflows using a complex spiked proteomic standard dataset," *Journal of Proteomics*, vol. 132, pp. 51–62, Jan. 2016. 4, 19, 26, 50, 51, 55, 123

[11] M. Sturm, A. Bertsch, C. Gröpl, A. Hildebrandt, R. Hussong, E. Lange, N. Pfeifer, O. Schulz-Trieglaff, A. Zerck, K. Reinert, and O. Kohlbacher, "OpenMS - an open-source software framework for mass spectrometry," *BMC Bioinformatics*, vol. 9, no. 1, p. 163, 2008. 4, 28, 30, 33, 45, 53, 103

[12] O. Kohlbacher, K. Reinert, C. Gröpl, E. Lange, N. Pfeifer, O. Schulz-Trieglaff, and M. Sturm, "TOPP–the OpenMS proteomics pipeline," *Bioinformatics (Oxford, England)*, vol. 23, no. 2, pp. e191–7, 2007. 4, 28, 30, 32, 33, 45, 103

[13] J. Junker, C. Bielow, A. Bertsch, M. Sturm, K. Reinert, and O. Kohlbacher, "TOPPAS: a graphical workflow editor for the analysis of high-throughput proteomics data," *Journal of Proteome Research*, vol. 11, pp. 3914–3920, July 2012. 4, 30, 33, 34, 35, 38, 39, 40

[14] A. I. Nesvizhskii, A. Keller, E. Kolker, and R. Aebersold, "A Statistical Model for Identifying Proteins by Tandem Mass Spectrometry," *Analytical Chemistry*, vol. 75, no. 17, pp. 4646–4658, 2003. 4, 23, 33, 103

[15] D. Kessner, M. Chambers, R. Burke, D. Agus, and P. Mallick, "ProteoWizard: open source software for rapid proteomics tools development," *Bioinformatics (Oxford, England)*, vol. 24, no. 21, pp. 2534–2536, 2008. 4, 17, 37, 103

[16] S. Aiche, T. Sachsenberg, E. Kenar, M. Walzer, B. Wiswedel, T. Kristl, M. Boyles, A. Duschl, C. Huber, M. R. Berthold, K. Reinert, and O. Kohlbacher, "Workflows for automated downstream data analysis and visualization in large-scale computational mass spectrometry," *Proteomics*, vol. 15, pp. 1443–1447, Apr. 2015. 5, 30, 41, 42

[17] M. R. Berthold, N. Cebron, F. Dill, T. R. Gabriel, T. Kötter, T. Meinl, P. Ohl, C. Sieb, K. Thiel, and B. Wiswedel, "KNIME: The Konstanz Information Miner," pp. 319–326, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. 5, 41

[18] P. Kacsuk, Z. Farkas, M. Kozlovszky, G. Hermann, A. Balasko, K. Karoczkai, and I. Marton, "WS-PGRADE/gUSE generic DCI gateway framework for a large variety of user communities," *J Grid Comput*, vol. 10, 2012. 5, 32, 42

[19] H. Weisser, S. Nahnsen, J. Grossmann, L. Nilse, A. Quandt, H. Brauer, M. Sturm, E. Kenar, O. Kohlbacher, R. Aebersold, and L. Malmström, "An Automated Pipeline for High-Throughput Label-Free Quantitative Proteomics.," *Journal of Proteome Research*, Feb. 2013. 5, 47, 49, 55, 61, 62, 67, 78, 81, 123

[20] E. Kenar, H. Franken, S. Forcisi, K. Wörmann, H.-U. Häring, R. Lehmann, P. Schmitt-Kopplin, A. Zell, and O. Kohlbacher, "Automated label-free quantification of metabolites from liquid chromatography-mass spectrometry data," *Molecular & Cellular proteomics : MCP*, vol. 13, pp. 348–359, Jan. 2014. 5, 41, 60

[21] C. Bielow, *Quantification and Simulation of Liquid Chromatography-Mass Spectrometry Data*. PhD thesis, Freie Universität Berlin, Germany, Oct. 2012. 8

[22] "Systematic and quantitative comparison of digest efficiency and specificity reveals the impact of trypsin quality on MS-based proteomics.," vol. 75, pp. 1454–1462, Feb. 2012. 8

[23] M. Mann and N. L. Kelleher, "Precision proteomics: the case for high resolution and high mass accuracy.," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 105, pp. 18132–18138, Nov. 2008. 8

[24] M. Lynch and E. Weiner, "HPLC: high performance liquid chromatography," *Environmental Science and Technology*, vol. 13, no. 6, pp. 666–671, 1979. 9

[25] *Introduction to Modern Liquid Chromatography*. John Wiley & Sons, Sept. 2011. 9

[26] R. G. Cooks and A. L. Rockwood, "The 'Thomson'. A suggested unit for mass spectroscopists," *Rapid Communications in Mass Spectrometry*, vol. 5, p. 93, Jan. 1991. 9

[27] A. D. McNaught and A. Wilkinson, *IUPAC. Compendium of Chemical Terminology, 2nd ed. (the "Gold Book")*. Blackwell Scientific Publications, May 1997. 9

[28] 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, pp. 64–71, Oct. 1989. 10

[29] T. Koichi, W. Hiroaki, I. Yutaka, A. Satoshi, Y. Yoshikazu, Y. Tamio, and M. T, "Protein and polymer analyses up to m/z 100 000 by laser ionization time-of-flight mass spectrometry," *Rapid Communications in Mass Spectrometry*, vol. 2, no. 8, pp. 151–153. 10

[30] S. Bugovsky, W. Winkler, W. Balika, and G. Allmaier, "Long time storage (archiving) of peptide, protein and tryptic digest samples on disposable nano-coated polymer targets for MALDI MS," *EuPA Open Proteomics*, vol. 8, pp. 48–54, Sept. 2015. 11

[31] "Principles of Electrospray Ionization," vol. 10, p. M111.009407, July 2011. 11

[32] M. Kinter and N. E. Sherman, *Protein Sequencing and Identification Using Tandem Mass Spectrometry*. John Wiley & Sons, Apr. 2005. 11

[33] P. H. Dawson, *Quadrupole Mass Spectrometry and Its Applications*. Elsevier, Oct. 2013. 12

[34] U. Boesl, "Time-of-flight mass spectrometry: Introduction to the basics.," *Mass spectrometry reviews*, vol. 36, pp. 86–109, Jan. 2017. 12

[35] M. Scigelova, M. Hornshaw, A. Giannakopulos, and A. Makarov, "Fourier Transform Mass Spectrometry," *Molecular & Cellular proteomics : MCP*, vol. 10, p. M111.009431, July 2011. 12

[36] R. H. Perry, R. G. Cooks, and R. J. Noll, "Orbitrap mass spectrometry: instrumentation, ion motion and applications.," *Mass spectrometry reviews*, vol. 27, pp. 661–699, Nov. 2008. 12

[37] Thermo Fisher Scientific (Bremen), "Orbitrap Mass Analyzer and Injector - https://commons.wikimedia.org/wiki/File:OrbitrapMA%26Injector.png," 2012. 14

[38] J. H. Gross, *Mass spectrometry: a textbook*. Springer Science & Business Media, 2006. 15, 17

[39] H. Steen and M. Mann, "The ABC's (and XYZ's) of peptide sequencing.," *Nature Reviews Molecular Cell Biology*, vol. 5, pp. 699–711, Sept. 2004. 15, 20, 21

[40] I. Eidhammer, H. Barsnes, G. E. Eide, and L. Martens, *Computational and Statistical Methods for Protein Quantification by Mass Spectrometry*. John Wiley & Sons, Feb. 2013. 17

[41] 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, pp. 229–233, Apr. 1995. 18

[42] A. Frank and P. Pevzner, "PepNovo: de novo peptide sequencing via probabilistic network modeling.," *Analytical Chemistry*, vol. 77, pp. 964–973, Feb. 2005. 20

[43] B. Fischer, V. Roth, F. Roos, J. Grossmann, S. Baginsky, P. Widmayer, W. Gruissem, and J. M. Buhmann, "NovoHMM: a hidden Markov model for de novo peptide sequencing.," *Analytical Chemistry*, vol. 77, pp. 7265–7273, Nov. 2005. 20

[44] S. Andreotti, G. W. Klau, and K. Reinert, "Antilope—A Lagrangian Relaxation Approach to the de novo Peptide Sequencing Problem," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 9, no. 2, pp. 385–394, 2012. 20

[45] B. Ma and R. Johnson, "De novo sequencing and homology searching.," *Molecular & Cellular proteomics : MCP*, vol. 11, p. O111.014902, Feb. 2012. 20

[46] D. N. Perkins, D. J. Pappin, D. Creasy, and J. S. Cottrell, "Probability-based protein identification by searching sequence databases using mass spectrometry data," *Electrophoresis*, vol. 20, no. 18, pp. 3551–3567, 1999. 22, 33, 46

[47] J. K. Eng, A. L. McCormack, and J. R. Yates, "An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database," *Journal of the American Society for Mass Spectrometry*, vol. 5, pp. 976–989, Nov. 1994. 22, 46

[48] L. Y. Geer, S. P. Markey, J. A. Kowalak, L. Wagner, M. Xu, D. M. Maynard, X. Yang, W. Shi, and S. H. Bryant, "Open mass spectrometry search algorithm," *Journal of Proteome Research*, vol. 3, no. 5, pp. 958–964. 22, 32, 33, 37, 81

[49] R. Craig and R. C. Beavis, "TANDEM: matching proteins with tandem mass spectra," *Bioinformatics (Oxford, England)*, vol. 20, no. 9, pp. 1466–1467, 2004. 22, 32, 33

[50] S. Kim and P. A. Pevzner, "MS-GF+ makes progress towards a universal database search tool for proteomics.," *Nature communications*, vol. 5, p. 5277, Oct. 2014. 22, 81

[51] J. K. Eng, T. A. Jahan, and M. R. Hoopmann, "Comet: An open-source MS/MS sequence database search tool," *Proteomics*, vol. 13, pp. 22–24, Dec. 2012. 22, 81

[52] J. Cox, N. Neuhauser, A. Michalski, R. A. Scheltema, J. V. Olsen, and M. Mann, "Andromeda: a peptide search engine integrated into the MaxQuant environment.," *Journal of Proteome Research*, vol. 10, pp. 1794–1805, Apr. 2011. 22

[53] L. Käll, J. D. Canterbury, J. Weston, W. S. Noble, and M. J. MacCoss, "Semi-supervised learning for peptide identification from shotgun proteomics datasets," *Nature Methods*, vol. 4, pp. 923–925, Oct. 2007. 22, 46

[54] M. The, M. J. MacCoss, W. S. Noble, and L. Käll, "Fast and Accurate Protein False Discovery Rates on Large-Scale Proteomics Data Sets with Percolator 3.0.," *Journal of the American Society for Mass Spectrometry*, vol. 27, pp. 1719–1727, Nov. 2016. 22

[55] S. Nahnsen, A. Bertsch, J. Rahnenführer, A. Nordheim, and O. Kohlbacher, "Probabilistic Consensus Scoring Improves Tandem Mass Spectrometry Peptide Identification," *Journal of Proteome Research*, vol. 10, pp. 3332–3343, Aug. 2011. 23

[56] D. Shteynberg, E. W. Deutsch, H. Lam, J. K. Eng, Z. Sun, N. Tasman, L. Mendoza, R. L. Moritz, R. Aebersold, and A. I. Nesvizhskii, "iProphet: multi-level integrative analysis of shotgun proteomic data improves peptide and protein identification rates and error estimates.," *Molecular & Cellular proteomics : MCP*, vol. 10, p. M111.007690, Dec. 2011. 23, 81

[57] E. W. Deutsch, L. Mendoza, D. Shteynberg, T. Farrah, H. Lam, N. Tasman, Z. Sun, E. Nilsson, B. Pratt, B. Prazen, J. K. Eng, D. B. Martin, A. I. Nesvizhskii, and R. Aebersold, "A guided tour of the Trans-Proteomic Pipeline," *Proteomics*, vol. 10, no. 6, pp. 1150–1159, 2010. 23, 103

[58] E. W. Deutsch, L. Mendoza, D. Shteynberg, J. Slagel, Z. Sun, and R. L. Moritz, "Trans-Proteomic Pipeline, a standardized data processing pipeline for large-scale reproducible proteomics informatics," *Proteomics. Clinical Applications*, vol. 9, pp. 745–754, Apr. 2015. 23

[59] K. Ma, O. Vitek, and A. I. Nesvizhskii, "A statistical model-building perspective to identification of MS/MS spectra with PeptideProphet.," *BMC Bioinformatics*, vol. 13 Suppl 16, p. S1, 2012. 23, 81

[60] L. Reiter, M. Claassen, S. P. Schrimpf, M. Jovanovic, A. Schmidt, J. M. Buhmann, M. O. Hengartner, and R. Aebersold, "Protein identification false discovery rates for very large proteomics data sets generated by tandem mass spectrometry.," *Molecular & Cellular proteomics : MCP*, vol. 8, pp. 2405–2417, Nov. 2009. 23

[61] M. Bantscheff, M. Schirle, G. Sweetman, J. Rick, and B. Kuster, "Quantitative mass spectrometry in proteomics: a critical review.," *Analytical and Bioanalytical Chemistry*, vol. 389, pp. 1017–1031, Oct. 2007. 24, 26

[62] S. E. Ong, B. Blagoev, I. Kratchmarova, D. B. Kristensen, H. Steen, A. Pandey, and M. Mann, "Stable isotope labeling by amino acids in cell culture, SILAC, as a simple and accurate approach to expression proteomics," *Molecular & Cellular proteomics : MCP*, 2002. 25, 46

[63] S. E. Ong and M. Mann, "Stable Isotope Labeling by Amino Acids in Cell Culture for Quantitative Proteomics," in *Quantitative Proteomics by Mass Spectrometry*, pp. 37–52, Totowa, NJ: Humana Press, 2007. 25

[64] J. W. Gouw, B. B. J. Tops, and J. Krijgsveld, "Metabolic labeling of model organisms using heavy nitrogen (15N).," *Methods in Molecular Biology*, vol. 753, pp. 29–42, 2011. 25

[65] S. P. Gygi, B. Rist, T. J. Griffin, J. K. Eng, and R. Aebersold, "Proteome analysis of low-abundance proteins using multidimensional chromatography and isotope-coded affinity tags.," *Journal of Proteome Research*, vol. 1, pp. 47–54, Jan. 2002. 25

[66] P. J. Boersema, R. Raijmakers, S. Lemeer, S. Mohammed, and A. J. R. Heck, "Multiplex peptide stable isotope dimethyl labeling for quantitative proteomics.," *Nature Protocols*, vol. 4, no. 4, pp. 484–494, 2009. 25

[67] P. L. Ross, "Multiplexed Protein Quantitation in Saccharomyces cerevisiae Using Amine-reactive Isobaric Tagging Reagents," *Molecular & Cellular proteomics : MCP*, vol. 3, pp. 1154–1169, Sept. 2004. 26, 46

[68] A. Thompson, J. Schäfer, K. Kuhn, S. Kienle, J. Schwarz, G. Schmidt, T. Neumann, and C. Hamon, "Tandem Mass Tags: A Novel Quantification Strategy for Comparative Analysis of Complex Protein Mixtures by MS/MS," *Analytical Chemistry*, no. 75, pp. 1895–1904, 2003. 26, 46

[69] D. H. Lundgren, S.-I. Hwang, L. Wu, and D. K. Han, "Role of spectral counting in quantitative proteomics," *Expert Review of Proteomics*, vol. 7, pp. 39–53, Jan. 2014. 26

[70] H. L. Röst, T. Sachsenberg, S. Aiche, C. Bielow, H. Weisser, F. Aicheler, S. Andreotti, H.-C. Ehrlich, P. Gutenbrunner, E. Kenar, X. Liang, S. Nahnsen, L. Nilse, J. Pfeuffer, G. Rosenberger, M. Rurik, U. Schmitt, J. Veit, M. Walzer, D. Wojnar, W. E. Wolski, O. Schilling, J. S. Choudhary, L. Malmström, R. Aebersold, K. Reinert, and O. Kohlbacher, "OpenMS: a flexible open-source software platform for mass spectrometry data analysis.," *Nature Methods*, vol. 13, pp. 741–748, Aug. 2016. 28, 30

[71] J. Häkkinen, G. Vincic, O. Månsson, K. Wårell, and F. Levander, "The proteios software environment: an extensible multiuser platform for management and analysis of proteomics data.," *Journal of Proteome Research*, vol. 8, pp. 3037–3043, June 2009. 28

[72] L. N. Mueller, O. Rinner, A. Schmidt, S. Letarte, B. Bodenmiller, M.-Y. Brusniak, O. Vitek, R. Aebersold, and M. Müller, "SuperHirn - a novel tool for high resolution LC-MS-based peptide/protein profiling," *Proteomics*, vol. 7, pp. 3470–3480, Oct. 2007. 28, 45

[73] D. May, W. Law, M. Fitzgibbon, Q. Fang, and M. McIntosh, "Software platform for rapidly creating computational tools for mass spectrometry-based proteomics.," *Journal of Proteome Research*, vol. 8, pp. 3212–3217, June 2009. 28

[74] J. Cox and M. Mann, "MaxQuant enables high peptide identification rates, individualized p.p.b.-range mass accuracies and proteome-wide protein quantification," *Nature Biotechnology*, vol. 26, pp. 1367–1372, Nov. 2008. 28, 32, 45, 78, 81

[75] J. Cox, M. Y. Hein, C. A. Luber, I. Paron, N. Nagaraj, and M. Mann, "Accurate proteome-wide label-free quantification by delayed normalization and maximal peptide ratio extraction, termed MaxLFQ," *Molecular & Cellular proteomics : MCP*, vol. 13, pp. 2513–2526, Sept. 2014. 28, 32, 45, 78, 81

[76] H. L. Röst, U. Schmitt, R. Aebersold, and L. Malmström, "pyOpenMS: a Python-based interface to the OpenMS mass-spectrometry algorithm library.," *Proteomics*, vol. 14, pp. 74–77, Jan. 2014. 30

[77] L. de la Garza, J. Veit, A. Szolek, M. Röttig, S. Aiche, S. Gesing, K. Reinert, and O. Kohlbacher, "From the desktop to the grid: scalable bioinformatics via workflow conversion," *BMC Bioinformatics*, vol. 17, no. 1, pp. 1–12, 2016. 30, 44

[78] J. Veit, T. Sachsenberg, A. Chernev, F. Aicheler, H. Urlaub, and O. Kohlbacher, "LFQProfiler and RNP xl: Open-Source Tools for Label-Free Quantification and Protein–RNA Cross-Linking Integrated into Proteome Discoverer," *Journal of Proteome Research*, vol. 15, pp. 3441–3448, Aug. 2016. 30, 48, 49, 51, 52, 54, 62

[79] B. Maclean, D. M. Tomazela, N. Shulman, M. Chambers, G. L. Finney, B. Frewen, R. Kern, D. L. Tabb, D. C. Liebler, and M. J. MacCoss, "Skyline: an open source document editor for creating and analyzing targeted proteomics experiments," *Bioinformatics (Oxford, England)*, vol. 26, pp. 966–968, Mar. 2010. 32

[80] A. Keller, J. K. Eng, N. Zhang, X.-j. Li, and R. Aebersold, "A uniform proteomics MS/MS analysis platform utilizing open XML file formats," *Molecular Systems Biology*, vol. 1, p. 2005.0017, 2005. 32

[81] L. Martens, M. Chambers, M. Sturm, D. Kessner, F. Levander, J. Shofstahl, W. H. Tang, A. Römpp, S. Neumann, A. D. Pizarro, L. Montecchi-Palazzi, N. Tasman, M. Coleman, F. Reisinger, P. Souda, H. Hermjakob, P.-A. Binz, and E. W. Deutsch, "mzML–a community standard for mass spectrometry data," *Molecular & Cellular proteomics : MCP*, vol. 10, no. 1, p. R110.000133, 2011. 32

[82] M. Eisenacher, "mzIdentML: an open community-built standard format for the results of proteomics spectrum identification algorithms," *Methods in Molecular Biology*, vol. 696, pp. 161–177, 2011. 32

[83] E. W. Deutsch, M. Chambers, S. Neumann, F. Levander, P.-A. Binz, J. Shofstahl, D. S. Campbell, L. Mendoza, D. Ovelleiro, K. Helsens, L. Martens, R. Aebersold, R. L. Moritz, and M.-Y. Brusniak, "TraML–A Standard Format for Exchange of Selected Reaction Monitoring Transition Lists," *Molecular & Cellular proteomics : MCP*, vol. 11, no. 4, p. R111.015040, 2012. 32

[84] D. Blankenberg, G. Von Kuster, N. Coraor, G. Ananda, R. Lazarus, M. Mangan, A. Nekrutenko, and J. Taylor, "Galaxy: a web-based genome analysis tool for experimentalists," *Current Protocols in Molecular Biology*, vol. Chapter 19, pp. Unit 19.10.1–21, Jan. 2010. 32

[85] J. Goecks, A. Nekrutenko, and J. Taylor, "Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences," *Genome Biology*, vol. 11, no. 8, p. R86, 2010.

[86] B. Giardine, C. Riemer, R. C. Hardison, R. Burhans, L. Elnitski, P. Shah, Y. Zhang, D. Blankenberg, I. Albert, J. Taylor, W. Miller, W. J. Kent, and A. Nekrutenko, "Galaxy: a platform for interactive large-scale genome analysis," *Genome Reseach*, vol. 15, no. 10, pp. 1451–1455, 2005. 32

[87] T. Oinn, M. Greenwood, M. Addis, N. Alpdemir, J. Ferris, K. Glover, C. Goble, A. Goderis, D. Hull, D. Marvin, P. Li, P. Lord, M. R. Pocock, M. Senger, R. Stevens, A. Wipat, and C. Wroe, "Taverna: lessons in creating a workflow environment for the life sciences," *Concurrency and Computation: Practice and Experience*, vol. 18, no. 10, pp. 1067–1100, 2006. 32

[88] D. Hull, K. Wolstencroft, R. Stevens, C. Goble, M. R. Pocock, P. Li, and T. Oinn, "Taverna: a tool for building and running workflows of services," *Nucleic Acids Research*, vol. 34, no. Web Server issue, pp. W729–32, 2006. 32

[89] B. Linke, R. Giegerich, and A. Goesmann, "Conveyor: a workflow engine for bioinformatic analyses," *Bioinformatics (Oxford, England)*, vol. 27, no. 7, pp. 903–911, 2011. 32

[90] B. Néron, H. Ménager, C. Maufrais, N. Joly, J. Maupetit, S. Letort, S. Carrere, P. Tuffery, and C. Letondal, "Mobyle: a new full web bioinformatics framework," *Bioinformatics (Oxford, England)*, vol. 25, no. 22, pp. 3005–3011, 2009. 32

[91] E. Deelman, G. Singh, M.-H. Su, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, K. Vahi, G. B. Berriman, J. Good, A. Laity, J. C. Jacob, and D. S. Katz, "Pegasus: A framework for mapping complex scientific workflows onto distributed systems," *Scientific Programming*, vol. 13, no. 3, pp. 219–237, 2005. 32

[92] I. Altintas, C. Berkley, E. Jaeger, M. Jones, B. Ludascher, and S. Mock, "Kepler: an extensible system for design and execution of scientific workflows," in *Proceedings. 16th International Conference on Scientific and Statistical Database Management, 2004.*, pp. 423–424, IEEE, 2004. 32

[93] J. S. de Bruin, A. M. Deelder, and M. Palmblad, "Scientific Workflow Management in Proteomics," *Molecular & Cellular proteomics : MCP*, vol. in Press, 2012. 32

[94] M. Specht, S. Kuhlgert, C. Fufezan, and M. Hippler, "Proteomics to go: Proteomatic enables the user-friendly creation of versatile MS/MS data evaluation workflows," *Bioinformatics (Oxford, England)*, vol. 27, pp. 1183–1184, Apr. 2011. 32

[95] R Development Core Team, "R: A Language and Environment for Statistical Computing," *Vienna Austria R Foundation for Statistical Computing*, vol. 1, no. 09/18/2009, pp. ISBN 3—-900051–07–0, 2008. 36, 40

[96] B. Hoekman, R. Breitling, F. Suits, R. Bischoff, and P. Horvatovich, "msCompare: a framework for quantitative analysis of label-free LC-MS data for comparative biomarker studies," *Molecular & Cellular proteomics : MCP*, 2012. 37

[97] J. Griss, A. R. Jones, T. Sachsenberg, M. Walzer, L. Gatto, J. Hartler, G. G. Thallinger, R. M. Salek, C. Steinbeck, N. Neuhauser, J. Cox, S. Neumann, J. Fan, F. Reisinger, Q.-W. Xu, N. Del Toro, Y. Pérez-Riverol, F. Ghali, N. Bandeira, I. Xenarios, O. Kohlbacher, J. A. Vizcaíno, and H. Hermjakob, "The mzTab data exchange format: communicating mass-spectrometry-based proteomics and metabolomics experimental results to a wider audience," *Molecular & Cellular proteomics : MCP*, vol. 13, pp. 2765–2775, Oct. 2014. 40

[98] D. Bouyssié, A. Gonzalez de Peredo, E. Mouton-Barbosa, R. Albigot, L. Roussel, N. Ortega, C. Cayrol, O. Burlet-Schiltz, J.-P. Girard, and B. Monsarrat, "Mascot file parsing and quantification (MFPaQ), a new software to parse, validate, and quantify proteomics data generated by ICAT and SILAC mass spectrometric analyses: application to the proteomics study of membrane proteins from primary human endothelial cells," *Molecular & Cellular proteomics : MCP*, vol. 6, pp. 1621–1637, Sept. 2007. 45

[99] K. Kramer, T. Sachsenberg, B. M. Beckmann, S. Qamar, K.-L. Boon, M. W. Hentze, O. Kohlbacher, and H. Urlaub, "Photo-cross-linking and high-resolution mass spectrometry for assignment of RNA-binding sites in RNA-binding proteins," *Nature Methods*, vol. 11, pp. 1064–1070, Oct. 2014. 45, 51, 53, 56, 123

[100] V. Dorfer, P. Pichler, T. Stranzl, J. Stadlmann, T. Taus, S. Winkler, and K. Mechtler, "MS Amanda, a Universal Identification Algorithm Optimized for High Accuracy Tandem Mass Spectra," *Journal of Proteome Research*, vol. 13, no. 8, pp. 3679–3684, 2014. 46

[101] T. Taus, T. Köcher, P. Pichler, C. Paschke, A. Schmidt, C. Henrich, and K. Mechtler, "Universal and Confident Phosphorylation Site Localization Using phosphoRS," *Journal of Proteome Research*, vol. 10, no. 12, pp. 5354–5362, 2011. 46

[102] O. Serang and W. S. Noble, "Faster mass spectrometry-based protein inference: junction trees are more efficient than sampling and marginalization by enumeration," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 9, pp. 809–817, May 2012. 47

[103] R. Smith, D. Ventura, and J. T. Prince, "LC-MS alignment in theory and practice: a comprehensive algorithmic review," *Briefings in Bioinformatics*, vol. 16, pp. 104–117, Jan. 2015. 60

[104] E. Lange, C. Gröpl, O. Schulz-Trieglaff, A. Leinenbach, C. Huber, and K. Reinert, "A geometric approach for the alignment of liquid chromatography-mass spectrometry data," *Bioinformatics (Oxford, England)*, vol. 23, pp. i273–81, July 2007. 61

[105] F. Beck, J. Geiger, S. Gambaryan, J. Veit, M. Vaudel, P. Nollau, O. Kohlbacher, L. Martens, U. Walter, A. Sickmann, and R. P. Zahedi, "Time-resolved characterization of cAMP/PKA-dependent signaling reveals that platelet inhibition is a concerted process involving multiple signaling pathways," *Blood*, vol. 123, pp. e1–e10, Jan. 2014. 65

[106] L. Nilse, F. C. Sigloch, M. L. Biniossek, and O. Schilling, "Toward improved peptide feature detection in quantitative proteomics using stable isotope labeling.," *Proteomics. Clinical Applications*, vol. 9, pp. 706–714, Aug. 2015. 78, 81

[107] M. Choi, Z. F. Eren-Dogu, C. Colangelo, J. Cottrell, M. R. Hoopmann, E. A. Kapp, S. Kim, H. Lam, T. A. Neubert, M. Palmblad, B. S. Phinney, S. T. Weintraub, B. Maclean, and O. Vitek, "ABRF Proteome Informatics Research Group (iPRG) 2015 Study: Detection of Differentially Abundant Proteins in Label-Free Quantitative LC-MS/MS Experiments.," *Journal of Proteome Research*, vol. 16, pp. 945–957, Feb. 2017. 78, 81, 82, 92, 123, 142

[108] S. Aiche, *Inferring Proteolytic Processes from Mass Spectrometry Time Series Data*. PhD thesis, Freie Universität Berlin, Germany, Sept. 2013. 78

[109] F. Mosteller and R. A. Fisher, "Questions and Answers," *The American Statistician*, vol. 2, pp. 30–31, Oct. 1948. 82

[110] M. Jeanmougin, A. de Reynies, L. Marisa, C. Paccard, G. Nuel, and M. Guedj, "Should We Abandon the t-Test in the Analysis of Gene Expression Microarray Data: A Comparison of Variance Modeling Strategies," *PLoS ONE*, vol. 5, p. e12336, Sept. 2010. 82

[111] Y. Benjamini and Y. Hochberg, "Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 57, no. 1, pp. 289–300, 1995. 84

[112] S. Dammeier, S. Nahnsen, J. Veit, F. Wehner, M. Ueffing, and O. Kohlbacher, "Mass-Spectrometry-Based Proteomics Reveals Organ-Specific Expression Patterns To Be Used as Forensic Evidence," *Journal of Proteome Research*, vol. 15, pp. 182–192, Dec. 2015. 98, 101, 104, 108, 109, 110, 111, 112, 115

[113] B. Karger, E. Meyer, P. J. Knudsen, and B. Brinkmann, "DNA typing of cellular material on perforating bullets," *Int. J. Legal Med*, vol. 108, no. 4, pp. 177–179, 1996. 97

[114] B. Karger, E. Meyer, and A. DuChesne, "STR analysis on perforating FMJ bullets and a new VWA variant allele," *Int. J. Legal Med*, vol. 110, no. 2, pp. 101–103, 1997. 97

[115] H. H. Maurer, "What is the future of (ultra) high performance liquid chromatography coupled to low and high resolution mass spectrometry for toxicological drug screening?," *Journal of chromatography. A*, vol. 1292, pp. 19–24, May 2013. 99

[116] J. Segura, J. A. Pascual, and R. Gutiérrez-Gallego, "Procedures for monitoring recombinant erythropoietin and analogues in doping control," *Analytical and Bioanalytical Chemistry*, vol. 388, pp. 1521–1529, Aug. 2007. 99

[117] O. Mezger, W. Heess, F. Hasslacher, and R. Heindl, "Determination of the Type of Pistol Employed from an Examination of Fired Bullets and Shells," *The American Journal of Police Science*, vol. 2, no. 6, pp. 473–499, 1931. 99

[118] P. Dieltjes, R. Mieremet, S. Zuniga, T. Kraaijenbrink, J. Pijpe, and P. de Knijff, "A sensitive method to extract DNA from biological traces present on ammunition for the purpose of genetic profiling," *Int. J. Legal Med*, vol. 125, pp. 597–602, July 2011. 99

[119] Uniform Crime Reports 2013 Online; Federal Bureau of Investigation of the US Department of Justice, "Percent of Offenses Cleared by Arrest or Exceptional Means." 99, 119

[120] M. Bauer, S. Polzin, and D. Patzelt, "Quantification of RNA degradation by semi-quantitative duplex and competitive RT-PCR: a possible indicator of the age of bloodstains?," *Forensic Sci. Int*, vol. 138, pp. 94–103, Dec. 2003. 99

[121] C. Hampson, J. Louhelainen, and S. McColl, "An RNA expression method for aging forensic hair samples," *Journal of forensic sciences*, vol. 56, pp. 359–365, Mar. 2011. 99

[122] K. Takahama, "Forensic application of organ-specific antigens," *Forensic Sci. Int*, vol. 80, pp. 63–69, June 1996. 99, 116

[123] F. Wehner, N. R. M. Moos, H.-D. Wehner, D. Martin, and M. M. Schulz, "Immunocytochemical examinations of biological traces on expanding bullets (QD-PEP)," *Forensic Sci. Int*, vol. 182, pp. 66–70, Nov. 2008. 99, 102, 111

[124] M. Beck, M. Claassen, and R. Aebersold, "Comprehensive proteomics," *Current opinion in biotechnology*, vol. 22, pp. 3–8, Feb. 2011. 99

[125] E. L. Huttlin, M. P. Jedrychowski, J. E. Elias, T. Goswami, R. Rad, S. A. Beausoleil, J. Villén, W. Haas, M. E. Sowa, and S. P. Gygi, "A tissue-specific atlas of mouse protein phosphorylation and expression," *Cell*, vol. 143, pp. 1174–1189, Dec. 2010. 100, 116

[126] P. R. Cutillas and B. Vanhaesebroeck, "Quantitative profile of five murine core proteomes using label-free functional proteomics," *Molecular & Cellular proteomics : MCP*, vol. 6, pp. 1560–1573, Sept. 2007. 100

[127] A. Keller, A. I. Nesvizhskii, E. Kolker, and R. Aebersold, "Empirical statistical model to estimate the accuracy of peptide identifications made by MS/MS and database search," *Analytical Chemistry*, vol. 74, pp. 5383–5392, Oct. 2002. 103

[128] M. Chambers, B. Maclean, R. Burke, D. Amodei, D. L. Ruderman, S. Neumann, L. Gatto, B. Fischer, B. Pratt, J. Egertson, K. Hoff, D. Kessner, N. Tasman, N. Shulman, B. Frewen, T. A. Baker, M.-Y. Brusniak, C. Paulse, D. Creasy, L. Flashner, K. Kani, C. Moulding, S. L. Seymour, L. M. Nuwaysir, B. Lefebvre, F. Kuhlmann, J. Roark, P. Rainer, S. Detlev, T. Hemenway, A. Huhmer, J. Langridge, B. Connolly, T. Chadick, K. Holly, J. Eckels, E. W. Deutsch, R. L. Moritz, J. E. Katz, D. B. Agus, M. MacCoss, D. L. Tabb, and P. Mallick, "A cross-platform toolkit for mass spectrometry and proteomics," *Nature Biotechnology*, vol. 30, pp. 918–920, Oct. 2012. 103

[129] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, Feb. 2011. 103, 105

[130] F. Nelli, "Machine Learning with scikit-learn," in *Python Data Analytics*, pp. 237–264, Berkeley, CA: Apress, Berkeley, CA, 2015. 103, 105

[131] J. A. Vizcaíno, E. W. Deutsch, R. Wang, A. Csordas, F. Reisinger, D. Ríos, J. A. Dianes, Z. Sun, T. Farrah, N. Bandeira, and others, "ProteomeXchange provides globally coordinated proteomics data submission and dissemination," *Nature Biotechnology*, vol. 32, no. 3, pp. 223–226, 2014. 103

[132] J. A. Vizcaíno, R. G. Côté, A. Csordas, J. A. Dianes, A. Fabregat, J. M. Foster, J. Griss, E. Alpi, M. Birim, J. Contell, and others, "The PRoteomics IDEntifications (PRIDE) database and associated tools: status in 2013," *Nucleic Acids Research*, vol. 41, no. D1, pp. D1063–D1069, 2013. 103

[133] B. Schölkopf, A. Smola, R. Williamson, and P. Bartlett, "New support vector algorithms," *Neural computation*, vol. 12, pp. 1207–1245, May 2000. 105

[134] L. Breiman, M. Last, and J. Rice, "Random Forests: Finding Quasars." Springer-Verlag, New York, 2003. 105

[135] H. Zhang, "Exploring Conditions for the Optimality of Naïve Bayes," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 19, no. 02, pp. 183–198, 2005. 105

[136] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool," *Journal of Molecular Biology*, vol. 215, no. 3, pp. 403–410, 1990. 112

[137] F. R. W. Hunt and C. Cranz, "Lehrbuch der Ballistik. Vol. I., External Ballistics," *The Mathematical Gazette*, vol. 13, no. 184, p. 212, 1926. 111

[138] T. Geiger, A. Wehner, C. Schaab, J. Cox, and M. Mann, "Comparative proteomic analysis of eleven common cell lines reveals ubiquitous but varying expression of most proteins," *Molecular & Cellular proteomics : MCP*, vol. 11, p. M111.014050, Mar. 2012. 116

[139] K. Van Steendam, M. De Ceuleneer, M. Dhaenens, D. Van Hoofstat, and D. Deforce, "Mass spectrometry-based proteomics as a tool to identify biological matrices in forensic science," *Int. J. Legal Med*, vol. 127, pp. 287–298, Mar. 2013. 116

[140] I. Prokic, B. S. Cowling, and J. Laporte, "Amphiphysin 2 (BIN1) in physiology and diseases," *Journal of molecular medicine (Berlin, Germany)*, vol. 92, pp. 453–463, May 2014. 117

[141] B. Cummins, M. L. Auckland, and P. Cummins, "Cardiac-specific troponin-I radioimmunoassay in the diagnosis of acute myocardial infarction," *American heart journal*, vol. 113, pp. 1333–1344, June 1987. 117

[142] M. Wilhelm, J. Schlegl, H. Hahne, A. M. Gholami, M. Lieberenz, M. M. Savitski, E. Ziegler, L. Butzmann, S. Gessulat, H. Marx, T. Mathieson, S. Lemeer, K. Schnatbaum, U. Reimer, H. Wenschuh, M. Mollenhauer, J. Slotta-Huspenina, J.-H. Boese, M. Bantscheff, A. Gerstmair, F. Faerber, and B. Kuster, "Mass-spectrometry-based draft of the human proteome," *Nature*, vol. 509, pp. 582–587, May 2014. 118

[143] A. W. Bell, E. W. Deutsch, C. E. Au, R. E. Kearney, R. C. Beavis, S. Sechi, T. Nilsson, J. J. M. Bergeron, and HUPO Test Sample Working Group, "A HUPO test sample study reveals common problems in mass spectrometry-based proteomics," *Nature Methods*, vol. 6, pp. 423–430, June 2009. 119

Appendix A

OptiQuant

Workflow	Tool	Parameter	Dataset			
			HQ	LQ	LQ + spike-in	iPRG
FFC	FeatureFinderCentroided	mass_trace:mz_tolerance (Da)	0.01	0.01	0.01	0.01
		mass_trace:min_spectra	10	7	7	7
		mass_trace:max_missing	1	2	2	2
		isotopic_pattern:charge_high	5	5	5	5
		isotopic_pattern:mz_tolerance (Da)	0.01	0.01	0.01	0.01
	IDFilter	charge	1:5	1:5	1:5	1:5
		score:pep	0	0	0	0.15
	IDMapper	best:n_peptide_hits	0	0	0	1
		mz_tolerance (ppm)	5	5	5	5
		mz_reference	peptide	peptide	peptide	peptide
		use_centroid_rt	true	true	true	false
		use_centroid_mz	true	true	true	true
	MapAlignerIdentification	min_run_occur	-	-	-	9
	FeatureLinkerUnlabeledQT	distance_RT:max_difference (s)	20	20	20	30
		distance_mz:max_difference (ppm)	5	10	10	10
		distance_mz:unit	ppm	ppm	ppm	ppm
FFMPX	FeatureFinderMultiplex	mz_tolerance (ppm)	6	10	10	10
		intensity_cutoff	0.0001	0.0001	0.0001	1000
	IDFilter	charge	1:5	1:5	1:5	1:5
		score:pep	0	0	0	0.15
	IDMapper	best:n_peptide_hits	0	0	0	1
		mz_tolerance (ppm)	5	5	5	5
		mz_reference	peptide	peptide	peptide	peptide
		use_centroid_rt	true	true	true	false
		use_centroid_mz	true	true	true	true
	MapAlignerIdentification	min_run_occur	-	-	-	9
	FeatureLinkerUnlabeledQT	distance_RT:max_difference (s)	20	20	20	30
		distance_mz:max_difference (ppm)	5	10	10	10
		distance_mz:unit	ppm	ppm	ppm	ppm
OptiQuant	MassTraceExtractor	common:noise_threshold_int	0.0001	0.0001	0.0001	10
		mtd:mass_error_ppm	10	10	10	10
		mtd:trace_termination_outliers	2	3	3	5
		mtd:min_sample_rate	0.8	0.7	0.7	0.5
		mtd:min_trace_length	5	5	5	10
		epd:min_fwhm	2	2	2	3
		epd:max_fwhm	9999	9999	9999	9999
	IDFilter	charge	1:5	1:5	1:5	1:5
		ignore_charge	true	true	true	true
	IDMapper	mz_tolerance (ppm)	5	5	5	5
		mz_reference	peptide	peptide	peptide	peptide
		use_centroid_rt	true	true	true	false
		use_centroid_mz	true	true	true	true
	FeatureLinkerUnlabeledKD	warp:min_rel_cc_size	0.5	0.5	0.5	0.7
		distance_RT:weight	1	1	1	0.1
		LOWESS:span	0.3	0.3	0.3	0.3
	OptiQuant	mz_tol (ppm)	5	10	10	5
		max_nr_traces	7	7	7	6
		min_averagine_score	0.85	0.85	0.85	0.9
		trace_preference	similarity	similarity	similarity	intensity
		adaptive_iso_mass_diff	true	true	true	false
		require_n_out_of_first_m	3/4	3/4	3/4	2/2
MaxQuant		maxCharge				5
		centroidMatchTol (ppm)				8
		centroidMatchTolInPpm				true
		maxMissedCleavage				2
		enzyme				Trypsin
		fixedModifications				Carbamidomethyl (C)
		variableModifications				Oxidation (M) Acetyl (Protein N-term) Gln->pyro-Glu Glu->pyro-Glu Deamidation (NQ)
		firstSearchTol (ppm)				20
		mainSearchTol (ppm)				4.5
		searchTolInPpm				true
		lfqMode				1
		matchBetweenRuns				true
		msmsParamsArray				FTMS

Table A.1: Parameter settings of all benchmarked workflows. Parameters not listed here were set to their default.

Comparison	Protein	Actual FC	OptiQuant FC	ϵ(FC)	FFC FC	ϵ(FC)	FFMPX FC	ϵ(FC)	MaxLFQ FC	ϵ(FC)
2 vs 1	ZRT4	5.64	5.89	0.25	6.85	1.20	7.28	1.63	5.69	**0.04**
	PGA4	-4.20	-2.48	1.72	-2.64	1.55	-3.43	**0.77**	-3.15	1.04
	SFG2	-2.91	-2.53	0.38	-3.17	-0.27	-3.62	-0.72	-2.98	**-0.07**
	VAC2	-0.24	-0.22	**0.02**	0.02	0.26	-0.15	0.09	-0.32	-0.08
	ISCB	-1.87	-2.63	**-0.76**	-2.78	-0.90	-3.07	-1.19	-2.88	-1.00
	UTR6	5.02	5.08	**0.06**	5.57	0.55	6.12	1.10	5.87	0.85
3 vs 1	ZRT4	0.14	-0.25	**-0.39**	-0.28	-0.42	-0.35	-0.49	-0.40	-0.53
	PGA4	-0.14	-0.57	-0.43	-0.44	**-0.30**	-0.71	-0.57	-0.58	-0.44
	SFG2	2.12	2.55	0.44	2.62	0.50	3.29	1.18	2.49	**0.38**
	VAC2	-2.12	-2.39	**-0.28**	-2.58	-0.46	-2.85	-0.73	-2.65	-0.53
	ISCB	-4.78	-5.29	**-0.51**	-6.25	-1.47	-3.30	1.49	-5.67	-0.89
	UTR6	4.78	4.69	**-0.09**	5.02	0.24	5.69	0.91	5.45	0.67
4 vs 1	ZRT4	-4.06	-0.62	3.44	-2.71	**1.35**	-1.92	2.14	-2.47	1.59
	PGA4	5.51	6.10	0.59	5.95	0.44	7.05	1.55	5.93	**0.43**
	SFG2	1.87	2.37	0.49	2.33	**0.46**	3.04	1.17	2.34	**0.46**
	VAC2	-5.02	-3.17	1.85	-6.06	-1.03	-0.52	4.51	-4.08	**0.94**
	ISCB	0.24	-0.14	-0.39	-0.21	-0.45	-0.28	-0.52	-0.07	**-0.31**
	UTR6	2.91	2.46	-0.44	2.87	**-0.03**	3.47	0.56	2.78	-0.13
3 vs 2	ZRT4	-5.51	-6.15	-0.64	-7.13	-1.62	-7.63	-2.12	-6.08	**-0.57**
	PGA4	4.06	1.91	-2.15	2.21	-1.85	2.72	**-1.34**	2.57	-1.49
	SFG2	5.02	5.08	**0.06**	5.79	0.77	6.92	1.90	5.47	0.45
	VAC2	-1.87	-2.18	**-0.30**	-2.60	-0.72	-2.70	-0.82	-2.32	-0.45
	ISCB	-2.91	-2.66	0.24	-3.47	-0.56	-0.23	2.68	-2.80	**0.11**
	UTR6	-0.24	-0.39	**-0.14**	-0.55	-0.31	-0.43	-0.19	-0.43	-0.19
4 vs 2	ZRT4	-9.70	-6.51	3.19	-9.55	**0.15**	-9.19	0.51	-8.15	1.55
	PGA4	9.70	8.58	-1.13	8.59	-1.11	10.48	0.78	9.09	**-0.62**
	SFG2	4.78	4.89	**0.11**	5.51	0.72	6.66	1.88	5.32	0.54
	VAC2	-4.78	-2.95	1.83	-6.07	-1.29	-0.37	4.42	-3.76	**1.03**
	ISCB	2.12	2.49	**0.37**	2.57	0.45	2.79	0.67	2.81	0.70
	UTR6	-2.12	-2.62	**-0.50**	-2.69	-0.58	-2.65	-0.54	-3.10	-0.98
4 vs 3	ZRT4	-4.20	-0.36	3.83	-2.43	**1.77**	-1.57	2.63	-2.07	2.12
	PGA4	5.64	6.66	1.02	6.38	**0.74**	7.77	2.12	6.51	0.87
	SFG2	-0.24	-0.19	0.06	-0.29	-0.05	-0.25	**-0.01**	-0.16	0.08
	VAC2	-2.91	-0.78	2.13	-3.48	**-0.57**	2.33	5.24	-1.43	1.48
	ISCB	5.02	5.15	**0.13**	6.04	1.01	3.02	-2.00	5.61	0.59
	UTR6	-1.87	-2.23	-0.36	-2.15	**-0.27**	-2.22	-0.35	-2.67	-0.80
Median				**0.06**		-0.16		0.72		0.095

Table A.2: True and estimated log fold changes for all benchmarked workflows, proteins, and sample comparisons. ϵ(FC) denotes the absolute error of the log fold change estimate. The lowest absolute error is printed in boldface for each row.

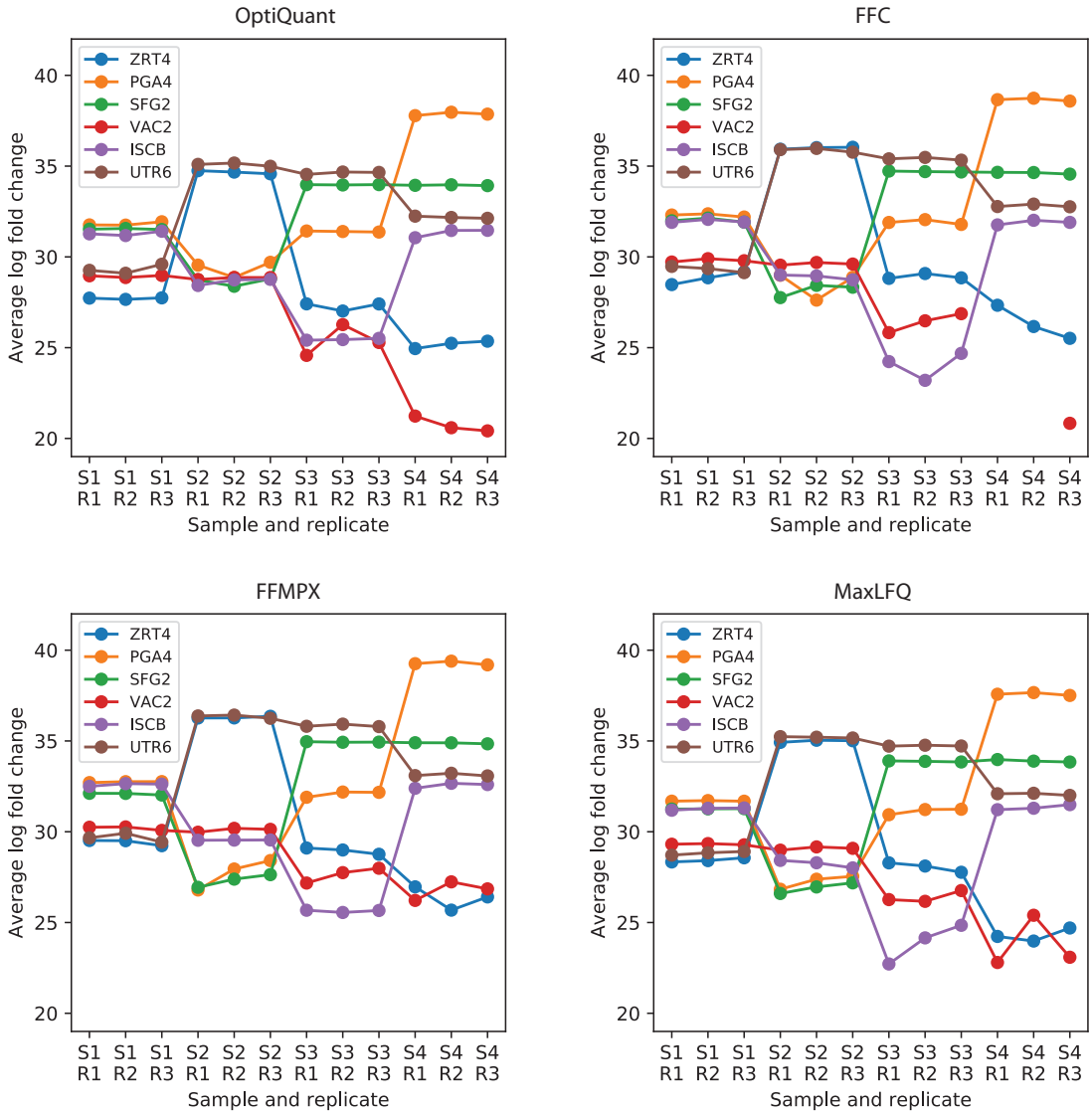

Figure A.1: Parallel coordinate plot of protein intensities (sum of peptide intensities) across all LC-MS runs of the iPRG 2015 dataset[107].

Appendix B

Abbreviations

API	application programming interface
AUC	area under the curve
BFS	breadth-first search
BSA	bovine serum albumin
CC	connected component
CID	collision-induced dissociation
CLI	command-line interface
CPU	central processing unit
CTD	Common Tool Descriptor
Da	Dalton
ESI	electrospray ionization
FDR	false discovery rate
FFC	FeatureFinderCentroided
FFMPX	FeatureFinderMultiplex
FP	false positive
FPR	false positive rate
FTICR	fourier-transform ion cyclotron
FTMS	fourier-transform mass spectrometry
FWHM	full width at half maximum
GKN	Generic KNIME Nodes
GPL	GNU General Public License
GUI	graphical user interface
gUSE	Grid and Cloud User Support Environment
HCD	higher-energy collisional dissociation
HPC	high-performance computing
HPLC	high-performance liquid chromatography

I/O	input/output
ICAT	isotope-coded afinity tagging
ID	identification
iPRG	Proteome Informatics Research Group
iTRAQ	isobaric tags for relative and absolute quantitation
KNIME	Konstanz Information Miner
LC	liquid chromatography
LFQ	label-free quantification
LOD	limit of detection
LOQ	limit of quantification
LTQ	linear trap quadrupole
m/z	mass-to-charge ratio
MAD	median absolute deviation
MALDI	matrix-assisted laser desorption/ionization
MIP	mixed-integer program
mRNA	messenger ribonucleic acid
MS	mass spectrometry
OLS	ordinary least squares
OQ	OptiQuant
pAUC	partial area under the curve
PCA	principle component analysis
PD	Proteome Discoverer (Thermo Scientific)
PEP	posterior error probability
ppm	parts per million
PPV	positive predictive value
PSM	peptide-spectrum match
QqQ	triple quadrupole
QT	quality threshold
RAM	random access memory
RF	radio frequency
RMSE	root-mean-square error
ROC	receiver operating characteristic
RT	retention time
SEM	standard error of the mean
SILAC	stable isotope labeling with amino acids in cell culture
SRM	selected reaction monitoring
STD	standard deviation
SVM	support vector machine

Th	Thompson
TIC	total ion chromatogram
TMT	tandem mass tags
TOF	time-of-flight
TOPP	The OpenMS Proteomics Pipeline
TOPPAS	The OpenMS Proteomics Pipeline Assistant
TP	true positive
TPP	Trans-Proteomic Pipeline
TPR	true positive rate
TTD	TOPP tool description
UV	ultra violet
WS-PGRADE	Web Services Parallel Grid Runtime and Developer Environment Portal
XIC	extracted ion chromatogram

Appendix C

Publications

Peer-Reviewed Journal Articles

- J Proteome Res. 2016 Sep 2;15(9):3441-8.
  LFQProfiler and RNP(xl): Open-Source Tools for Label-Free Quantification and Protein-RNA Cross-Linking Integrated into Proteome Discoverer.
  Veit J, Sachsenberg T, Chernev A, Aicheler F, Urlaub H, Kohlbacher O.

- Nat Methods. 2016 Aug 30;13(9):741-8.
  OpenMS: a flexible open-source software platform for mass spectrometry data analysis.
  Röst HL, Sachsenberg T, Aiche S, Bielow C, Weisser H, Aicheler F, Andreotti S, Ehrlich HC, Gutenbrunner P, Kenar E, Liang X, Nahnsen S, Nilse L, Pfeuffer J, Rosenberger G, Rurik M, Schmitt, **Veit J**, Walzer M, Wojnar D, Wolski WE, Schilling O, Choudhary J, Malmström L, Aebersold R, Reinert K, Kohlbacher O.

- BMC Bioinformatics. 2016 Mar 12;17:127.
  From the desktop to the grid: scalable bioinformatics via workflow conversion.
  de la Garza L, **Veit J**, Szolek A, Röttig M, Aiche S, Gesing S, Reinert K, Kohlbacher O.

- J Proteome Res. 2016 Jan 4;15(1):182-92.
  Mass-Spectrometry-Based Proteomics Reveals Organ-Specific Expression Patterns To Be Used as Forensic Evidence.
  Dammeier S*, Nahnsen S*, **Veit J***, Wehner F, Ueffing M, Kohlbacher O.

- Blood. 2014 Jan 30;123(5):e1-e10.
  Time-resolved characterization of cAMP/PKA-dependent signaling reveals that platelet inhibition is a concerted process involving multiple signaling pathways.
  Beck F, Geiger J, Gambaryan S, **Veit J**, Vaudel M, Nollau P, Kohlbacher O, Martens L, Walter U, Sickmann A, Zahedi RP.

- J Proteome Res. 2012 Jul 6;11(7):3914-20.
  TOPPAS: a graphical workflow editor for the analysis of high-throughput proteomics data.
  Junker J*+, Bielow C*, Bertsch A, Sturm M, Reinert K, Kohlbacher O.

Conference Posters

- American Society for Mass Spectrometry (ASMS) – San Antonio – 2016
  RNPxl 2 - Protein-RNA interaction site localization from UV cross-linked peptide-RNA oligonucleotides in Proteome Discoverer 2.1.
  Sachsenberg T, **Veit J**, Chernev A, Sharma K, Hofele R, Qamar S, Zaman U, Kappert C, Kramer K, Pfeuffer J, Liang X, Reinert K, Lenz C, Urlaub H, Kohlbacher O.

- Human Proteome Organization (HUPO) – Vancouver, Canada – 2016
  LFQProfiler a free plugin for labelfree quantification in Proteome Discoverer.
  Veit J, Aicheler F, Pfeuffer J, Weisser H, Sachsenberg T, Reinert K, Kohlbacher O.

- American Society for Mass Spectrometry (ASMS) – Vancouver, Canada – 2012
  An integrated framework for the assessment of statistical significance in label-free quantification of proteins.
  Junker J+, Nahnsen S, Beck F, Zahedi R, Sickmann A, Kohlbacher O.

* co-first authors
+ maiden name