Analysis of mass spectrometric data: peak picking and map alignment

Dissertation
zur Erlangung des akademischen Grades eines
Doktors der Naturwissenschaften
im Fachbereich Mathematik und Informatik
der Freien Universität Berlin

vorgelegt von

Eva Lange

Berlin 2008

Datum des Kolloquiums: 13. Juni 2008

Dekan:
Professor Dr. Ralph-Hardo Schulz

Betreuer:
Professor Dr. Knut Reinert
Freie Universität Berlin
Institut für Informatik
Algorithmische Bioinformatik
Takustraße 9
D - 14195 Berlin

Gutachter:
Professor Dr. Knut Reinert, Freie Universität Berlin, Berlin
Professor Dr. Oliver Kohlbacher, Eberhard Karls Universität Tübingen, Tübingen

Abstract

We study two fundamental processing steps in mass spectrometric data analysis from a theoretical and practical point of view.

For the detection and extraction of mass spectral peaks we developed an efficient peak picking algorithm that is independent of the underlying machine or ionization method, and is able to resolve highly convoluted and asymmetric signals. The method uses the multiscale nature of spectrometric data by first detecting the mass peaks in the wavelet-transformed signal before a given asymmetric peak function is fitted to the raw data. In two optional stages, highly overlapping peaks can be separated or all peak parameters can be further improved using techniques from nonlinear optimization. In contrast to currently established techniques, our algorithm is able to separate overlapping peaks of multiply charged peptides in LC-ESI-MS data of low resolution. Furthermore, applied to high-quality MALDI-TOF spectra it yields a high degree of accuracy and precision and compares very favorably with the algorithms supplied by the vendor of the mass spectrometers. On the high-resolution MALDI spectra as well as on the low-resolution LC-MS data set, our algorithm achieves a fast runtime of only a few seconds.

Another important processing step that can be found in every typical protocol for label-free quantification is the combination of results from multiple LC-MS experiments to improve confidence in the obtained measurements or to compare results from different samples. To do so, a multiple alignment of the LC-MS maps needs to be estimated. The alignment has to correct for variations in mass and elution time which are present in all mass spectrometry experiments. For the first time we formally define the multiple LC-MS raw and feature map alignment problem using our own distance function for LC-MS maps. Furthermore, we present a solution to this problem. Our novel algorithm aligns LC-MS samples and matches corresponding ion species across samples. In a first step, it uses an adapted pose clustering approach to efficiently superimpose raw maps as well as feature maps. This is done in a star-wise manner, where the elements of all maps are transformed onto the coordinate system of a reference map. To detect and combine corresponding features in multiple feature maps into a so-called consensus map, we developed an additional step based on techniques from computational geometry. We show that our alignment approach is fast and reliable as compared to five other alignment approaches. Furthermore, we prove its robustness in the presence of noise and its ability to accurately align samples with only few common ion species.

Acknowledgments

The research presented in this thesis was carried out in the Algorithmic Bioinformatics group of the Freie Universität Berlin from 2003 to 2007. First and foremost, I would like to thank my supervisor Knut Reinert, not only for the opportunity to pursue this research and to gain insight into the exciting field of bioinformatics, but also for creating a hospitable atmosphere in which it was a pleasure to work. Finally, I am deeply grateful for Knut's valuable scientific advice I could always rely on. I would also like to thank Oliver Kohlbacher, who co-supervised my thesis and directed me through many fruitful discussions.

This thesis can loosely be split in two parts, peak picking and map alignment. Andreas Hildebrandt helped me a great deal with my work on peak picking and supported me when I wrote my first publication. What Andreas was for peak picking, Clemens Gröpl was for map alignment—Thank you guys! I also want to thank my direct collaborators in the OpenMS project: Marc Sturm, Andreas Bertsch, Clemens Gröpl, Andreas Hildebrandt, Rene Hussong, Nico Pfeifer, Ole Schulz-Trieglaff, Alexandra Zerck, Knut Reinert, and Oliver Kohlbacher. Alexandra significantly contributed to the present thesis while writing her own Master's thesis. Ralf Tautenhahn of the Leibniz Institute of Plant Biochemistry in Halle carried out a part of the evaluation of map alignment algorithms. Especially in the early years, Andreas Döring spent endless hours teaching me how to program C++. For that I am in his debt. Michael Dorr and Caroline Kühnel read and commented on early versions of this thesis.

Last but not least I would like to thank my parents for their continuous support through all the years.

Contents

Part I

Introduction and background

Chapter 1

Guide to the thesis

This thesis is divided into three parts. In this first part, we will elaborate on mass spectrometry in general and the OpenMS framework for the analysis of mass spectrometric data. The second and the third part each present one major line of research. They both follow the same structure: we shall start with an overview of some basic theoretical concepts (Chapter 5 and 11), then give an introduction to the peak picking problem (Chapter 6) and LC-MS map alignment, respectively (Chapter 12). After a description of the state of the art and related work in Chapter 7 and 13, Chapter 8 and 14 are devoted to our own contribution. In Chapter 9 and 15, experimental results are presented; results are discussed and conclusions are drawn in Chapter 10 and 16.

An overview of the notational conventions is given in the next section. When specific terms are first defined, they are put in *italics* and their abbreviation, which may be used later on, is given in parentheses. Some useful terms are also described in the glossary.

1.1 Notations

$\mathbb{N}$	Positive integers including 0
$\mathbb{N}^+$	Positive integers excluding 0
$\mathbb{R}$	Real numbers
$\mathbb{C}$	Complex numbers
z^*	Complex conjugate of $z \in \mathbb{C}$
L^p	Space of functions such that $\int_{-\infty}^{+\infty} \lvert f(t)\rvert^p \, dt < +\infty$ with $p \in \mathbb{N}^+$
$h \star s$	Convolution of two continuous signals $h, s \in L^1$: $h \star s = \int_{-\infty}^{+\infty} h(u)s(t-u) \, du$
id	Identity transformation

Chapter 2

Motivation and own contribution

A fundamental discovery of the last century was that of the structure and importance of the genome. Over the past decades, the full genetic information could be sequenced for a variety of organisms. As of December 2007, the NCBI Genome database lists 24 eukaryotic genomes, most notably the human genome that was fully sequenced only in 2003 [Collins et al., 2003], after a draft sequence had been published two years earlier [Lander et al., 2001; Venter et al., 2001]. Despite the availability of the sequence information, many important biological questions remain unsolved. The approximately 21,000 human genes [Imanishi et al., 2004] can be expressed into more than a million human proteins [Jensen, 2004] by complex interactions such as alternative splicing and single nucleotide polymorphisms (SNPs), which take place during the transcription and translation process, as well as a plethora of possible post-translational modifications (PTMs) (see Figure 2.1). To understand living cells, these gene end products are at least as important as the genetic "blueprint" itself.

The term *proteome* was initially proposed 1994 at the first congress "From Genome to Proteome" in Siena; it referred to the description of the **prote**ins described by a gen**ome** at a particular time in a given cell, tissue, species, etc. Hence, a single genome in an organism corresponds to a multitude of proteomes, because the protein composition varies with changing conditions. Tyers and Mann [2003] expanded the term *proteomics*, the analysis of the proteome, to almost everything "post-genomics" related to proteins. Today, proteomics research is no longer limited to the study of all proteins, but includes the characterization of all protein isoforms and modifications, the interactions between them, the structural description of proteins, and their higher-order complexes.

The analysis of proteins has not just become interesting within the last decades; research has been actively pursued in this field for almost a century. However, at the time, analyzing techniques were limited and many researchers spent their whole careers on single proteins, though the consensus already was that individual proteins are not able to carry out complex biological

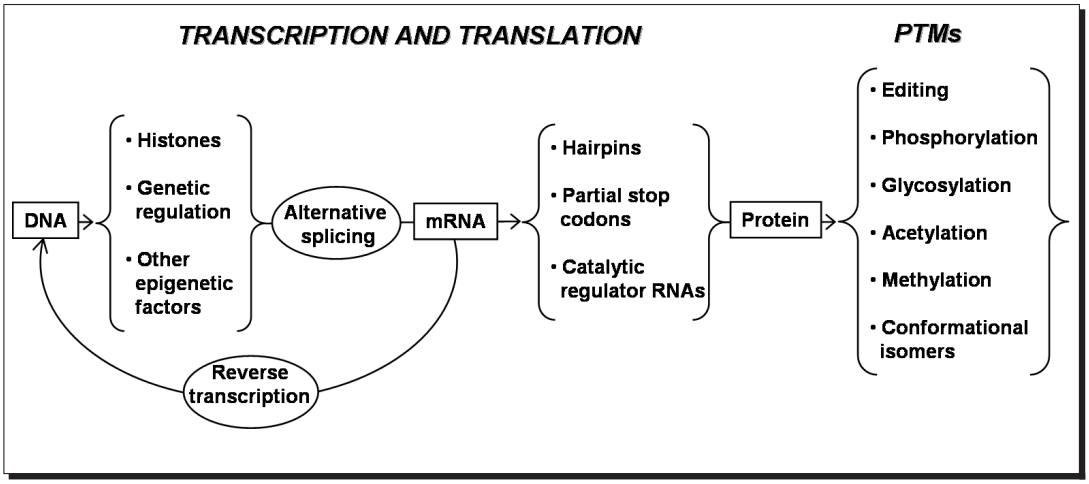

Figure 2.1: *The different subsequent steps on the way from a genetic DNA sequence to a final protein end product in eukaryotes. The protein synthesis starts with the transcription of the DNA sequence into pre-RNA. Alternative splicing and SNPs modify the pre-mRNA and result in mRNA. The nucleic mRNA sequence is afterward translated into the amino-acid sequence of the resulting protein. The protein is often chemically modified in a subsequent step. PTMs, such as glycosylation or phosphorylation, are typically performed to achieve specific functional objectives or may be the result of metabolic changes caused by disease states.*

functions and always need to interact with other proteins.

The development of a multitude of sophisticated analytical techniques within the last decades [Honoré et al., 2004] as well as the huge increment in the entries in protein and nucleic acid databases now allows for the solution of a lot of interesting cell biology questions. Today, the main approaches of proteomics research are: the analysis of protein interactions, the analysis of protein PTMs, the analysis of protein structure, and protein profiling. Protein profiling deals with the sketching of complex networks and pathways of proteins and the generation of protein-protein linkage maps. Another task is the detection of quantitative changes in protein abundance that can be used, e.g., to determine the cellular function of proteins. Furthermore, protein profiling aims at the annotation and correction of genomic sequences. The Human Proteome Initiative [O'Donovan et al., 2001] has already annotated $29,275$ human sequences in the UniProtKB/Swiss-Prot database (December 2007, release 54.6). These sequences were derived from about $17,806$ human genes. The $11,469$ additional sequences correspond to alternatively spliced isoforms. The UniProtKB/Swiss-Prot database (December 2007, release 54.6) furthermore contains $53,201$ experimental or predicted PTMs of human proteins and $37,240$ polymorphisms (many of which are linked with disease states).

Expression profiling can not only be done for a whole cell, but also on cellular compartments and organelles and their time-resolved dynamics. Thus, proteomics data also have a huge influ-

ence on clinical diagnosis [Petricoin et al., 2002] and the detection of biomarkers. Furthermore, it is essential for systems biology, which aims to combine different genomics and proteomics results obtained from the same biological system to gain a better understanding of complex biological processes [Csete and Doyle, 2002; Ideker et al., 2001].

These questions require not only an abundance of genetic information and powerful experimental techniques, but also sophisticated analytical methods to process and put together all the resulting data. *Computational proteomics* aims at the automated analysis of proteomics data, which is clearly necessary due to the high complexity and the sheer amounts of data.

In this thesis, we shall concentrate on two important steps in typical analytical procedures for proteomics data, and present efficient algorithms to analyze mass spectrometric measurements of complex protein samples. Nowadays, mass spectrometers have become the workhorse for high-throughput protein identification and quantification; in the following, we will therefore briefly describe these two mass spectrometry applications.

The analysis of proteomic samples requires a very sensitive tool since the concentrations of the proteins in a proteome can vary extremely. This so-called "dynamic range" can be up to 12 orders of magnitude in body fluids and 7 orders of magnitude in cells [Fenyo et al., 1998].

Mass spectrometry (MS) is highly sensitive, but to obtain stable molecular ions from large biomolecules such as proteins is difficult. Only the development of two soft-ionization techniques in the late 1980's, *Matrix-Assisted Laser Desorption/Ionization* (MALDI) [Karas and Hillenkamp, 1988; Tanaka et al., 1988] and *Electrospray Ionization* (ESI) [Alexandrov et al., 1984; Fenn et al., 1989; Yamashita and Fenn, 1984] allowed for the routine use of mass spectrometry as a sensitive analytical tool for complex proteomic samples.

Mass spectrometers comprise of an ion source ionizing the analyte components; a mass analyzer separating the ions according to their *mass-to-charge ratio (m/z)*; and a detector measuring the *amount of ions* or *intensity* at each m/z value unit. The one-dimensional signal resulting from a mass spectrometric measurement is called a *mass spectrum*.

The accuracy of a measured protein mass itself does not allow for a successful identification of a protein's amino-acid sequence. However, the masses of the peptides which are produced by a digestion of the protein with an enzyme of known cleavage can be used to identify the protein.

In 1993 five different groups [Henzel et al., 1993; Mann et al., 1993; Pappin et al., 1993; James et al., 1993; Yates et al., 1993] developed algorithms that use the pattern of peptide masses determined by MS together with the knowledge about the cleavage specificity of the enzyme, and a protein database to uniquely identify proteins in MS data. The main idea is to find the matching protein by correlation of the measured peptide mass pattern with theoretical peptide mass patterns, resulting from the in silico digest of the proteins in the database. This approach is called *peptide mass fingerprinting (PMF)* or *peptide mass mapping*.

The development of *tandem mass spectrometers* or *MS/MS* allowed for another protein identification technique [Hunt et al., 1986] more specific than PMF. These instruments have more than one analyzer and the idea is to isolate specific ions in the gas phase within the instrument. These so-called precursor ions are fragmented by collision-induced dissociation (CID) [Jennings, 1968] and allow the recording of *MS/MS* or *tandem spectra* [Biemann, 1992]. Since peptide ions fragment in a sequence-dependent manner, the MS/MS spectrum of a peptide, in principle, represents its amino acid sequence. Hence, given the fragment patterns in MS/MS spectra along with the m/z values of the precursor ions the proteins in a sample can be identified directly [Ma et al., 2003; Dančík et al., 1999; Zhang, 2004; Taylor and Johnson, 2001] or by means of the sequences in a protein [Tabb et al., 2001; Perkins et al., 1999; Craig and Beavis, 2004; Geer et al., 2004] or EST database [Choudhary et al., 2001]. In reference to the shotgun genomics sequence approach in which DNA is broken into smaller pieces prior to sequencing and reassembling in silico, the identification of complex protein mixtures based on the digestion of proteins into peptides and sequencing them using tandem mass spectrometry is called *shotgun proteomics*.

Peptide mixtures of very high complexity often require an additional separation step to physically separate parts of the sample prior to the injection into the mass spectrometer. One commonly used approach is LC-MS (or LC-MS/MS), which is the coupling of MS (or MS/MS) to *liquid chromatography (LC)*. In LC the analyte solvent mixture, the so-called mobile phase, is forced through a chromatographic column, the so-called stationary phase. Analyte components are separated according to their interaction with the stationary phase and therefore elute at specific time points, so-called *retention times (RT)*. The eluting analyte solvent mixture is introduced into a mass spectrometer for a determination of the mass to charge ratio of the eluting analytes. The resulting signal consists of a sequence of MS spectra. Each of these spectra, called a *scan*, represents a snapshot of the peptides eluting from the column during a fixed time interval. We call the collection of all unprocessed scans originating from an LC-MS run an *LC-MS raw map*. Each element of an LC-MS map represents the ion count that is measured at a certain RT and m/z value. Figure 2.2 shows a part of an LC-MS raw map and the mass spectrum measured after 558.88 s.

Mass spectrometry allows not only for protein identification, but also for protein quantification. Even though the relationship between the amount of analyte present in a sample and the measured signal intensity is complex and incompletely understood, MS is a proven technique for relative and absolute quantification experiments [Ong and Mann, 2005]. Therefore, the measured ion counts are used to derive a relationship between the peptide quantities of interest. Absolute quantification uses isotope-labeled homologs of specific proteolytic peptides from the target protein [Gerber et al., 2003; Gröpl et al., 2005; Mayr et al., 2006; Kirkpatrick et al., 2005]. However, relative quantification can be achieved either by a labeled or by a label-free approach. Labeled quantification uses isotope or mass tag labeling of peptides, and the two samples of interest are covalently modified by isotopically different and therefor distinguish-

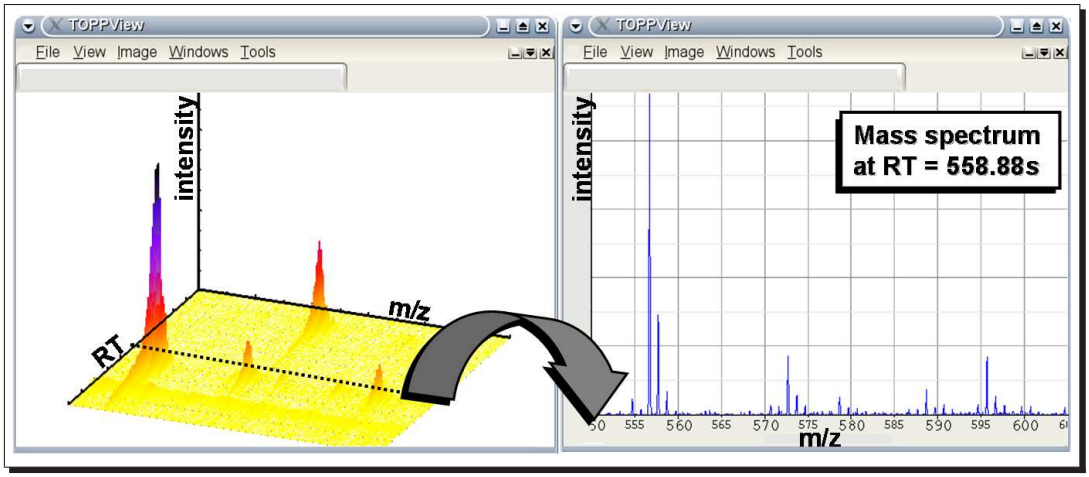

Figure 2.2: *Left: Three-dimensional plot of an LC-MS raw map. The map comprises a collection of mass spectra measured at subsequent retention times. Right: Mass spectrum obtained at* 558.88*s.*

able chemical reagents [Zhou et al., 2002; Ong and Mann, 2005]. Although these techniques bypass problems due to ion-suppressive effects of co-eluting peptides, they are often expensive and require the comparisons of only two to four samples, which prevent retrospective comparisons and complicate large studies with multiple samples. The label-free quantification approach is a promising alternative and allows for the quantitative comparison of multiple samples. Several studies have demonstrated that mass spectral peak intensities of peptide ions correlate well with protein abundances in complex samples [Bondarenko et al., 2002; Wang et al., 2003; Schulz-Trieglaff et al., 2007; Old et al., 2005].

A successful protein identification and quantification requires the accurate and precise determination of the m/z and intensity values corresponding to all peptides in a probe. Typically, an analysis pipeline that extracts the information of interest from the LC-MS data is composed of the following operations (steps of particular relevance to this thesis are printed in bold)

- signal filtering and baseline removal: remove noise and baseline artifacts,
- **peak picking: find and extract the accurate positions, heights, total ion counts, and FWHM values of all mass spectral peaks,**
- identification algorithm: identify the proteins in a sample given the mass spectral peak information,
- feature detection and quantification: detect and extract patterns of peaks that correspond to the same charge variant of a peptide,
- intensity normalization: normalize the ion counts,

- **multiple map alignment: correct the distortion of the RT and m/z dimension of multiple raw or feature maps; in case of feature maps, assign corresponding features afterward**,

- classification algorithms and biomarker discovery: find differentially expressed peak or feature patterns that can be used to classify samples, e.g., from different cell states.

Depending on the underlying type of mass spectrometer, a raw LC-MS map can have a size of several hundred megabytes up to several gigabytes, whereas only a small fraction of data contains the signal of interest. This accentuates the need for fast and effective algorithms for each of the analysis steps mentioned above to allow for high throughput proteomics approaches.

Two essential steps for the analysis of MS-based proteomics data are peak picking and multiple LC-MS map alignment. In the following, we will introduce both problems and outline our novel approaches to solve them.

Peak picking in mass spectra

The detection and extraction of mass spectral peaks plays an important role in each identification and quantification analysis pipeline. Whereas a reliable protein or peptide identification mainly depends on the accurate and precise determination of the m/z values of the peptide ions, quantification needs exactly determined ion counts corresponding to the peptides in a sample. A general approach, which extracts all the mentioned characteristics of the interesting signal, even of low abundant peaks, without any loss of information, would facilitate not only protein identification and quantification, but also biomarker discovery.
Each peak picking algorithm is confronted with a number of problems due to the nature of mass spectrometric data. An ideal mass analyzer would be able to distinguish ions even with slightly different m/z values, but as in all physical experiments, a mass spectrum is afflicted with uncertainties resulting from random fluctuations in measurement. Furthermore, chemical noise and baseline artifacts might also perturb results. A typical mass spectrometric measurement is shown in Figure 2.2. Since the measurement of the same peptide ions does not result in a single impulse at a certain m/z value, but in an asymmetric peak-shaped response, the detection of the correct m/z value is hampered. Due to limitations of mass resolution and high charge states, mass spectral peaks might overlap strongly.

Each peak picking technique should overcome the mentioned difficulties above and extract the information of interest of all mass spectral peaks. Almost all state-of-the-art algorithms are custom-tailored for either identification, quantification, or biomarker discovery. However, we developed the first generic peak picking that yields all relevant information in one step. Even in the presence of noise and baseline artifacts, it computes accurately the m/z position,

the maximum intensity, the total ion count, and the full-width-at-half-maximum of each mass spectral peak (see Figure 2.3) not only for well-resolved, but also for overlapping peaks.

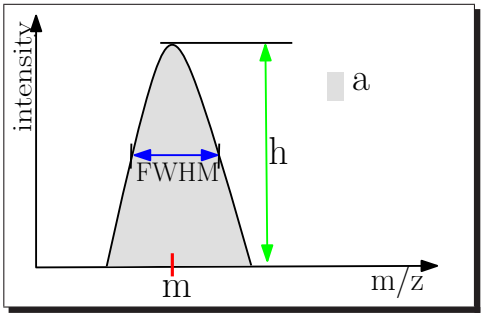

Figure 2.3: *Important features of a mass spectral peak: position m, height (maximum intensity) h, full width at* 50% *height (FWHM), and the total ion count a.*

Our algorithm furthermore extracts information about the peak shape, which might facilitate further analysis steps. The method uses the multiscale nature of spectrometric data by first detecting the mass peaks in the wavelet-transformed signal before a given asymmetric peak function is fitted to the raw data. In two optional stages, the resulting fit can be further improved and strongly overlapping peaks can be separated using techniques from nonlinear optimization. The algorithm does not make assumptions about the underlying machine or ionization method, which makes the algorithm robust for different experimental settings, and achieves real-time performance.

Multiple LC-MS map alignment

Application scenarios for the quantitative information in LC-MS maps range from additive series in analytical chemistry over analysis of time series in expression experiments to applications in clinical diagnostics. A common requirement is that the same peptides in different measurements have to be related to each other; in other words, multiple LC-MS maps have to be aligned. Such an alignment can either be computed on raw, unprocessed LC-MS maps at the beginning of a comparative proteomics data analysis pipeline or it can be computed on extracted features at the end of the pipeline. Due to experimental uncertainties, the problem stays the same in both cases: distorted retention time and m/z positions of the elements of an LC-MS map. To overcome this problem and to allow for the assignment of corresponding peptides in different maps, these distortions have to be corrected (see Figure 2.4).

We developed the first formal definition of the multiple LC-MS raw and feature map alignment problem using a new distance function for LC-MS maps, which takes the different grade of distortion of the two dimensions into account. Transforming the estimation of a suitable mapping into a well-known problem of computational geometry, the partial approximative point pattern

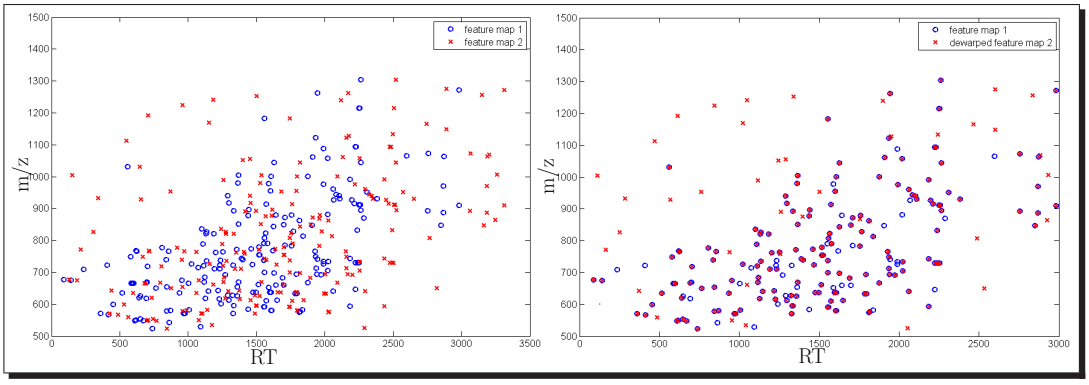

Figure 2.4: *Left: Two feature maps are shown that share* 80% *of common elements, but the strong distortion of the RT dimension masks the correspondence. Right: A proper correction of the distortion of feature map 2 (dewarping) superposes the corresponding features of the two maps.*

matching problem, we developed a fast and effective solution based on the pose-clustering approach. This so-called superposition algorithm is generic and might be used to map LC-MS raw maps onto each other and thereby to solve the multiple raw map alignment problem. Furthermore, it enables the superposition of multiple feature maps. For the solution of the multiple feature map alignment problem, a subsequent processing step is necessary. Hence, we developed a sophisticated grouping algorithm based on a nearest neighbors search that assigns corresponding features in multiple feature maps and computes a so-called consensus map.

The algorithms for peak picking as well as for LC-MS map alignment are integrated into *OpenMS* [Sturm et al., 2008], an open-source framework for the analysis of mass spectrometric data. Furthermore, they are available as a command line tool in the OpenMS proteomics pipeline *TOPP* [Kohlbacher et al., 2007].

Chapter 3

Mass spectrometry in proteomics

The following sections will introduce the reader to the field of mass spectrometry based proteomics. We will shortly summarize the principles of mass spectrometry and present the ideas of tandem mass spectrometry as well as LC-MS, which is a combination of high performance liquid chromatography and mass spectrometry. For a deeper insight into these topics we refer the reader to the literature, e.g., Lehmann [1995]; Smith [2005]; Jurisica and Wigle [2005]; Aebersold and Mann [2003]; Cañas et al. [2006].
In the last two sections we introduce MS-based protein identification and quantification, which are the most important applications of mass spectrometry in the field of proteomics.

3.1 Principles of mass spectrometry

Nowadays, mass spectrometers are well-established instruments for the analysis of proteomic samples. They produce ions of the analytical compounds and separate these ions according to their *mass-to-charge ratios (m/z)*. The measurement is carried out by the three components of a mass spectrometer: an *ion source*, a *mass analyzer*, and a *detector*. In Figure 3.1 the most popular MS components are listed.

Ion source. In the ion source the analyte components are ionized. The two main soft-ionization technologies that produce stable molecular ions from large biomolecules are *Electrospray Ionization* (ESI) [Alexandrov et al., 1984; Fenn et al., 1989; Yamashita and Fenn, 1984] and *Matrix-Assisted Laser Desorption/Ionization* (MALDI) [Karas and Hillenkamp, 1988; Tanaka et al., 1988].

During the ESI process, the liquid analyte solution containing the peptide/protein sample is

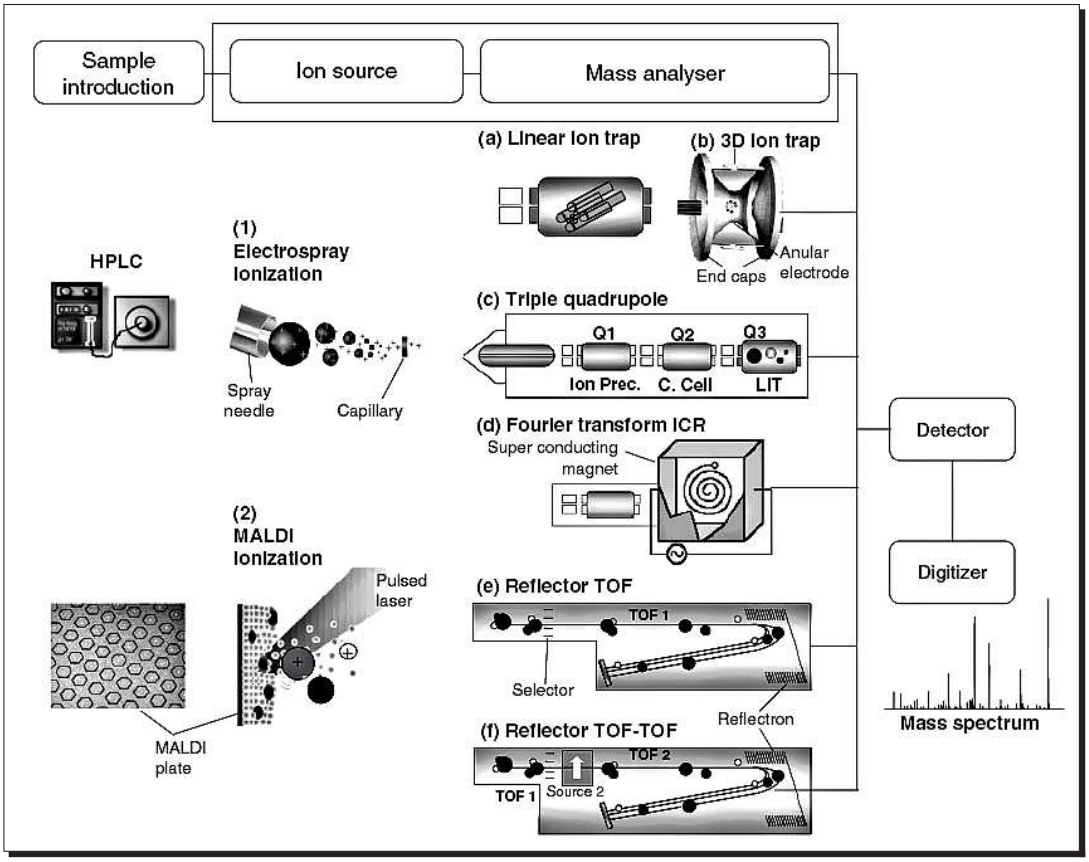

Figure 3.1: *Main components of a mass spectrometer (figure taken from Cañas et al. [2006]). Sample introduction device, ionization source for ion generation, mass analyzer for ion separation, and ion detector to transform analogue signals into digital signals and record a mass spectrum. Common ionization sources for proteomic research are ESI and MALDI. Widespread mass analyzers are ion traps (a) linear, and (b) three-dimensional; (c) triple quadrupoles; (d) Fourier transform cyclotrons; (e) and (f) time-of-flight (TOF). Usually ion trap and quadrupole analyzers are coupled to ESI ion sources, whereas TOF analyzers are usually combined with MALDI ion sources.*

forced through a narrow-bore spray capillary, to which a high potential has been applied. The high potential causes the emerging solution to disperse into a fine spray of charged droplets. These micro-drops evaporate very quickly until the number of charges on their surface becomes very high and surpasses the Rayleigh limit, at which point they explode forming smaller micro-drops. This process continues until the analyte ion escapes the droplet, which is called ion desorption. The ESI process generates ions in multiple charge states.

Because ESI produces ions directly from solution, it is easily coupled to a liquid-chromatography (see Section 3.1.2) or capillary electrophoresis system, which separates the protein mixture over time.

The ionization during the MALDI process is based upon the ultraviolet light (UV) absorption capability of a matrix compound. In a first step, the matrix and the peptide/protein sample are mixed in an appropriate solvent and spotted onto a MALDI plate. After the evaporation of the solvent, co-crystallized analyte molecules embedded in matrix crystals are obtained. When a laser is fired at the crystals in the MALDI spot the energy is absorbed by the matrix, which is partially vaporized and which carries intact analyte molecules into the gas phase. During the expansion of the MALDI plume, protons are exchanged between analytes and matrix molecules, resulting in the formation of positively and negatively charged analyte molecules.

Mass analyzer. The mass analyzer separates the ions according to their *mass-to-charge ratio (m/z)*. This is achieved by the generation of electric or magnetic fields inside the instrument. These fields separate the ions influencing their spatial trajectories, velocity, or direction. The four basic types of mass analyzers used in proteomics research are the ion trap, Fourier transform ion cyclotron (FT-MS), time-of-flight (TOF), and quadrupole analyzers. They differ in design and performance, each with its own strength and weakness.

Ion trap analyzers capture or "trap" ions for a certain time interval and allow for the mass analysis of the trapped ions by the variation of the amplitude of an impressed high-frequency storage field. These instruments are robust, sensitive and relatively inexpensive, but they have a relatively low mass accuracy. The linear or two-dimensional ion trap is more sensitive and has higher resolution and mass accuracy than traditional, three-dimensional ion traps.
The FT-MS instruments are also trapping mass analyzers, although they capture the ions under high vacuum in a high magnetic field. Their strengths are high sensitivity, mass accuracy, resolution, and dynamic range. But in spite of the enormous potential, the expense, operational complexity, and low peptide-fragmentation efficiency of FT-MS instruments have limited their routine use in proteomics research.
TOF analyzers are the simplest mass analyzers. They essentially consist of a flight tube in high vacuum. The square root of the flight time of an ion along the tube is proportional to its mass, and lighter ions arrive at the detector more quickly than those of higher mass. The pathway for the ions in a TOF is reversed and enlarged using an electrostatic mirror to reflect ions at the end of the field-free region. The electrostatic mirror might compensate for small kinetic energy differences of ions by allowing a deeper penetration of faster ions. In a TOF-TOF instrument two TOF sections are separated by a collision cell. The collision cell allows for the selection of ions of a particular m/z value in a first TOF mass analyzer and the measurement of a mass spectrum of the fragmented ions in the second TOF analyzer. TOF instruments have high sensitivity, resolution, and mass accuracy.
Quadrupole analyzers separate ions by time-varying electric fields between four rods, which permit a stable trajectory only for ions of a particular desired m/z. Despite triple quadrupole systems having a limited mass range, they are useful in the highly selective ion-scanning mode that is optimized for monitoring precursor ions with particular features of interest.

Detector. The detector registers the relative abundance of ions at each m/z value. The resulting measurement called *mass spectrum* consists of a plot of ion abundance versus its m/z ratio (see Figure 2.2).

The different set-ups of the MS components account for the different mass spectrometry platforms. MALDI is usually coupled to TOF analyzers that measure the mass of intact peptides, whereas ESI has typically been coupled to ion traps and triple quadrupole instruments. But within the last years new configurations of ion sources and mass analyzers have found wide application in protein analysis. To allow the fragmentation of MALDI-generated precursor ions, MALDI ion sources have recently been coupled to quadrupole ion-trap mass spectrometers and TOF instruments.
In the following section we will briefly introduce tandem mass spectrometry, which allows for the fragmentation of selected precursor ions.

3.1.1 Tandem mass spectrometer (MS/MS)

Tandem mass spectrometry is a specialized MS technique that allow for peptide "sequencing". Tandem mass spectrometers comprise at least two mass analyzers (e.g., the triple-quadrupole in Figure 3.1) and therefore this technique is also called *MS/MS*. Peptide ions of interest are first selected in a precursor ion scan. Typically, the computer controlling the tandem mass spectrometer automatically selects those ions with a high abundance. These so-called precursor or parent ions are electromagnetically isolated and subjected to energetic collisions in order to induce peptide fragmentation. The *collision induced dissociation (CID)* of peptides results in a range of structurally significant product ions. In most cases, the peptide bond is being cleaved between the carbonyl-carbon and the amide-nitrogen. If the charge remains on the N-terminal fragment the ion is called *b-ion* and if the C-terminal fragment retains the charge it is called *y-ion*.
Given the m/z values and intensities of the b- and y-ions along with the m/z value of the precursor ion, the peptide sequence can be determined automatically (see Section 3.2).

3.1.2 Liquid chromatography-mass spectrometry (LC-MS)

For the analysis of peptide mixtures of high complexity, mass spectrometers are often coupled to liquid chromatography (LC) to gain a second physical separation of the analytical compounds. The LC step spreads out the parts of the sample solution over time on the basis of some property of the molecules, such as hydrophobicity. Therefore, the analyte is solved in a liquid, which is called *mobile phase*, and then forced through a column of the so-called *stationary phase* with high pressure. Reverse-phase high performance liquid chromatography, for example, uses a tubular column packed with some material made up of hydrophobic molecules.

Depending on the specific chemical or physical interactions of the analytical components with the stationary phase, they are retarded in the column for a certain time. The time at which a specific analyte elutes from the column, is called the *retention time (RT)*. In LC-MS the liquid that elutes from the column is directly introduced into a mass spectrometer and at certain points in time a mass spectral measurement of eluting droplets is obtained. This results in a collection of consecutively determined mass spectra, whereby each mass spectrum is labeled with a unique retention time.

The LC step in LC-MS experiments might avoid two undesired effects of MS. Firstly, peptides with the same m/z values might have different RT values and are not analyzed by the mass spectrometer at the same time, such that the ambiguity in the MS signal is reduced. Secondly, the number of ions being simultaneously analyzed by the mass spectrometer is decreased. Effects such as ion suppression, where one ion's signal suppresses the signal of another ion, are diminished [Annesley, 2003].

Liquid chromatography carried out with mobile phases of fixed composition or eluent strength, so-called isocratic elution, generally does not work well for proteomic samples. The time periods t that are needed until all molecules of a certain protein/peptide are eluted can vary strongly and a single mobile phase does not provide adequate separation. Furthermore, the retention of protein/peptide molecules can be extremely sensitive to small changes in mobile phase composition, which results in varying t values for the same compound in two different measurements. Other undesired effects caused by the application of isocratic separation to a mixture of macromolecules are that, usually, some sample components elute immediately (with no separation), whereas other components elute so slowly that it appear as if they never leave the column.
The application of *gradient elution* should avoid the mentioned drawbacks of isocratic elution. In gradient elution the mobile phase is continuously changed during separation, such that the retention of later peaks is continually reduced; that is, the mobile phase becomes steadily stronger as the separation proceeds.
The gradient separates the analyte mixtures as a function of the affinity of the analyte for the current mobile phase composition relative to the stationary phase.
Most gradient separations use linear gradients, where the affinity of the analyte for the mobile phase composition relative to the stationary phase is linearly increased over time. Other popular gradients are: gradient delay, and step gradient [Snyder and Dolan, 2007].
With gradient elution, there is a much smaller problem with irreproducible retention times for large molecules; nevertheless, the variation of retention times of the same compound in different measurements is quite strong and has to be corrected before different LC-MS measurements can be compared.

3.2 Protein identification

In this section we introduce two popular protein/peptide identification approaches.

Peptide mass fingerprint Although the accurate mass measurement of a protein does not allow for the unique identification of the protein, its cleavage products can be used to determine the protein identity. Digesting the protein using an enzyme of known cleavage, the accurate determined masses of the resulting peptides can be used as a *peptide mass fingerprint (PMF)* of the protein.
Typically, the PMF strategy starts with an initial separation of proteins in a sample by SDS-PAGE. Separated proteins are afterward visualized (by staining with silver nitrate) and usually digested in-gel with specific enzymes (e.g., trypsin). For the subsequent MS analysis, the resulting proteolytic peptides are extracted from the gel piece. To achieve highly accurate mass measurements, TOF analyzers are typically used in combination with MALDI or ESI ion sources. The peptide mass fingerprint is determined by the extraction of the set of measured peptide masses. With this method proteins in mixtures of low complexity can be identified with good high throughput compatibility [Pappin et al., 1993] and a high sensitivity even below the femtomole range [Schuerenberg et al., 2000].

In 1993 five different groups [Henzel et al., 1993; Mann et al., 1993; Pappin et al., 1993; James et al., 1993; Yates et al., 1993] developed algorithms that use the experimental mass profile and match it against the theoretical masses obtained from the in-silico digestion at the same enzyme cleavage sites of all protein amino acid sequences in the database. The proteins in the database are then ranked according to the number of peptide masses matching their sequence within a given mass error tolerance.

Peptide fragmentation data The identification of proteins from tandem mass spectra of their proteolytic peptides [Hunt et al., 1986] represents a more specific identification method than peptide mass fingerprinting and even allows for the analysis of complex proteomics mixtures.
This approach, just like the PMF method, requires the digestion of the proteins in a sample with an enzyme of known cleavage. Based on the shotgun genomics sequence approach in which DNA is broken into smaller pieces prior to sequencing and reassembling in silico, the identification of complex protein mixtures based on the digestion of proteins into peptides and sequencing them using tandem mass spectrometry is called *shotgun proteomics*.
As mentioned in Section 3.1.1, the collision-induced dissociation of peptides results in a range of structurally significant b and y product ions. These ions of overlapping sequence fragments allow for the determination of the full amino acid sequence by the calculation of the mass difference between fragment ions differing by one amino acid. The direct derivation of the peptide sequence from the tandem spectrum is called *de novo* sequencing [Ma et al., 2003;

Dančík et al., 1999; Zhang, 2004; Taylor and Johnson, 2001]. A major advantage of this approach is that it does not require any protein database and even allows for the identification of unknown proteins. However, it requires the correct determination of the ion types and is confounded by factors such as noise, missing peaks, and additional peaks.
Other protein identification approaches use the protein sequences in databases [Tabb et al., 2001; Perkins et al., 1999; Craig and Beavis, 2004; Geer et al., 2004] or nucleotide data, as the incomplete nucleotide sequences contained in the diverse EST databases [Choudhary et al., 2001]. Given the m/z value of a precursor ion, a database of predicted MS/MS spectra is created for all matching peptides using the rules of peptide fragmentation. The experimental MS/MS spectrum is compared to all predicted spectra and the best matching peptides are determined using a predefined scoring system.

3.3 Protein quantification

Many proteomic studies require the relative or absolute amounts of the proteins present in a biological sample: the spectrum ranges from additive series in analytical chemistry [Gröpl et al., 2005], over analysis of time series in expression experiments [Bisle et al., 2006; Niittylä et al., 2007], to applications in clinical diagnostics [Vissers et al., 2007], in which we want to find statistically significant markers describing certain disease states. Despite the relationship between the amount of analyte present in a sample and the measured signal intensity being complex and incompletely understood, it could be shown that mass spectral peak intensities of peptide ions correlate well with protein abundances in complex samples [Bondarenko et al., 2002; Wang et al., 2003; Schulz-Trieglaff et al., 2007; Old et al., 2005], and that the comparison of signals from the same peptide under different conditions can give a rough estimate of relative protein abundance between multiple proteomes [Ong and Mann, 2005].
Mass spectrometry allows for the determination of two different quantitative pieces of information. Absolute quantification experiments estimate the amount of the substance in question, whereas in relative quantification experiments the amount of substance is defined in relation to another measure of the same substance. In the following two sections we introduce two approaches to determine the quantitative information of interest.

3.3.1 Label-free quantification

Label-free quantification [Bondarenko et al., 2002; Wang et al., 2003; Schulz-Trieglaff et al., 2007; Old et al., 2005] is a promising method for relative quantification. Signal intensities of the same peptide in different LC-MS maps are compared directly. Not only does this approach require the accurate determination of the signal intensities belonging to a certain peptide, but it also needs the correct assignment of corresponding peptide signals across different measure-

ments.

3.3.2 Labeled quantification

Labeled quantification uses the stable isotope labeling of proteins or peptides to determine their absolute or relative quantities. This approach is mainly designed to quantify proteomes of only two to four different states. To this end, the proteins/peptides of one sample are labeled and afterward combined with the unlabeled sample. Depending on the labeling method the same proteins/peptides show a specific mass difference in the mass spectrometric measurement. Several methods have been developed, which are mainly distinguished by the way the stable isotope labels are introduced into the protein or peptide [Ong and Mann, 2005]: spiking in an isotopically labeled analog [Gerber et al., 2003; Gröpl et al., 2005; Mayr et al., 2006; Kirkpatrick et al., 2005], incorporation through an enzyme during protein digestion [Yao et al., 2001, 2003], introducing a chemical, isotopically labeled tag onto peptides or proteins [Gygi et al., 1999; Ross et al., 2004], or having cells that incorporate the label metabolically [Oda et al., 1999; Ong et al., 2002].
Although these techniques bypass problems due to ion-suppressive effects of co-eluting peptides, which can affect label-free quantification experiments, they are very costly and prevent retrospective comparisons and complicate large studies with multiple samples.

Chapter 4

OpenMS—An open-source framework for mass spectrometry

The high complexity and the sheer amount of MS-based proteomics data require sophisticated analytical methods. The information extraction from LC-MS data can be classified into a series of smaller analysis steps

- signal filtering and baseline removal: remove noise and baseline artifacts,
- peak picking: find and extract the accurate positions, heights, total ion counts, and FWHM values of all mass spectral peaks,
- identification algorithm: identify the proteins in a sample given the mass spectral peak information,
- feature detection and quantification: detect and extract patterns of peaks that correspond to the same charge variant of a peptide,
- intensity normalization: normalize the ion counts,
- multiple map alignment: correct the distortion of the RT and m/z dimension of multiple raw or feature maps; in case of feature maps, assign corresponding features afterward,
- classification algorithms and biomarker discovery: find differentially expressed peak or feature patterns that can be used to classify samples, e.g., from different cell states.

A label-free quantification protocol might consist of a process involving signal filtering and baseline removal, peak picking, quantification, normalization, multiple map alignment, and marker finding. On the other hand, an identification pipeline might be composed of signal

filtering and baseline removal, a peak picking step, and an identification algorithm. Small algorithmic components for each analytical step allow for the development of tools for both analytical aims and might be readily combined into more complex workflows or tools.

In 2003, the Algorithmic Bioinformatics group at the Freie Universität Berlin and the Department for Simulation of Biological Systems of Tübingen University initiated an academic project for proteomic data analysis that realizes the modular idea of problem solving. *OpenMS—a framework for mass spectrometry* [Sturm et al., 2008] is flexible and serves as a framework for developing mass spectrometry data analysis tools, providing everything from basic data structures over file input/output (I/O) and visualization to sophisticated algorithms for the analysis steps mentioned above. Thus, OpenMS allows developers to focus on new algorithmic approaches instead of implementing infrastructure. The high flexibility of OpenMS stands out against other existing academic tools for proteomic data analysis, e.g., MapQuant [Leptos et al., 2006], MASPECTRAS [Hartler et al., 2007], msInspect [Bellew et al., 2006], MZMine [Katajamaa et al., 2006], SpecArray [Li et al., 2005], Trans-Proteomic Pipeline (TPP) [Keller et al., 2005], Viper [Monroe et al., 2007], Superhirn [Mueller et al., 2007], and XCMS [Smith et al., 2006]. These tools are typically monolithic and hard to adapt to new experiments. Furthermore, they often concentrate on only one step of the analysis, e.g., quantification, peptide identification, or map alignment, or combine a few steps into a pipeline.

4.1 The map concept

The data that is produced by the combination of multi-dimensional LC and subsequent MS can be viewed as a set of multidimensional discrete points. In LC-MS such a data point is described by retention time, m/z, and intensity. The collection of all these data points is called an *LC-MS raw map*. The analysis of this raw data is done through several steps, which in our view correspond to a series of map transformations. Figure 4.1 shows the map types and transformation steps.

Signal filtering and baseline removal steps are performed on raw LC-MS maps. The output of these transformations is again an preprocessed LC-MS raw map. Depending on the underlying type of mass spectrometer, a raw LC-MS map can have a size of several hundred megabytes up to several gigabytes, whereas only a small fraction of data contains the signal of interest. Thus, data reduction is a central concept of OpenMS. It comprises two transformation steps, which are peak picking and feature detection and quantification. During the peak picking process, the mass spectral peaks are detected and important information, such as their accurate positions, heights, total ion counts, and FWHM values, is extracted. We call the resulting data of a peak picking step a *LC-MS peak map*. The subsequent feature detection and quantification step is again a data reduction step, at which the two-dimensional signals created by some chemical

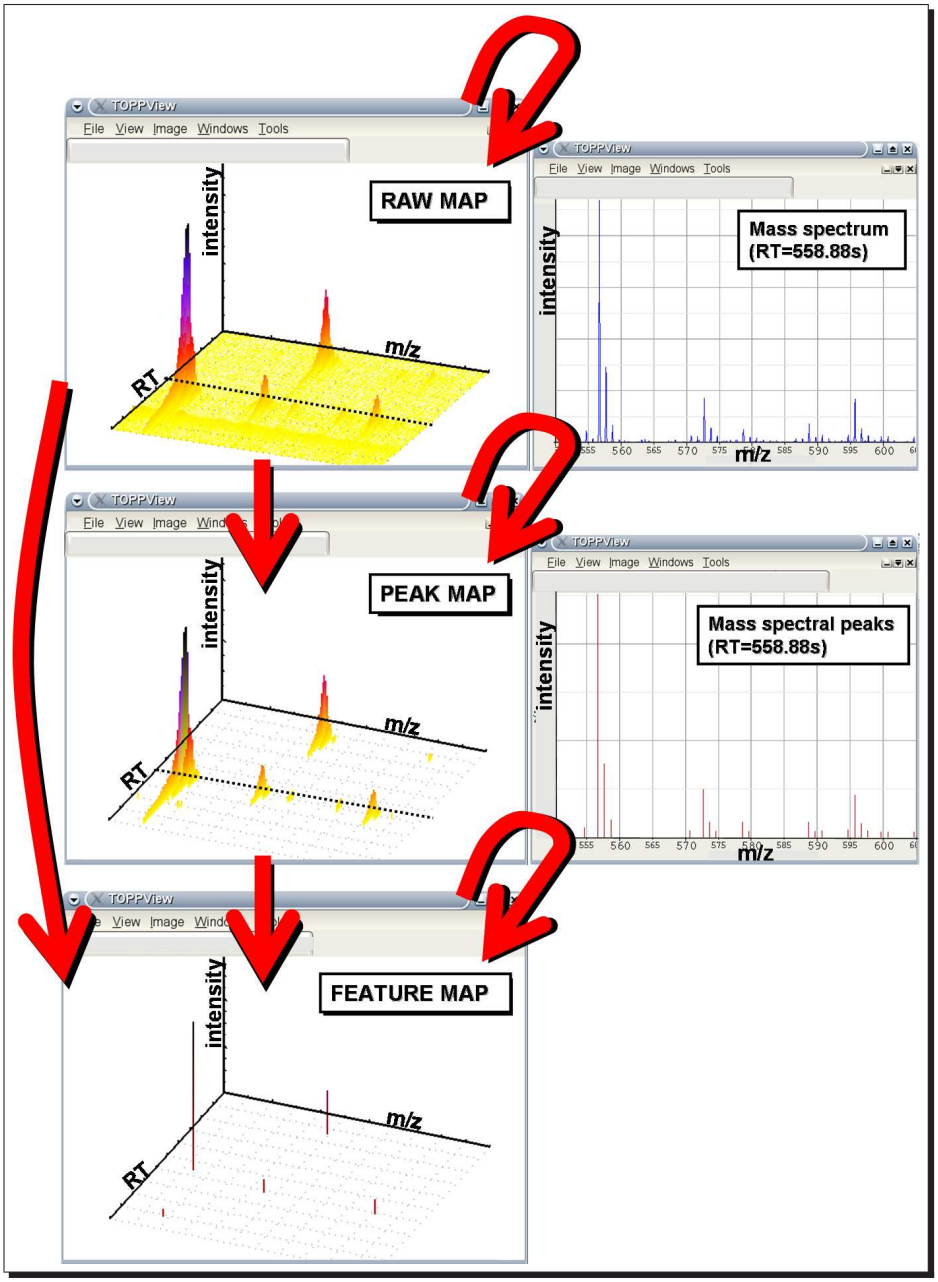

Figure 4.1: *Top: An LC-MS raw map and its mass spectrum at RT=558.88 s. Middle: The corresponding LC-MS peak map and the extracted mass spectral peaks at RT=558.88 s. Bottom: The corresponding feature map. The red arrows indicate the possible transformations.*

entities (e.g., peptides) are grouped together into a so-called *LC-MS feature map*. A feature is characterized by its isotopic pattern in mass-to-charge dimension and by the elution profile

in retention time dimension. Features have summary coordinates, such as centroid retention time, average monoisotopic peak position, or summed intensities. The feature detection and quantification step can either be performed on a peak LC-MS map or a raw LC-MS map. A transformation, such as intensity normalization, can be performed on either raw, peak, or feature maps, whereas the output type of this operation is the same as the input type.

4.2 Design and implementation

OpenMS is intended to offer a rich functionality while keeping in mind the design goals of ease-of-use, robustness, extensibility, and portability. We will now briefly describe the techniques used to achieve these goals. The subsequent sections describe the overall architecture and the features of OpenMS.

4.2.1 Design goals

Ease-of-use. The object-oriented programming paradigm aims at mapping real-world entities to comprehensible data structures and interfaces. Combining it with a coding style that enforces consistent names of classes, methods, and member variables leads to intuitive usability of a software library. For these reasons we adapted this paradigm for OpenMS. A second important feature of a software framework is documentation. We decided to use Doxygen [van Heesh] to generate the class documentation from the source code, which ensures consistency of code and documentation. The documentation is generated in HTML format making it easy to read with a web browser. OpenMS also provides a tutorial that introduces the most important concepts and classes using example applications.

Robustness. Although robustness is not crucial when developing new algorithms, it is essential if a new method will be applied routinely to large scale datasets. Typically, there is a trade-off between performance and robustness. OpenMS tries to address both issues equally. In general, we try to tolerate recoverable errors, e.g., files that do not entirely fulfill the format specifications. On the other hand, exceptions are used to handle fatal errors. To check for correctness, unit tests are implemented for each method of a class. These tests check the behavior for both valid and invalid use. Additionally, preprocessor macros are used to enable additional consistency checks in debug mode, which are then disabled in productive mode for performance reasons.

Extensibility. Since OpenMS is based on several external libraries, it is designed for the integration of external code. All classes are encapsulated in the *OpenMS* namespace to avoid

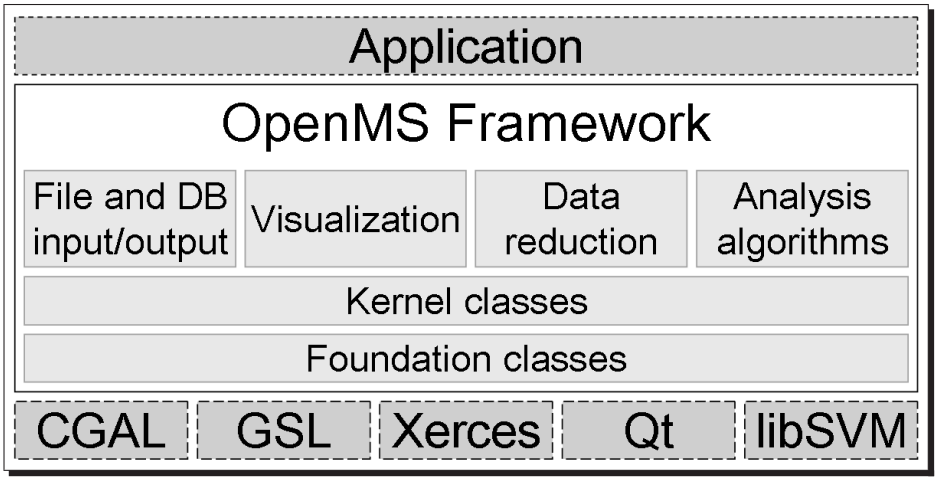

Figure 4.2: *The overall design of OpenMS (figure taken from [Sturm et al., 2008].*

symbol clashes with other libraries. Through the use of template code, many data structures are adaptable to specific problems. For example, it is possible to replace the representation of the mass-spectrometric peak or to replace the container in which a spectrum stores the peaks. Also, OpenMS supports standard formats and is itself open-source software. The use of standard formats ensures that applications developed with OpenMS can be easily integrated into existing analysis pipelines. OpenMS source code is located on SourceForge [SourceForge], a repository for open-source software. This allows users to participate in the project and to contribute to the code base.

Portability. Currently, OpenMS can be compiled on most Unix-like platforms (e.g., MacOS, Solaris, Linux) and has been thoroughly tested on several Linux distributions. Through the use of ANSI C++, porting it to other platforms poses no major problem.

The second emphasis of OpenMS, besides the design goals, is rich functionality. The framework offers data structures to handle MS data and metadata. It supports visualization of the data, file I/O, and database I/O. This more basic functionality is complemented by a variety of algorithms for data analysis. All key analysis steps such as signal processing, quantification, and peptide identification are addressed. The overall architecture and some selected features are illustrated in the following sections.

4.2.2 Overall architecture and features

The overall design of OpenMS is shown in Figure 4.2. From a bird's-eye view, the OpenMS concept is quite simple. Applications can be implemented using OpenMS, which in turn relies on several external libraries: Qt [QT] provides visualization, database support, and a platform abstraction layer. Xerces [XERCES] allows XML file parsing. libSVM [Chang and Lin] is used for machine learning tasks. The Computational Geometry Algorithms Library (CGAL) [Overmars, 1996; Fabri et al., 1996] provides data structures and algorithms for geometric computations. The GNU Scientific Library (GSL)[Galassi et al.] is used for different mathematical and statistical tasks.

OpenMS can itself be subdivided into several layers. At the very bottom are the foundation classes, which implement low-level concepts and data structures. They include basic concepts (e.g., factory pattern, exception handling), basic data structures (e.g., string, points, ranges) and system-specific classes (e.g., file system, time). The kernel classes, which capture the actual MS data and metadata, are built upon the foundation classes. Finally, there is a layer of higher-level functionality that relies on the kernel classes. This layer contains database I/O, file I/O supporting several file formats, data reduction functionality, and all other analysis algorithms.

4.3 Example algorithms and features

In the following we will present some algorithms for different analysis steps.

4.3.1 Standardized file formats

Standardized data exchange formats are especially important because they allow the easy integration of different software tools into a single analysis pipeline. Therefore, OpenMS supports most non-proprietary file formats, e.g., mzData[Orchard et al., 2006] and mzXML[Pedrioli et al., 2004]. As there are no standard file formats for quantification and peptide identification data, we created our own formats for these tasks (featureXML and idXML). Eventually, these formats will be replaced by standard formats released by the HUPO-PSI. Currently, we are actively contributing to the development of the upcoming standards mzML and analysisXML. mzML is intended to replace both the mzData and the mzXML format. analysisXML captures the results of peptide and protein search engines.

4.3.2 Database support

Most tools developed so far operate on files. Because of the constantly growing data volume created by LC-MS experiments, database systems will become more and more important for data management. Therefore, we developed a database adapter that can persistently store the kernel data structures in an SQL database. Through the use of Qt database adapters as an additional layer of abstraction, the implementation is able to employ most SQL compliant relational database management systems (including MySQL, PostgreSQL, ORACLE, and DB2).

4.3.3 Visualization

A very useful tool for data analysis is visual inspection. It can instantly reveal properties of the data that would go unnoticed using command line tools. Errors during separation or polymeric contamination of the sample can, for example, be easily noticed during visual inspection of an LC-MS map. OpenMS provides widgets that display a single spectrum or a peak map. A single spectrum is displayed by a standard plot of raw or peak data. A peak map is displayed either in a 2D view from a bird's-eye perspective with color-coded intensities or in a 3D view. Figures 2.2 and 4.1 show examples of the 3D map and the spectrum view.

4.3.4 Signal processing

OpenMS offers several filters to reduce chemical and random noise as well as baseline trends in MS measurements. Raw spectra may either be de-noised by a Savitzky-Golay filter or a peak-area-preserving Gaussian low-pass filter. Both smoothing filters are commonly used and recommended for spectrometric data [Savitzky and Golay, 1964; Press et al., 2002].
For the baseline in MS experiments, no universally accepted analytical expression exists. Hence, we decided to implement a non-linear filter, known in morphology as the top-hat operator [Soille, 1998]. This filter does not depend on the underlying baseline shape and its applicability to MS measurements has already been shown in [Breen et al., 2000].

4.3.5 Peak picking

For the extraction of the accurate information about the mass spectral peaks in a raw spectrum we developed an efficient peak picking algorithm [Lange et al., 2006] that uses the multiscale nature of spectrometric data. First, the peak positions are determined in the wavelet-transformed signal. Afterward, important peak parameters (centroid, area, height, full-width-at-half-maximum, signal-to-noise ratio, asymmetric peak shape) are extracted by fitting an asymmetric peak function to the raw data. In two optional steps overlapping peaks can be

separated, or the resulting fit can be further improved by using techniques from nonlinear optimization. In contrast to currently established techniques, our algorithm yields accurate peak positions even for noisy data with low resolution and is able to separate overlapping peaks of multiply charged peptides.

Our peak picking algorithm is described in more detail in Chapter 8.

4.3.6 Feature detection and quantification

Feature detection is a central concept in OpenMS. As noted before, a feature is a signal in an LC-MS map, which is, e.g., caused by a peptide ion. Each feature is characterized by its mass-to-charge ratio, the centroid of its elution curve, and the signal area.

OpenMS includes several algorithms for the detection of peptidic features in LC-MS data, tailored for datasets of different mass resolutions and measured on various instrument types. Our approaches are based on a two-dimensional model. We use the concept of an average amino acid (also called *averagine*) to approximate the amino acid composition for a peptide of a given mass. From this we can estimate its atomic composition and derive its isotope distribution in a mass spectrum [Horn et al., 2000]. Similarly, we approximate the elution curve by a Gaussian or exponentially modified Gaussian distribution [Di Marco and Bombi, 2001]. In addition, our isotope pattern model takes different mass resolutions into account by incorporating a parameter for the width of the isotopic peaks in a feature.

Fitting the two-dimensional model is a relatively expensive computational task. Therefore, it is important to select the candidate regions carefully. We designed a novel algorithm [Schulz-Trieglaff et al., 2007] that uses a hand-tailored isotope wavelet [Hussong et al., 2007] to filter the mass spectra for isotopic patterns for a given charge state. The isotope wavelet explicitly models the isotope distribution of a peptide. This pre-filtering results in a lower number of potential peptide candidates that need to be refined using the model fit.

4.3.7 LC-MS map alignment

An important step in a typical LC-MS analysis workflow is the combination of results from multiple experiments, e.g., to improve confidence in the obtained measurements or to compare results from different samples. In order to do so, a suitable mapping or *alignment* between the datasets needs to be established. The alignment has to correct for (random and systematic) variations in the observed elution time and mass-to-charge ratio that are inevitable in experimental datasets.

OpenMS offers algorithms to align multiple experiments and to match the corresponding ion species across many samples [Lange et al., 2007] . A novel and generic algorithm was devel-

oped to correct for the variation of retention time and mass-to-charge dimensions between two maps. It uses an adapted pose clustering approach [Ballard, 1981; Stockman et al., 1982] to efficiently superimpose raw maps as well as feature maps.

To detect and combine corresponding features in multiple feature maps into a so-called *consensus map*, we developed an algorithm based on techniques from computational geometry. The superimposition algorithm and the algorithm for the determination of a consensus map are combined to a star-wise approach for the alignment of multiple raw or feature maps. Overall, the methods are fast, reliable, and robust, even in the presence of many noise signals and large random fluctuations of retention time.

Our alignment approach is described in more detail in Chapter 14.

4.3.8 Retention time prediction

A major problem with existing tandem mass spectrometry identification routines lies in the significant number of false positive and false negative annotations. Until now, standard algorithms for protein identification have not used the information gained from separation processes usually involved in peptide analysis, such as retention time information, which are readily available from chromatographic separation of the sample. Identification can thus be improved by comparing measured to predicted retention times. Current prediction models are derived from a set of measured test analytes but they usually require large amounts of training data.
OpenMS offers a new kernel function, the *paired oligo-border kernel (POBK)*, which can be applied in combination with support vector machines to a wide range of computational proteomics problems. This enables the user to predict peptide adsorption/elution behavior in strong anion-exchange solid-phase extraction (SAX-SPE) and ion-pair reversed-phase high-performance liquid chromatography (IP-RP-LC). Using the retention time predictions for filtering significantly improves the fraction of correctly identified peptide mass spectra. OpenMS offers a wrapper class to the libsvm [Chang and Lin] for support vector learning. Our *POBK* is well-suited for the prediction of chromatographic separation in computational proteomics and requires only a limited amount of training data. Usually 40 peptides or less are sufficient. A more detailed description of the methods for retention time prediction, as well as the application of the retention time prediction to improve tandem MS identification results, can be found in [Pfeifer et al., 2007].

4.4 TOPP—The OpenMS Proteomics Pipeline

OpenMS has been successfully used for the implementation of *TOPP—The OpenMS Proteomics Pipeline* [Kohlbacher et al., 2007]. TOPP is a set of computational tools that can

be chained together to tailor problem-specific analysis pipelines for LC-MS data. It transforms most of the OpenMS functionality into small command line tools that are the building blocks for more complex analysis pipelines. Each tool handles a well-defined functionality in the area of proteomics data analysis. The functionality of the tools ranges from data preprocessing (e.g., file format conversion, baseline reduction, noise reduction, peak picking, map alignment) over quantification (labeled and label-free) to identification (wrapper tools for Mascot [Perkins et al., 1999], Sequest [Tabb et al., 2001], InsPecT [Tanner et al., 2005] and OMSSA [Geer et al., 2004]). The individual applications range from very trivial to rather complex tasks, but their combined value arises from the fact that they share a common interface, common formats, and common configuration files. They can thus be combined like building blocks to perform more complex analysis tasks, an idea already used in similar toolboxes in bioinformatics, e.g., in EMBOSS [Olson, 2002]. Chaining is achieved through makefiles, simple shell scripts, or as components of complex workflow systems in distributed or GRID environments, e.g., by workflow systems such as Taverna [Oinn et al., 2004]. In order to make the TOPP components easy to combine, we only use standard file formats such as mzData and analysisXML. This also facilitates the integration of external tools supporting standard formats. A pipeline-specific control file provides parameters to all components and directs the data flow between them. In the control file, a set of parameters for each individual invocation of a tool can be provided. For tasks that cannot be done with TOPP, wrapper components are provided to integrate commonly used applications. Furthermore, manual analyses during the development of a pipeline are supported through a system of log files allowing the reconstruction of every processing step. The debugging output can be turned off as soon as the pipeline works as intended.

One of the design goals is user-friendliness. Hence, all TOPP components share a common base interface and provide a detailed description for all parameters. Additionally, a full documentation of all components and examples is available on our web site.

Part II

Peak picking

Chapter 5

Mathematical preliminaries

The following sections should provide the reader with some mathematical background for our peak picking algorithm proposed in Chapter 8. We introduce some basic statistical terms and summarize the mathematical background of the continuous wavelet theory as well as the Levenberg-Marquardt algorithm.

5.1 Uncertainties in measurements

The following overview of uncertainties in measurements is based on Bevington and Robinson [2002]. In all physical experiments errors and uncertainties result from *random fluctuations* in measurement and *systematic errors*. That means if we make a measurement x_1 of a quantity x, we expect our observation x_1 to approximate the quantity, but we do not expect that the measured and the true value are equal. Let us first consider the *random error* and neglect the systematic error. An N-fold repetition of the experiment would distribute the observed values $x_1, \ldots, x_N$ around the correct value x. If we could make an infinite number of measurements then we could describe exactly the distribution of the data points and understand the process that generates the data points. In practice, however, we can only hypothesize the existence of such a distribution that determines the probability of getting any particular observation in a single measurement. This distribution is called the *parent distribution*. Similarly, we can hypothesize that the acquired data points are samples from the parent distribution. They form the so-called *sample distribution*. In the limit the sample distribution becomes the parent distribution.

The *mean* $\bar{x}$ of an experimental distribution is the sum of all measurements x_i divided by N

$$\frac{1}{N}\sum_{i=0}^{N-1} x_i$$

and the mean μ of the parent population is defined as the limit

$$\mu = \lim_{n\to\infty}\left(\frac{1}{N}\sum_{i=0}^{N-1} x_i\right).$$

The mean is equivalent to the *centroid* or *average value* of the quantity *x*.
A measure of how far the samples fluctuate from the mean is the *standard deviation* σ. It is the square root of the *variance* σ^2, which represents the power of the fluctuation

$$\sigma^2 \equiv \lim_{N\to\infty}\left(\frac{1}{N}\sum_{i=0}^{N-1}(x_i-\mu)^2\right) = \lim_{N\to\infty}\left(\frac{1}{N}\sum_{i=0}^{N-1}{x_i}^2\right) - \mu^2.$$

The standard deviation defines the width of the distribution and for this reason it acts as an indicator for the repeatability of a measurement. Without the knowledge of any "true" value it is a measure for the *precision* of a measurement.
A measure of how close the result of an experiment comes to the "true" value is defined by the term *accuracy*. It defines the deviation of the mean $\bar{x}$ from the "true" value *x* and results from systematic errors. These are errors that may result from faulty calibration of equipment or from bias on the part of the observer. They become repeated in exactly the same manner each time the measurement is conducted. These errors are not easy to detect and not easily studied by statistical analysis and must be determined from an analysis of experimental conditions and techniques.
For any experiment precision and accuracy must be considered simultaneously; high precision measurements that are highly inaccurate as well as accurate measurements with low precision are both useless.

5.2 Introduction to wavelet theory

The analysis of signals (e.g., a recorded speech signal, or a mass spectrum) requires the determination of a suitable representation of the signal. A representation of the signal by a series of coefficients, based on an analysis function, facilitates the analysis procedure. This can be achieved by a transformation, or decomposition of the signal on a set of basis functions prior to processing in the transform domain. One example of a signal transformation is the transformation from the time domain to the frequency domain. The oldest method for this is the Fourier transform developed in 1807 by Joseph Fourier. In 1980 the French seismologist Jean Morlet initiated the formalism of Wavelet theory, which is another very powerful transformation method. In contrast to the Fourier transform the Wavelet transform determines not only information about the frequencies in a signal, but it also preserves information about the localization of the different frequencies in the signal in a near-optimal manner.
Wavelet theory can be divided into the following main categories:

1. Continuous wavelet transforms (CWT).
2. Discrete wavelet transforms (DWT): (a) orthonormal bases of wavelets and (b) redundant discrete systems (frames).

In the following two sections, we will introduce the reader into the classical Fourier transform and the windowed Fourier transform as well as the limitations of these methods. This should facilitate the idea and theory of the CWT described in Section 5.2.3. These sections are based on [Mallat and Hwang, 1992; Mallat, 1998; Valens, 2004; Kaiser, 1994; Alsberg et al., 1997] and on the lecture "Digital signal processing I" held by Til Aach at the University of Lübeck. Readers who are also interested in the DWT and wavelet applications, e.g., multiresolution analysis, are referred to literature with more extensive wavelet theory coverage [Mallat, 1998; Kaiser, 1994].

5.2.1 Classical Fourier transform

The standard *Fourier transform* or the *Fourier integral S* of a signal $s \in L^1$ is defined as:

$$S(f) := \int_{-\infty}^{+\infty} s(t) \exp(-j2\pi ft)\, dt \text{ with } j := \sqrt{-1}, \tag{5.1}$$

where S measures "how much" of oscillations at the frequency f there is in s. A useful way of understanding the Fourier transform is to say that the signal s has been projected onto a set of basis functions $s_E := \exp(j2\pi ft) = \cos(2\pi ft) + j\sin(2\pi ft)$. The basis functions in this case are the cosine and sine functions represented by complex exponential functions. If $s \in L^1$ this integral does converge and

$$|S(f)| \leq \int_{-\infty}^{+\infty} |s(t)|\, dt \leq +\infty. \tag{5.2}$$

The Fourier transform is thus a bounded function and it is continuous because

$$|S(f) - S(\zeta)| \leq \int_{-\infty}^{+\infty} |s(t)|\, |\exp(-j2\pi ft) - \exp(-j2\pi\zeta t)|\, dt \leq |f - \zeta| \int_{-\infty}^{+\infty} |s(t)|\, dt. \tag{5.3}$$

If s is also integrable, the *inverse Fourier transform* is given by

$$s(t) := \int_{-\infty}^{+\infty} S(f) \exp(j2\pi ft)\, df. \tag{5.4}$$

The inversion formula Equation 5.4 decomposes s as a sum of sinusoidal waves of amplitude $S(f)$. By using this formula, as in Equation 5.3 we can show that the hypothesis $S \in L^1$ implies that s must be continuous. The reconstruction Equation 5.4 is therefore not proved for discontinuous functions. This motivates an extension of the Fourier transform to the space L^2 of functions s with a finite energy $\int_{-\infty}^{+\infty} |s(t)|^2\, dt < +\infty$. By working in the *Hilbert space* L^2 of

functions, we also have access to all facilities provided by the existence of an inner product. The inner product or *cross correlation* of $s \in L^2$ and $g \in L^2$ is given by

$$\langle s, g \rangle = \int_{-\infty}^{+\infty} s(t) g^*(t)\, dt,$$

where g^* denotes the complex conjugate of g. The resulting norm in L^2 is

$$\|s\|^2 = \langle s, s \rangle = \int_{-\infty}^{+\infty} |s(t)|^2\, dt.$$

The inner product and norms in L^2 are conserved by the Fourier transform up to a factor of 2π and it holds the *Parseval formula*

$$\int_{-\infty}^{+\infty} s(t) g^*(t)\, dt = \frac{1}{2\pi} \int_{-\infty}^{+\infty} S(f) G^*(f)\, df$$

and the *Plancherel formula* with

$$\int_{-\infty}^{+\infty} |s(t)|^2\, dt = \frac{1}{2\pi} \int_{-\infty}^{+\infty} |S(f)|^2\, df.$$

In the following, we will introduce two important properties of the Fourier transform that are used later.

Theorem 5.2.1: Scaling. Let $s \in L^1$ and $a \in \mathbb{R}$. The Fourier transform of $s(at)$ is given by

$$\frac{1}{|a|} S\left(\frac{f}{a}\right).$$

That means a compression in time is equivalent to stretching the Fourier spectrum and scaling all frequency components up by a factor of $|a|$.

The most important property of the Fourier transform for signal processing applications is the *convolution theorem*. It is another way to express the fact that sinusoidal waves s_E are eigenvalues of convolution operators "$\star$".

Theorem 5.2.2: Convolution. Let $s \in L^1$ and $h \in L^1$. The function

$$g := h \star s = \int_{-\infty}^{+\infty} h(u) s(t-u)\, du$$

given by the convolution of the signal s with h is in L^1 and

$$G(f) = H(f) S(f).$$

A convolution is an integral that expresses the amount of overlap of one function as it is shifted over another function. Theorem 5.2.2 states that the convolution in the time domain equals a multiplication in the frequency domain.

The response $g := s \star h$ of a linear time-invariant system can be calculated from its Fourier transform $G(f) = S(f)H(f)$ with the inverse Fourier formula

$$g(t) = \int_{-\infty}^{+\infty} G(f) \exp(j2\pi ft)\, df,$$

which yields

$$g(t) = \int_{-\infty}^{+\infty} H(f)S(f) \exp(j2\pi ft)\, df.$$

Each frequency component $S(f)$ is amplified or attenuated by $H(f)$. Such a convolution is thus called *frequency filtering*, and $H(f)$ is the *transfer function* of the filter.

The big disadvantage of a Fourier transform is that it has only frequency resolution and no time resolution. Although all frequencies present in a signal can be determined, information about their locations in the signal is not provided. In the past decades several solutions have been developed to overcome this problem. They are based on a representation of the signal in the time and frequency domain at the same time. To achieve a joint time-frequency representation the signal of interest is cut into several parts and the parts are analyzed separately. Although this approach of signal analysis will give more information about the when and where of different frequency components, it is not clear how to cut the signal. The windowed Fourier transform, introduced in the following section, represents a feasible solution to this problem.

5.2.2 Windowed Fourier transform

In 1946, Dennis Gabor found a solution to the problem of missing time resolution in the Fourier transform. He introduced *windowed Fourier atoms* to measure localized frequency components of sounds. The idea is to use a window of finite length and move it along the signal in question. For each sliding step an FT on that local region in time is calculated. Gabor used a real and symmetric window $g(t) = -g(t)$ that is translated by b and modulated at the frequency ζ:

$$g_{b,\zeta}(t) = \exp(j2\pi\zeta t)g(t-b).$$

It is normalized so that $\|g_{b,\zeta}\| = 1$ for any $(b,\zeta) \in \mathbb{R}^2$. The resulting *windowed Fourier transform* $STFT(b,\zeta)$ of $s \in L^2$ is

$$STFT(b,\zeta) = \langle s, g_{b,\zeta} \rangle = \int_{-\infty}^{+\infty} s(t)g(t-b) \exp(-j2\pi\zeta t)\, dt.$$

This transform is also called the *short time Fourier transform (STFT)* because the multiplication by $g(t-b)$ localizes the Fourier integral in the neighborhood of $t = b$.

The signal is decomposed into a set of basis functions that are windowed versions of the original sine and cosine functions. Accordingly, the results from an STFT analysis can be understood as a projection onto each of these basis functions located in time and frequency.

In the STFT we have to fix the length of the window as well as to select the type of window function. Both will be affecting to the resolution, either in the time or frequency domain. Resolution can be intuitively understood as the degree of detail that is shown in each domain. A short window length will have a good time resolution, i.e., we can see detailed changes happening in time. Suppose that we want to know exactly all the frequency components present at a certain moment in time. Cutting out only this very short time window using a Dirac pulse and transforming it to the frequency domain would fail because the problem here is that cutting the signal corresponds to a convolution between the signal and the cutting window. Since multiplication in the time domain is identical to convolution in the frequency domain (see Theorem 5.2.2) and since the Fourier transform of a Dirac pulse contains all possible frequencies, the frequency components of the signal will be smeared out all over the frequency axis (see Theorem 5.2.1). A large window will have opposite properties: poor resolution in the time domain and good resolution in the frequency domain. It is useful here to imagine the window as a box containing sinusoidal waves. Since the box has a finite length, there must be a lower limit to the frequencies of the waves it can contain. If the wavelength of the wave is too large it cannot fit into the box. If we start out with a large window, there will be a lower limit in the transform to the resolving power along the frequency direction. The upper limit to the frequency resolution corresponds to the sampling frequency of the discrete signal at hand. If a new STFT is performed with a shorter window size, there will be a new lower limit to the frequency resolution. If we have a signal containing spikes, there will be problems with localizing in time those spikes with a large window (blurring). The resolution in the frequency domain, however is very good. Decreasing the window size will reduce the blurring along the time direction but increase it in the frequency direction. One of the purposes with using the wavelet transform is to improve on the resolution problem. This will, in this case, correspond to selecting different sizes of the sliding window according to the frequency range we wish to investigate.

5.2.3 Continuous wavelet transform (CWT)

The wavelet transform or wavelet analysis is probably the most recent solution to overcome the shortcomings of the Fourier transform and determines information about both domains at the same time. The FT assumes that the frequency content of the signal is constant throughout the entire signal and thus that it is effectively periodic. Thereby, a FT over the whole time domain does not allow to focus on local frequency distribution variations.

In wavelet analysis, the usage of so-called *wavelets*, that are fully scalable modulated functions

solves the signal-cutting problem. The wavelet is shifted along the signal and for every position the spectrum is calculated. This process is repeated with varying wavelet width (scale), which results in a collection of time-scale representations of the signal, all with different resolutions. On the large scale global properties can be seen, whereas the small scales show the details. Thus, going from large scale to small scale is, in this context, equal to zooming in.

A function $\psi \in L^2$ with zero average:

$$\int_{-\infty}^{+\infty} \psi(t)\, dt = 0 \tag{5.5}$$

is called a *mother wavelet*. It is normalized, $\|\psi\| = 1$, and centered in the neighborhood of $t = 0$. The *wavelets* are generated from this single basic wavelet by scaling $\psi(t)$ by a and translating it by b:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right), \tag{5.6}$$

whereby all wavelets remain normalized with $\|\psi_{a,b}\| = 1$. The *continuous wavelet transformation* or the wavelet integral of $s \in L^2$ at b and scale a is defined as

$$W_s(a,b) = \langle s, \psi_{a,b}\rangle = \frac{1}{\sqrt{a}}\int_{-\infty}^{+\infty} s(t)\psi^*_{a,b}\left(\frac{t-b}{a}\right)\, dt. \tag{5.7}$$

Equation 5.7 can also be rewritten as a convolution

$$W_s(a,b) = \frac{1}{\sqrt{a}}\int_{-\infty}^{+\infty} s(t)\psi^*(t)\, dt = s \star \overline{\psi}_a \tag{5.8}$$

with

$$\overline{\psi}_a = \frac{1}{\sqrt{a}}\psi^*\left(\frac{-t}{a}\right).$$

Using the Theorem 5.2.1, the Fourier transform of $\overline{\psi}_a$ is given by

$$\overline{\Psi}(f) = \sqrt{a}\Psi(f)^*(af) \tag{5.9}$$

whereby $\Psi(f)$ is the Fourier transform of $\psi(t)$. Since $\Psi(f)(0) = \int_{-\infty}^{+\infty}\psi(t)\, dt = 0$ it appears that Ψ is the transfer function of a bandpass filter. The convolution in Equation 5.8 computes the wavelet transform with dilated bandpass filters.

The most important properties of wavelets are the *admissibility* and the *regularity conditions*. The admissibility condition

$$C_\psi = \int_0^{+\infty} \frac{|\Psi(f)|^2}{f}\, df < +\infty$$

guarantees the reconstruction of square integrable functions $\psi \in L^2$ without loss of information. To ensure that this integral is finite $\Psi(0) = 0$ must hold, which explains why we imposed

that wavelets must have zero average. If furthermore $\Psi(f)$ is continuously differentiable the admissibility condition is satisfied.
Regularity is a quite complex concept and we will give only an idea about it by using the concept of vanishing moments. If we expand the wavelet transform Equation 5.7 into a Taylor series $\gamma(a,b)$ at $t=0$ until order n (let $b=0$ for simplicity) we get [Sheng, 1996]:

$$\gamma(a,0) = \frac{1}{\sqrt{a}}\left[\sum_{i=0}^{n} s^{(i)}(0)\int_{-\infty}^{+\infty}\frac{t^i}{i!}\psi\left(\frac{t}{a}\right)\,dt + O(n+1)\right]. \tag{5.10}$$

Here, $s^{(i)}$ stands for the i-th derivative of s and $O(n+1)$ means the rest of the expansion. If we define the moments of the wavelet by

$$M_i = \int_{-\infty}^{+\infty} t^i \psi(t)\,dt$$

then we can rewrite Equation 5.10 into the finite development

$$\gamma(a,0) = \frac{1}{\sqrt{a}}\left[s(0)M_0 a + \frac{s^{(1)}(0)}{1!}M_1 a^2 + \frac{s^{(2)}(0)}{2!}M_2 a^3 + \ldots + \frac{s^{(n)}(0)}{n!}M_n a^{n+1} + O(a^{n+2})\right] \tag{5.11}$$

Resulting from the admissibility condition it holds $M_0 = 0$ for the zeroth moment M_0 and therefore the first term in the right-hand side of Equation 5.11 is zero. If we now manage to make the other moments up to M_n zero as well, then the wavelet transform coefficients $\gamma(a,b)$ will decay as fast as a^{n+2} for a smooth signal $s(t)$. In the literature these are known as the vanishing moments of a wavelet. If a wavelet has N vanishing moments, then the approximation order of the wavelet transform is also N. Hence, with a wavelet of order N any polynomial signal up to order $N-1$ can be represented completely in scaling space. Accordingly, more vanishing moments means that the scaling function can represent more complex signals accurately.

5.3 The Levenberg-Marquardt method for non-linear least squares fitting

The following introduction into non-linear least squares fitting and the derivation of the *Levenberg-Marquardt algorithm* is based upon Madsen et al. [2004].
The *non-linear least squares problem* is defined as follows:

Definition 5.3.1: Given a vector function $f:\mathbb{R}^n \to \mathbb{R}^m$ with $m \geq n$. We want to minimize $\|f(a)\|$, or equivalently to find

$$a^* = \arg\min_a \{F(a)\} \tag{5.12}$$

where

$$F(a) = \frac{1}{2}\sum_{i=1}^{m}(f_i(a))^2 = \frac{1}{2}\|f(a)\|^2 = \frac{1}{2}f(a)^T f(a). \tag{5.13}$$

An important source of non-linear least squares problems is data fitting, where we are given a set of data points $\{(x_1,y_1),\ldots,(x_n,y_n)\}$ and a model $M(a,x_i)$ that depends on the parameters $a=(a_1,\ldots,a_n)^T$. We assume that there exists a parameter set a^+ so that $y_i = M(a^+,x_i)+\varepsilon_i$ where the $\{\varepsilon_i\}$ are measurement errors on the data ordinates, assumed to behave like random noise. For a least squares fit we might determine the minimizer a^* by computing the residuals $f_i(x) = y_i - M(a,x_i)$ $(i=1,\ldots,m)$ for any choice of a and take the parameters which result in the minimal sum of squared residuals. The global minimizer is very hard to find in general, and in the following we will concentrate on solving the simpler problem of finding a local minimizer for F.
The *local minimization problem* is given by

Definition 5.3.2: Given $F:\mathbb{R}^n\to\mathbb{R}$. Find a^* such that $F(a^*)\leq F(a)$ for $\|a-a^*\|<\delta$.

We will now define some conditions of a *local minimizer* a^* that might be used to solve the local minimization problem. Assume that the so-called *cost function* F is differentiable and so smooth that the following Taylor expansion is valid,

$$F(a+h) = F(a)+h^Tg+\frac{1}{2}h^THh+O(\|h\|^3), \tag{5.14}$$

where g is the *gradient*,

$$g\equiv F'(x)(a) := \begin{pmatrix}\frac{\partial F}{\partial a_1}\\ \vdots \\ \frac{\partial F}{\partial a_n}\end{pmatrix} \tag{5.15}$$

and H is the *Hessian matrix*

$$H\equiv F''(x)(a) := \begin{pmatrix}\frac{\partial^2 F}{\partial a_1\partial a_1} & \cdots & \frac{\partial^2 F}{\partial a_1\partial a_n}\\ \vdots & \ddots & \vdots \\ \frac{\partial^2 F}{\partial a_n\partial a_1} & \cdots & \frac{\partial^2 F}{\partial a_n\partial a_n}\end{pmatrix}. \tag{5.16}$$

If a^* is a local minimizer and $\|h\|$ is sufficiently small, then we cannot find a point a^*+h with a smaller F-value. Combining this observation with Equation 5.14 we get

Theorem 5.3.1: Necessary condition for a local minimizer. If a^* is a local minimizer, then $g^*\equiv F'(a^*)=0$.

We call a parameter set a_s that satisfies the necessary condition a *stationary point* for F. Thus, a local minimizer is also a stationary point, but so is a local maximizer. A stationary point that is neither a local maximizer nor a local minimizer is called a *saddle point*. In order to determine whether a given stationary point is a local minimizer or not, we need to include the

second order term in the Taylor series Equation 5.14. Inserting a_s, we see that

$$F(a_s+h) = F(a_s) + \frac{1}{2}h^T H_s h + O(\|h\|^3) \tag{5.17}$$

with $H_s = F''(a_s)$. From the Definition 5.16 of the Hessian it follows that any H is a symmetric matrix. Furthermore, if we request that H_s is positive definite, then its eigenvalues are greater than some number $\delta > 0$, and $h^T H_s h > \delta\|h\|^2$. This shows that for $\|h\|$ sufficiently small the third term on the right-hand side of Equation 5.17 will vanish. Since $\frac{1}{2}h^T H_s h$ is positive we get a sufficient condition for a local minimizer:

Theorem 5.3.2: Sufficient condition for a local minimizer. Assume that a_s is a stationary point and that $F''(a_s)$ is positive definite. Then a_s is a local minimizer.

If H_s is negative definite, then a_s is a local maximizer. If H_s is indefinite (i.e., it has both positive and negative eigenvalues), then a_s is a saddle point.

All methods for non-linear optimization search iteratively for the local minimizer a^*. From a starting point a_0 the method produces a series of vectors $a_1, a_2, \ldots$, which is assumed to converge to a^*, a local minimizer for the given function, see Theorem 5.3.2. Most methods have measures that enforce the descending condition

$$F(a_{k+1}) < F(a_k). \tag{5.18}$$

This condition should avoid the convergence to a maximizer or a saddle point.
The so-called *steepest descent methods* or *gradient methods*, which are introduced in the next section, satisfy the descending condition Equation 5.18 in each step of the iteration. One step from the current iterate a_k consists in: 1. Find a descent direction h_d, and 2. Find a step length giving a good decrease in the F-value. Therefore the variation of the F-value along the half line starting at a and with direction h is considered. From the Taylor expansion Equation 5.14 we see that

$$\begin{aligned} F(a+\alpha h) &= F(a) + \alpha h^T F'(a) + O(\alpha^2) \\ &\simeq F(a) + \alpha h^T F'(a) \text{ for } \alpha \text{ sufficiently small.} \end{aligned}$$

We say that h is a *descent direction* if $F(a+\alpha h)$ is a decreasing function of α at $\alpha = 0$. This leads to the following definition.

Definition 5.3.3: For F at a, h is a descent direction if $h^T F'(a) < 0$.

If no such h exists, then $F'(a) = 0$, showing that in this case a is stationary. Otherwise, we have to choose α, i.e., how far we should go from a in the direction given by h_d, so that we get a decrease in the value of the objective function.

5.3.1 The Steepest Descent method

From Definition 5.3.3 we see that when we perform a step αh with positive α, then the relative gain in function value satisfies

$$\lim_{\alpha \to 0} \frac{F(a) - F(a+\alpha h)}{\alpha \|h\|} = -\frac{1}{\|h\|} h^T F'(a) = -\|F'(a)\| \cos(\theta)$$

where θ is the angle between the vectors h and $F'(a)$. This shows that we get the greatest gain rate if $\theta = \pi$, i.e., if we use the steepest descent direction h_{sd} given by

$$h_{sd} = -F'(a). \tag{5.19}$$

In Section 5.3.3 we will describe a powerful non-linear optimization technique that does not need the implementation of second derivatives and combines the steepest descent method with the *Gauss-Newton method* that is presented in the next section.
In the remainder of this section we introduce some formulas of derivatives of F, which we will need in the following.

Provided that f has continuous second partial derivatives, we can write its Taylor expansion as

$$f(a+h) = f(a) + J(a)h + O(\|h\|^2) \tag{5.20}$$

where $J \in \mathbb{R}^{m \times n}$ is the Jacobian matrix. This is a matrix containing the first partial derivatives of the function components,

$$J(a) := \begin{pmatrix} \frac{\partial f_1}{\partial a_1}(a) & \dots & \frac{\partial f_1}{\partial a_n}(a) \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial a_1}(a) & \dots & \frac{\partial f_m}{\partial a_n}(a) \end{pmatrix}. \tag{5.21}$$

As regards $F : \mathbb{R}^n \to \mathbb{R}$, it follows from the first formulation in Equation 5.13 that

$$\frac{\partial F}{a_j}(a) = \sum_{i=1}^{m} f_i(a) \frac{\partial f_i}{a_j}(a). \tag{5.22}$$

Thus, the gradient Equation 5.15 is

$$F'(a) = J(a)^T f(a). \tag{5.23}$$

We shall also need the Hessian of F. From Equation 5.22 we see that the element in position (j,k) is

$$\frac{\partial^2 F}{\partial a_j \partial a_k}(a) = \sum_{i=1}^{m} \left(\frac{\partial f_i}{\partial a_j}(a) \frac{\partial f_i}{\partial a_k}(a) + f_i(a) \frac{\partial^2 f_i}{\partial a_j \partial a_k}(a) \right),$$

showing that

$$F''(a) = J(a)^T J(a) + \sum_{i=1}^{m} f_i(a) f_i''(a). \tag{5.24}$$

5.3.2 Gauss-Newton algorithm

The Gauss-Newton algorithm uses the first derivatives of the components of the vector function f to determine the minimizer a^*. It is based on a linear approximation to the components of f in the neighborhood of a. For small $\|h\|$ we see from the Taylor expansion Equation 5.20 that

$$f(a+h) \simeq l(a) \equiv f(a) + J(a)h. \tag{5.25}$$

Inserting this in the definition Equation 5.13 of F we see that

$$\begin{aligned} F(a+h) \simeq L(h) &\equiv \frac{1}{2} l(h)^T l(h) \\ &= \frac{1}{2} f^T f + h^T J^T f + \frac{1}{2} h^T J^T J h \\ &= F(a) + h^T J^T f + \frac{1}{2} h^T J^T J h \end{aligned} \tag{5.26}$$

(with $f = f(a)$ and $J = J(a)$). The Gauss-Newton step h_{gn} minimizes $L(h)$,

$$h_{gn} = \arg\min_h \{L(h)\}.$$

It follows that the gradient and the Hessian of L are, respectively,

$$L'(h) = J^T f + J^T J h, \qquad L''(h) = J^T J. \tag{5.27}$$

Comparison to Equation 5.23 shows that $L'(0) = F'(a)$. Further, we see that the matrix $L''(h)$ is independent of h. It is symmetric and if J has full rank, i.e., if the columns are linearly independent, then $L''(h)$ is also positive definite. This implies that $L(h)$ has a unique minimizer, which can be found by solving

$$(J^T J) h_{gn} = -J^T f. \tag{5.28}$$

This is a descent direction for F since

$$h_{gn}^T F'(a) = h_{gn}^T (J^T f) = -h_{gn}^T (J^T J) h_{gn} < 0.$$

5.3.3 Levenberg-Marquardt algorithm

Levenberg [Levenberg, 1944] and later Marquardt [Marquardt, 1963] suggested to use a *damped Gauss-Newton method*. In a damped method, the step h_{dm} is determined as

$$h = h_{dm} \equiv \arg\min_h \{L(h) + \frac{1}{2}\mu h^T h\}, \tag{5.29}$$

with the *damping parameter* $\mu \geq 0$. The term $\frac{1}{2}\mu\|h\|^2$ is introduced to penalize large steps. The step h_{lm} is defined by the following modification to Equation 5.28,

$$(J^T J + \mu I) h_{lm} = -g \text{ with } g = J^T f \text{ and } \mu \geq 0. \tag{5.30}$$

Here, $J = J(a)$ and $f = f(a)$. The damping parameter μ has several effects

- For all $\mu > 0$ the coefficient matrix is positive definite, and this ensures that h_{lm} is a descent direction, since

$$h_{lm}^T F'(a) = h_{lm}^T (J^T f) = -h_{lm}^T (J^T J) h_{lm} < 0.$$

- If the current iterate is far away from the solution and μ is large and we get a short step in the steepest descent direction

$$h_{lm} \simeq \frac{1}{\mu} g = \frac{1}{\mu} F'(x).$$

- If the current iterate is close to the solution and μ is very small, then $h_{lm} \simeq h_{gn}$.

Thus, the damping parameter influences both the direction and the size of the step.
The stopping criteria for the algorithm should incorporate that at a global minimizer we have $F'(a^*) = g(a^*) = 0$, so we can use

$$\|g\|_\infty \leq \varepsilon_1 \tag{5.31}$$

where ε_1 is a small, positive number, chosen by the user. Another relevant criterion is to stop if the change in a is small,

$$\|a_{new} - a\| \leq \varepsilon_2 (\|a\| + \varepsilon_2). \tag{5.32}$$

This expression gives a gradual change from relative step size ε_2 when $\|a\|$ is large to absolute step size ε_2^2 if a is close to 0. As in all iterative processes, we also need a safeguard against an infinite loop,

$$k \leq k_{max}. \tag{5.33}$$

Both, ε_2 and k_{max} are chosen by the user.

Chapter 6

Introduction to peak picking

Over the last decade mass spectrometry has become a prominent technique in the field of proteomic research. It allows for the large-scale characterization of hundreds to thousands of proteins in complex biological samples by the resolution of proteins or peptides with respect to their m/z values. Regardless of the MS-based experimental procedure, we are interested in those m/z values that correspond to measurements of proteins or peptides. In most cases, only small parts of a full mass spectrum represent the interesting signal. To decrease the amount of data and allow for further analysis steps, we need methods that extract the information we are interested in from the mass spectrum.

Subject to the MS-based experimental procedure, different aspects of the signal can be of interest. MALDI-TOF instruments are often used for the identification of proteins, whereby the record of m/z values of the detected peptides in the mass spectrum serves as a peptide mass fingerprint. This pattern is usually distinctive and characteristic for the excised protein and used to identify the protein from a sequence data base. The more accurate the m/z values in the pattern are, the lower is the number of possible protein candidates and the more reliable is the identification result. The identification of proteins using tandem mass spectrometry is also subject to the determination of accurate m/z values for the parent ion as well as the fragment ions in the tandem spectra. However, mass spectrometric experiments that compare changes of perturbations in the proteomes of distinct samples depend on the accurate quantification of the proteins in the measurement. Therefore, the total ion counts of the detected peptides in the mass spectra have to be determined precisely.
Another important application of MS is the field of clinical proteomics. To discover potential biomarkers, differentially expressed proteins in different SELDI-TOF mass spectra are detected. Especially low abundant proteins may play an important role and thereby their m/z values and ion counts should be carefully extracted.
A general approach, which extracts all the mentioned characteristics of the interesting signal,

e.g., accurate m/z positions along with the respective ion count, without any loss of information, would facilitate any of the proposed analytical aims.

The following section briefly introduces the reader into the nature of mass spectrometric data and the aspect of the interesting signal in proteomics MS measurements. It summarizes expertise from [Henderson and McIndoe, 2005; de Hoffmann et al., 2001; Smith, 2005; Jurisica and Wigle, 2005; Hilario et al., 2006] as well as from lectures given by Knut Reinert and Clemens Gröpl in 2006 at the Freie Universität Berlin.

6.1 Nature of mass spectrometric measurements

As described in Section 3.1, a mass spectrum is produced by the three components of an MS system. The ion source that produces the protein or peptide ions represents the first component. The mass analyzer constitutes the second component separating the ions with respect to some unique properties, which result from the imposition of an electric or magnetic field. The values of the instrument variables imply certain m/z values. The ion detector, which records the ion currently generated by the ions emanating from the mass analyzer, represents the third component of an MS system.

An ideal mass analyzer would be able to distinguish ions even with slightly different m/z values, but as in all physical experiments, a mass spectrum is afflicted with uncertainties resulting from random fluctuations in measurement. Ions that have the same m/z value do not necessarily strike the detector at the same precise instant, because ions having the same m/z value have a small range of initial energies as they leave the ion source and thus are not expected to reach the analyzer and detector at exactly the same time.
Another reason for the blur of an m/z measurement is the imprecision of the mass analyzer. The instrument variables of a mass analyzer might not always express the m/z value of an ion precisely and therefore, not all ions will pass off the analyzer, when the value of the appropriate instrument variable corresponds to the correct m/z value [Smith, 2005].
The measurement of several ions with identical m/z values yields a peak shape in the mass spectrum that is centered around the real m/z value of the ions. This pattern is called a mass spectral peak:

Definition 6.1.1: A *mass spectral peak* is a localized maximum signal produced by the detector, which represents the ions of some chemical entity.

Figure 6.1 shows the mass spectra of the ions $[C_5H_6O_4]^+$,$[C_6H_{10}O_3]^+$ and $[C_9H_{22}]^+$ resulting from mass analyzers with different m/z separation capabilities. A commonly used term for the separation capability is *resolution*. A low resolution analyzer (e.g., quadrupole/ion trap in

low resolution mode or linear TOF) cannot discriminate the three ions and just a single peak is observed in the mass spectrum. At slightly higher resolution (e.g., quadrupole/ion trap in maximum resolution mode) the higher m/z ion is differentiated, but the remaining two ions appear just as a single peak at an m/z value intermediate between the two real values. Three peaks can be clearly observed at a resolution of 5000 (e.g., reflectron TOF), and the signals are baseline resolved at 10000 resolution (e.g., high performance reflectron TOF, FTICR).

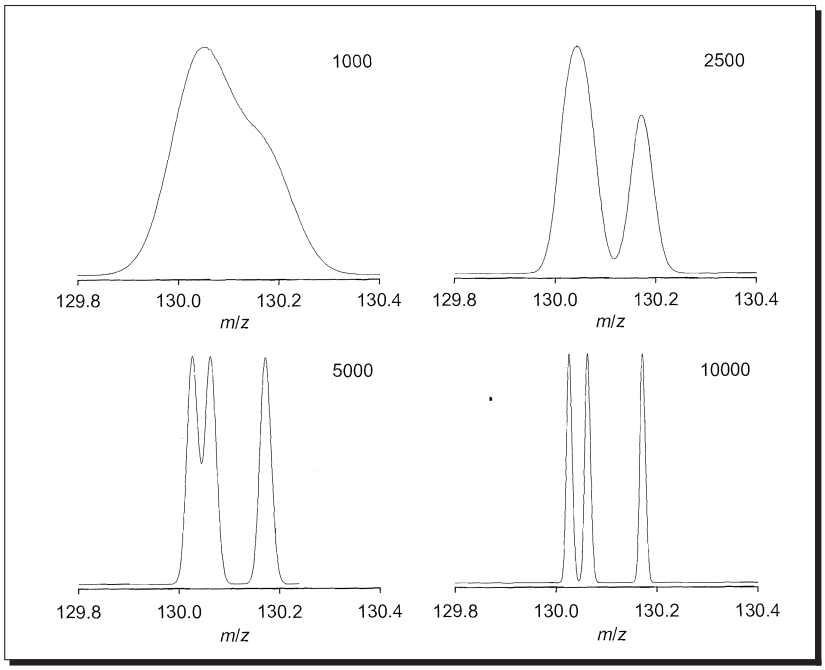

Figure 6.1: *Effect of increasing resolution in differentiating the ions* $[C_5H_6O_4]^+$,$[C_6H_{10}O_3]^+$ *and* $[C_9H_{22}]^+$. *The monoisotopic masses are* 130.0266, 130.0630, 140.1722 *m/z respectively (figure taken from Henderson and McIndoe [2005]).*

Figure 6.1 shows that the correspondence between peaks in the spectrum and the ions formed by the component is only one to one if the mass spectrometer is able to resolve the ions of different components. The higher the resolution, the narrower the peaks, and the better they are separated in the mass spectrum. One common definition of resolution, which is also used in Figure 6.1, is defined with respect to the full width of the peak at the half maximum intensity (FWHM):

Definition 6.1.2: The resolution R_{FWHM} is defined by

$$R_{FWHM} = \frac{m}{\Delta m},$$

whereby m is the maximum m/z position of the mass spectral peak and Δm its FWHM value.

Figure 6.2 illustrates the FWHM of a mass spectral peak.

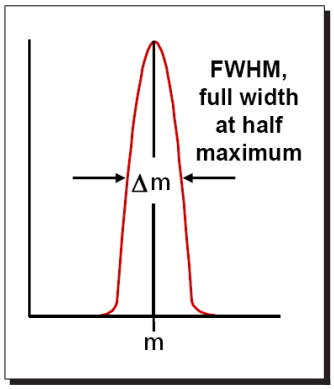

Figure 6.2: *FWHM of a mass spectral peak.*

With a sufficient resolution the ions presenting proteins or peptide components in a sample are not only represented by one peak in a mass spectrum, but instead by a number of so-called *isotopic peaks*. The proteinogenic amino acids consist of a combination of five elements: $C, H, S, O,$ and N. For all of them there exist different isotopes. Isotopes are atoms of the same element that differ in mass as they have different numbers of neutrons while containing the same number of protons and electrons. In addition to the isotope ^{12}C, carbon also has the ^{13}C isotope, hydrogen occurs in the isotopes ^{1}H and ^{2}H, and nitrogen in ^{14}N and ^{15}N, respectively. Oxygen has three isotopes $^{16}O, ^{17}O, ^{17}O$ and sulfur even four: $^{32}S, ^{33}S, ^{34}S, ^{35}S, ^{36}S$.
The *monoisotopic mass* of typical organic compounds is the sum of the masses of the atoms in a molecule using the lightest isotope mass of each atom. The existence of a corresponding monoisotopic peak in the spectrum depends again on the resolution of the MS system.

Figure 6.3 illustrates how the aspect of an isotopic envelope varies with increasing resolution. The figure shows the theoretical, isotopic pattern of doubly charged bombesin ions. This peptide is composed of 14 amino acids and has a monoisotopic mass of 1637.8329 Th. Accordingly, the monoisotopic mass of a doubly charged peptide ion is 819.4201 Th. An instrument resolution of 1000 (which corresponds to an FWHM of the mass spectral peaks of approximately 0.8 Th) does not provide the differentiation of individual isotopic peaks. However, a mass spectrometer with a resolution of 2500 is able to separate the isotopic peaks (FWHM $\approx$ 0.33 Th) and allows for an estimate of the monoisotopic mass with respect to the monoisotopic peak. But the isotopic peaks in this spectrum still slightly overlap. In case of a resolution of 5000 the isotopic peaks (FWHM $\approx$ 0.16 Th) in the mass spectrum are baseline-resolved, that is to say ions of the individual isotopes are clearly discriminated and result in three non-overlapping mass spectral peaks. The precise separation capability of a mass spectrometer with resolution 10000 produces three narrow baseline-resolved peaks (FWHM $\approx$ 0.08 Th).

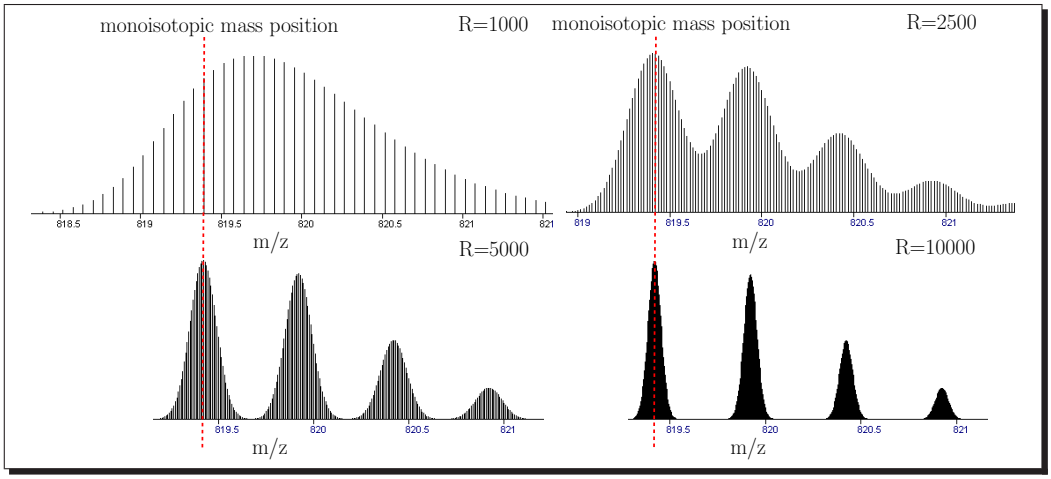

Figure 6.3: *Effect of increasing resolution in differentiating the isotopic peaks of bombesin (amino acid sequence: Gln-Gln-Arg-Leu-Gly-Asn-Gln-Trp-Ala-Val-Gly-His-Leu-Met-NH_2; UniProt entry P84214). The theoretical monoisotopic mass of the doubly charged peptide is* 819.4201 *Th. The isotopic envelope is simulated using the tool Isotopica [de Cossio et al., 2004].*

Horn et al. [2000] observed that the distance between isotopic peaks is $\frac{1.00235}{z}$ Th measuring peptide ions with charge z. We call the uniform spacing of isotopic peaks the *peptide mass rule*.

A monoisotopic mass spectrum is defined as a list of the monoisotopic m/z values extracted from the original raw mass spectrum. The mass spectral peak representing the monoisotopic mass is not always the most abundant isotopic peak in a spectrum despite it containing the most abundant isotope for each atom. This is due to the fact that as the number of atoms in a molecule increases the probability that the entire molecule contains at least one heavy isotope increases as well. For example, if there are 100 carbon atoms in a molecule—whereas each of them has an approximately 1% chance of being a heavy isotope—then the whole molecule is most likely to contain at least one heavy isotope.

As we have exemplarily seen on the basis of bombesin in Figure 6.3, depending on the resolution of the mass analyzer the mass spectral peaks either represent the measurement of multiple isotopic ions of a peptide, or the measurement of the individual isotopic ions. Accordingly, the apex of a peak either belongs to a more or less precise measurement of the m/z value of the isotope ions, or to an estimate of the average isotopic mass. However, the average atomic mass of an element is defined as the weighted average of the masses of all its naturally occurring stable isotopes. In Figure 6.3 the apex of the peak resulting from a resolution $R_{FWHM} = 1000$ corresponds approximately to the average mass of 819.96 Th, whereby the apex positions of the peaks measured with resolution $R_{FWHM} = 10000$ are quite good estimates of the theoretical isotopic m/z values and they precisely follow the peptide mass rule.

As we have seen, the interesting signal in a mass spectrum is represented by mass spectral peaks. Unfortunately only a small fraction of all maxima in a mass spectrum belongs to measurements of peptide or protein ions, all others are caused by noise. In mass spectrometric measurements we have to distinguish two types of noise, chemical and random noise.

Chemical or colored noise is a significant source of background interference in ESI mass spectra. This chemical noise is a fixed pattern noise, which manifests itself at specific m/z ratios. It results from the mass analysis of charged species other than the analyte compound. Interferences are either ions or salt adducts in the electrospray solution, species generated electrochemically, or neutral species present in the atmosphere around the ESI spray that are charged in the gas phase by proton transfer. If ESI is coupled by means of liquid chromatography (LC), chemical noise can be very abundant at the beginning and at the end of the elution process. In MALDI-MS, chemical noise is mainly produced by clusters of matrix molecules that are abundant in the sample mixture.

Random, electronic, or white noise is any source of undesired interference whose time of occurrence is not correlated with the signal and reveals some sort of background noise at virtually every m/z value. It is assumed that it arises primarily from electronic noise in the detector of the MS instrument.

Both types of noise may mask or mimic the interesting signal, where the chemical noise represents the harder problem, because it has a pattern in the m/z domain similar to that of the signal. In most cases mass spectra are not only disturbed by noise, but also by the so called *baseline*. In MALDI spectra, chemical noise can be very abundant in the lower mass range causing a strong upward drift in the baseline of the mass spectra, which falls off rapidly with increasing mass. In ESI spectra, chemical noise can form a bump in the baseline in the intermediate mass range. Figure 6.4 illustrates the additive composition of a mass spectrum by mass spectral peaks, baseline, and noise.

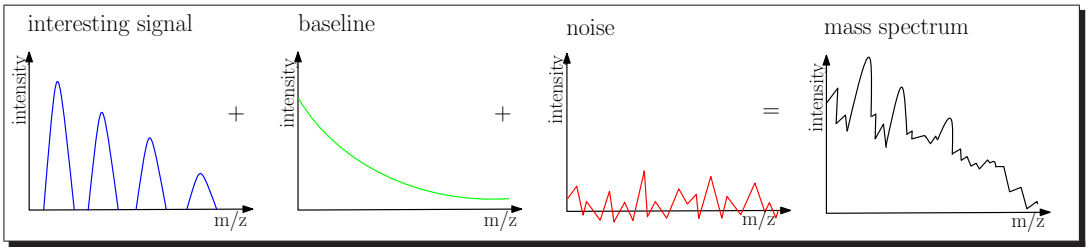

Figure 6.4: *Mass spectral peaks of the interesting chemical entities are afflicted by baseline and noise signal.*

6.2 Peak picking problem

The previous section describes the nature of mass spectrometric measurements and we have seen that the mass spectral peaks represent the interesting information. In identification experiments using PMF or tandem MS, their m/z positions can be used to identify the proteins or peptides in a sample. In clinical proteomic experiments the m/z positions of the protein peaks can be used to assign corresponding peaks in multiple spectra and to derive a proteomic fingerprint of multiple samples. However, quantitative LC-MS-based applications use either the peak area (summed over the elution time of the component) or the maximum peak height in ion counts [Bondarenko et al., 2002; Wang et al., 2003; Schulz-Trieglaff et al., 2007; Old et al., 2005] to yield relative or absolute estimates of the peptide or protein concentrations in a sample.

Accordingly, an algorithm that facilitates all the different analysis goals should determine the *accurate peak positions*, their *maximum intensities*, as well as the *total ion counts* represented by the peaks. Furthermore, it should estimate the *FWHM values* of the peaks that are associated with the resolution of the mass analyzer. Figure 6.5 illustrates the four important features of a mass spectral peak.
We call an algorithm that determines the peak features of interest a *peak picking algorithm* and define the peak picking problem as

Peak Picking Problem:
Given k raw mass spectra $k \in \mathbb{N}^+$.
Find the accurate positions, heights, total ion counts, and FWHM values of all mass spectral peaks in the presence of noise and baseline artifacts.

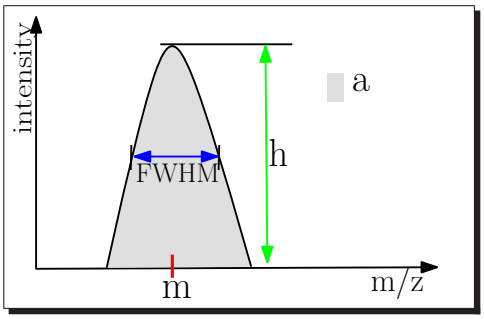

Figure 6.5: *Important features of a mass spectral peak: position m, height (maximum intensity) h, full width at* 50% *height (FWHM), and the total ion count a.*

As mentioned in the previous section, the signal may be masked or mimicked by uncertainties in the measurement (see Figure 6.4).

The two sources of interference, baseline and noise, disturb the interesting mass spectral peak features in different ways and any peak picking algorithm should overcome these difficulties. Random noise, which is not correlated with the signal, reveals some sort of background noise at virtually every m/z value. It is represented by narrow bumps in the mass spectrum, which can be easily distinguished from real mass spectrometric peaks. Since noise superimposes on the mass spectrometric peaks, it may shift the "true" peak positions and may also tamper with the peak heights. Only the total ion count should remain more or less unaffected, because the mean of white noise is zero. Isotopic peaks are hard to distinguish from chemical noise, because it has a pattern in the m/z domain similar to that of the signal. Colored noise peaks that are not removed from the spectrum can lead to false positive and negative identifications. In mass spectrometry, as in all physical experiments, errors and uncertainties result not only from random fluctuations in measurement, but also from systematic errors. Systematic errors in mass spectrometry are caused by a poor calibration and result in a high loss of accuracy. Particularly MS-based identification experiments using TOF analyzers depend on the proper correction of these calibration errors. Calibration algorithms are covered by a separate research area, which is not subject of this thesis and are handled elsewhere [Strittmatter et al., 2003; Gobom et al., 2002; Tan and Brown, 2002; Wolski et al., 2005].

Besides noise and baseline, a peak picking algorithm is faced with two more problems. The first problem is given by the *overlap of mass spectral peaks*. Peaks may be convoluted due to two reasons: 1) a poor resolution of the mass analyzer, or 2) highly charged protein or peptide ions. An example of a poor resolution is shown in Figure 6.6. The isotopic peaks of the doubly charged bombesin ions strongly overlap with a resolution of $R_{FWHM} \approx 2300$ around m/z 810 (LC-ESI-ion trap mass spectrometry).

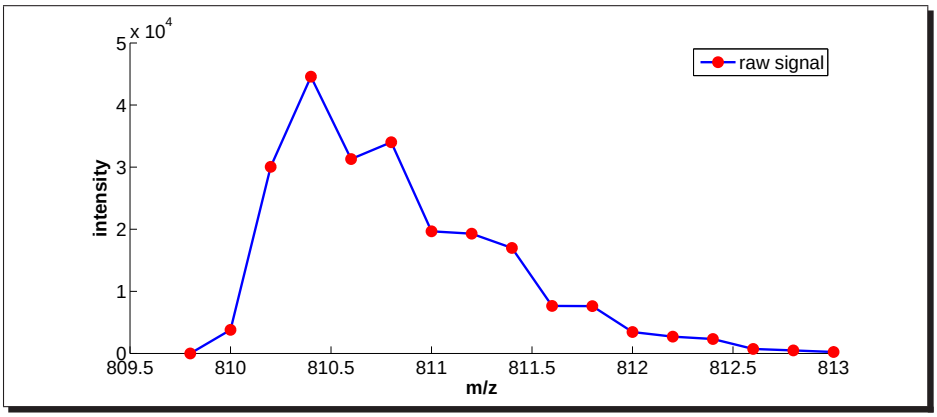

Figure 6.6: *Raw mass spectrum of doubly charged bombesin ions (p-Glu-Gln-Arg-Leu-Gly-Asn-Gln-Trp-Ala-Val-Gly-His-Leu-Met-NH_2). The raw data points (circles) are linearly interpolated. The poor resolution of the ion trap analyzer yields in a strong overlap of the four isotopic peaks.*

The capability of a peak picking algorithm to separate overlapping isotopic patterns is very important for LC-MS quantification experiments. In a quantification pipeline the peak picking algorithm is directly followed by the so-called *feature finding* process collecting all isotopic peaks of a peptide and combining them to a feature. To enable a reliable charge prediction with respect to the isotopic pattern, the isotopic peaks have to be discriminated and their positions have to be accurately determined.
Besides the decrease in resolution, an increase in charge state will also result in convolved isotopic patterns, because the higher the charge state, the smaller the distance gets between the isotopic peaks according to the peptide mass rule.

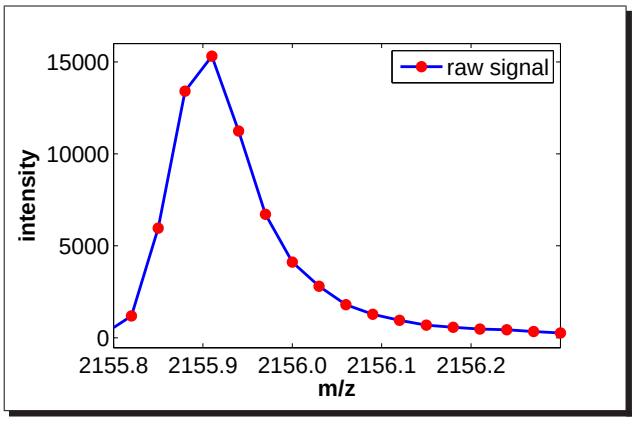

Figure 6.7: *Asymmetric peak in a mass spectrum measured with an MALDI-TOF instrument. The raw data points (circles) are linearly interpolated.*

The second difficulty a peak picking algorithm has to overcome is a considerable *asymmetry of mass spectral peaks*. Imperfections in the mass analyzer often add up to a peak skewness. For example, in quadrupole mass spectrometers asymmetry results from hyperbolic or circular section electrodes and may be increased by manufacturing imperfections and is also affected by fringe fields at the ion entrance and exit positions. Features of the ion source may also affect the peak shape [Gibson and Taylor, 2003]. Kempka et al. [2004] state that the geometrical position where the ions are produced in the ion source, as well as the initial velocity of the ions, will affect their flight time and hence, the shapes of peaks in the resulting mass spectra. Figure 6.7 shows an isotopic peak in a MALDI-TOF mass spectrum.

This asymmetry has to be considered in any peak picking approach, because it hampers a correct mass to charge determination.

Chapter 7

Related work

The entry of mass spectrometry into the analytical biotechnology in the 1990s as a technique for the identification and quantification of proteins and peptides was accompanied by the development of algorithms to process the resulting data. Many peak picking algorithms were established, which determine, tailored to the objective of the mass spectrometric measurement, important information about mass spectral peaks, such as accurate monoisotopic peak positions, centroid positions of resolved isotopic peaks, and the ion counts of mass spectral peaks (height or area under the curve). Most of the peak picking algorithms are designed for a specific instrument type or a particular application and cannot be used to extract all interesting information from a mass spectrum.

A lot of peak picking algorithms were designed to enable the accurate identification and characterization of proteins using peptide mass fingerprints in MALDI-TOF spectra [Breen et al., 2000; Wehofsky and Hoffmann, 2001; Kempka et al., 2004; Samuelsson et al., 2004; Gras et al., 1999; Berndt et al., 1999]. Some other peak picking algorithms were developed with respect to biomarker discovery in MALDI-TOF or SELDI-TOF spectra [Yasui et al., 2003; Randolph and Yasui, 2006; Tibshirani et al., 2004; Coombes et al., 2005; Yu et al., 2006; Mantini et al., 2007; Du et al., 2006]. The remaining peak picking algorithms are related to the general analysis of MS or LC-MS proteomics data [Wehofsky and Hoffmann, 2002; Horn et al., 2000; Strittmatter et al., 2003; Katajamaa et al., 2005; Bellew et al., 2006; Li et al., 2005; Andreev et al., 2003].

Most publications propose methods beyond a peak detection algorithm. In particular, these are methods

- for baseline and noise correction [Breen et al., 2000; Samuelsson et al., 2004; Berndt et al., 1999; Coombes et al., 2005; Yu et al., 2006; Mantini et al., 2007; Katajamaa et al., 2005; Bellew et al., 2006; Li et al., 2005; Andreev et al., 2003],

- a calibration method [Yasui et al., 2003; Kempka et al., 2004; Samuelsson et al., 2004; Bellew et al., 2006; Strittmatter et al., 2003; Gras et al., 1999],
- a deconvolution algorithm [Wehofsky and Hoffmann, 2002; Horn et al., 2000; Bellew et al., 2006; Li et al., 2005],
- a deisotoping algorithm [Wehofsky and Hoffmann, 2001, 2002; Horn et al., 2000; Breen et al., 2000; Samuelsson et al., 2004; Gras et al., 1999; Berndt et al., 1999; Li et al., 2005],
- a method for the alignment of peak lists [Randolph and Yasui, 2006; Tibshirani et al., 2004],
- an algorithm for peptide mass fingerprinting [Samuelsson et al., 2004; Gras et al., 1999; Berndt et al., 1999],
- or a pattern classification method [Tibshirani et al., 2004].

The existing peak picking algorithms can not only be classified by the MS-based application or instrument type they are developed for, but also by their way of detecting the mass spectrometric peaks in spectra.
As mentioned in Section 6.1 (see Figure 6.4), mass spectra are composed of three different terms, which are a high-frequency noise term, a low-frequency baseline or background term, plus the information we are interested in that occupies a frequency range in between noise and baseline [Tan and Brown, 2002].
Most of the proposed peak picking algorithms successively correct noise and baseline in a mass spectrum [Breen et al., 2000; Berndt et al., 1999; Samuelsson et al., 2004; Tibshirani et al., 2004; Mantini et al., 2007; Katajamaa et al., 2005; Andreev et al., 2003; Gras et al., 1999], or try to detect the peaks directly in the unprocessed raw mass spectrum [Wehofsky and Hoffmann, 2001; Yasui et al., 2003; Kempka et al., 2004; Wehofsky and Hoffmann, 2002; Horn et al., 2000; Strittmatter et al., 2003].

Several groups [Yasui et al., 2003; Tibshirani et al., 2004; Mantini et al., 2007] use a very simple peak picking strategy that searches for local maxima in SELDI-TOF and MALDI-TOF spectra. All data points that have the highest intensity among a certain number of neighboring data points are defined as “protein intensity peaks”, and extracted from the spectrum. Prior to peak detection, several groups [Mantini et al., 2007; Tibshirani et al., 2004] filter the noise in the mass spectra, using a “loess” or a low-pass Kaiser filter. Subsequent to the smoothing process, Mantini et al. [2007] estimate a baseline and noise level with respect to the kurtosis of the data and filter out peaks with a low signal-to-noise value. This simple peak detection method is not able to distinguish the mass spectral peaks of interest from chemical noise peaks, because it does not incorporate the width of mass spectral peaks.

Strittmatter et al. [2003] use a fit of a Gaussian mixture to model the observed asymmetry of peak shapes in LC-ESI-TOF mass spectra. Two Gaussian distributions are used to estimate the m/z value of each peak in the mass spectrum. The second peak is used to fit the tailing effect at the high-mass end of the peak distribution, whereas the midpoint of the first Gaussian function represents the m/z position of the mass spectral peak. In connection with a calibration method for LC-ESI-TOF machines (which should be transferable to other instrumentation, such as FT-ICR and ion trap instruments), they achieve a considerable improvement in mass accuracy for non-convoluted LC-ESI-TOF data. Kempka et al. [2004] elaborate on this mixture modeling and also test other mixtures such as a Lorentzian and a Gaussian curve. They accomplish the fit of two Gaussian distributions in the time-of-flight dimension and use the midpoint of the first Gaussian function as the flight-time of the peak distribution. Afterward some of the determined flight times were used to estimate the coefficients in a fourth-order polynomial function, which provides the relationship between known m/z values and the picked flight times [Gobom et al., 2002]. They compare their results to those obtained by commercial peak picking algorithms (SNAP) and conclude that they perform better for most peaks. For small and considerably skewed peaks, the improvement in accuracy is up to fivefold. Strittmatter et al. [2003] and Kempka et al. [2004] have shown how important it is to consider the skewness of peaks during peak picking, but the improvement in mass accuracy is only shown by Kempka et al. using MALDI-TOF data without convoluted peaks, which are baseline or close to baseline separated. Furthermore, their peak picking approach detects the peaks in the time-of-flight dimension, which limits its application to TOF mass spectrometric data.

Gras et al. [1999] determined the monoisotopic peak positions in MALDI-TOF mass spectra. In a first step a noise and baseline level is estimated. Each data point that exceeds the noise level is used as a starting point for a fit of a normalized average isotopic pattern obtained by an in silico digestion of proteins in the SWISS-PROT database [Bairoch and Apweiler, 2000]. In the vicinity of the starting point, an error function is evaluated. The lowest minimum of the error function indicates the monoisotopic peak position. To enable the separation of overlapping isotopic pattern, the average peak distribution is subtracted from the spectrum and the monoisotopic peak finding process is iterated. Berndt et al. [1999] propose a similar approach, which differs only in the estimation of the baseline and the fitting method. They use a Levenberg-Marquardt algorithm to fit the average isotopic distribution to the data.
Breen et al. [2000] use a Poisson distribution to model the isotopic pattern instead of a sum of Gaussian functions. They accelerate the matching of the isotopic pattern to the data by an enhanced preprocessing. In a first step, they use mathematical morphology and watershed algorithms to extract the individual isotopic peaks in a mass spectrum. In a second step, they fit the Poisson model to the data to determine which peak in a group is the monoisotopic peak. Breen et al. prove the sensitivity of their peak picking method by comparing the automatically detected monoisotopic mass spectra with monoisotopic spectra that were manually determined.
Wehofsky and Hoffmann [2001] use a mass-dependent average isotopic pattern to deisotope

mass spectra of peptides, but the iterative process of fitting the theoretical pattern to the spectrum and subtracting it afterward remains the same as in the methods described above.
The peak extraction approach of Samuelsson et al. [2004] differs from the other monoisotopic peak picking algorithms. After baseline and noise estimation, similar to the process described in Breen et al. [2000], Samuelsson et al. initially extract isotopic peaks, defined by consecutive data points exceeding a certain signal-to-noise value. Afterward, consecutive peaks are grouped into clusters and a convex programming problem is formulated. The minimization procedure corresponds to the objective of determining the lowest number of peptides and their m/z values, which, given the measured peak intensities and the template isotope distributions, can account for the isotopic pattern of the cluster.

For the monoisotopic peak detection in ESI mass spectra several groups [Horn et al., 2000; Wehofsky and Hoffmann, 2002] adapted the deisotoping approach for MALDI spectra data by a charge deconvolution. The deisotoping methods are very similar to the methods presented for MALDI mass spectra and use the fit of an average isotopic pattern. The successive fit of theoretical isotopic pattern to the raw spectrum leads to a high runtime of the proposed monoisotopic peak picking methods. Except of Breen et al. [2000], the monoisotopic peak picking algorithms do not make use of an enhanced preprocessing and work directly on the raw spectra.

Andreev et al. [2003] developed a further peak picking algorithm, which uses the 2D structure of peaks corresponding to a sample component in LC-MS data. They use a matched filter to minimize chemical as well as random noise. The matched filter is the optimal linear filter for maximizing the signal-to-noise ratio in the presence of additive noise. A matched filter uses the peak and noise characteristics to detect interesting signal in the data. Andreev et al. estimate the noise characteristics in “vacant” EIC, assumes the chromatographic peak shape to be Gaussian, and uses this information to obtain a properly matched filter.
After filtering each of the EICs using the matched filter, the actual peak picking is performed, based on comparison of scores generated for each peak candidate with a certain threshold. In a first step, a score for each EIC is computed, which indicates the presence or absence of peaks in the chromatogram. The peaks of EICs that have a score greater than a certain threshold represent peak candidates. To examine the peak shape in m/z dimension, a score is computed, based on the comparison of the intensity at the peak apex position with the intensities of the neighboring m/z values.
As a final step, the monoisotopic peaks were selected from the isotopic clusters and then peaks corresponding to sodium and potassium adducts were determined and eliminated from the peak list. The scoring rules include several parameters which are determined by trial and error, but they plan to apply machine learning algorithms and large training data sets in order to determine the optimum values of both the score parameters as well as the threshold.

The two simple peak picking strategies of Katajamaa et al. [2005] are implemented in a soft-

ware package for the analysis of LC-MS data. The first strategy, which is recommended for data already picked by the instrument, searches for all local maxima in the mass spectra that exceed a certain threshold. The second recursive strategy additionally considers the width of the peaks at each local maximum.

A different class of peak picking approaches takes advantage of the local and multiscale properties of spectral signals by separating a signal into its individual frequency contributions using the wavelet transformation [Alsberg et al., 1997] or quadrature filters [Granlund and Knutsson, 1995]. Coombes et al. [2005]; Bellew et al. [2006]; Li et al. [2005] use the wavelet transform only for the noise correction of spectral data and search for peaks in the smoothed signal. Randolph and Yasui [2006]; Du et al. [2006] isolate the contributions of the analyte signal from background and noise in order to detect the peaks directly on the corresponding scales in the wavelet transform. Yu et al. [2006] use the logarithm transformation and a Gabor filter to detect isotopic patterns.

Randolph and Yasui [2006] propose a method that cannot be directly understood as a peak picking method. The interesting signal extracted by their approach does not stringently correspond to the positions of mass spectral peaks, but rather indicates interesting changes in the spectrum intensities. Randolph et al. decompose MALDI mass spectra into the sum of constituent functions, each containing a particular scale of the signal. At the j^{th} dyadic scale, the "scale-j detail function D_j" reflects the scale-based changes in a spectrum that occur across a 2^j-unit domain. The subsampling of the CWT at dyadic scales retaining all locations is called translation-invariant wavelet transform (TIWT). Randolph et al. locate so-called "scale-j features" defined as local maxima in D_j. The set of all local maxima in D_j does not correspond to the set of local maxima in the spectrum, but corresponds to local changes in the spectrum, of scale j, as extracted by D_j. The existence of a scale-j feature is not defined in terms of the intensity of the signal at that position. Rather it depends on a relative change in the intensity over a region whose width depends on the scale j. Hence, it indicates inflections or shoulders in the spectrum. The maxima in D_j are determined using wavelet families having one and two vanishing moments.
To extract interesting feature patterns from different MALDI spectra, they build histograms for the scale-j feature locations detected in the detail functions D_j of all spectra. They claim that most relevant features are described by a small subset of scales.

Coombes et al. [2005] use the translation-invariant undecimated discrete wavelet transform (UDTW) for noise filtering of SELDI spectra. Afterward, the baseline is removed and the peaks are detected via a local maximum search in the preprocessed data. The peak endpoints are defined by the adjacent local minima. Flat peaks as well as peaks with a too small width, are filtered out and peaks that lie to close together are combined.

Yu et al. [2006] developed an algorithm to extract isotopic patterns in poorly resolved MALDI spectra measured in linear mode. To reduce the dynamic range of the intensities of a spectrum,

they initially compute the logarithmic transform of the spectrum. Yu et al. utilize the constant distance of peaks isotopic patterns of charge one. After some preprocessing, including baseline correction and noise filtering, they apply a Gabor quadrature filter to detect the isotopic pattern in the data. The impulse response of the Gabor filter is defined by a harmonic function multiplied by a Gaussian distribution; consequently its frequency response happens to be a Gaussian bandpass filter. This filter will therefore respond to some frequency range in the signal. The Gabor filter is centered at a frequency corresponding to a wavelength of $\lambda = 1$ Th. Therefore, maxima in the transformed signal indicate possible location of an isotopic pattern in the spectrum. If the peaks in the quadrature filtered signal exceed a certain width, the maximum positions define the peak positions in the spectrum.

The peak picking algorithms of several groups [Li et al., 2005; Bellew et al., 2006] are implemented in the software packages *SpecArray* and *msInspect* for the analysis of LC-MS data. Li et al. [2005] use the TIWT to smooth each scan in an LC-MS raw map. Local maxima in the smoothed spectra that exceed a certain threshold define the mass spectral peaks. Bellew et al. [2006] suggest that they also use the wavelet transform to facilitate the peak picking process, but unfortunately neither in Bellew et al. [2006], nor in the user guide of msInspect they describe the peak picking procedure in more detail.

Du et al. [2006] propose a CWT-based approach for the detection of mass spectral peaks in SELDI-TOF mass spectra. Due to the varying peak width of protein peaks with respect to the m/z dimension, they search for the peaks in 33 scales of the wavelet transformed spectrum using a Mexican hat mother wavelet. Major peak locations correspond to ridges occurring on several successive scales. To detect the interesting peaks, they therefore locate all local maxima on each scale and link corresponding maxima of adjacent scans together to so-called “ridge lines”. Furthermore, they compute the signal-to-noise ratio of each maximum using the smallest scale for the estimation of the noise level. If the length of a ridge line exceeds a certain threshold and the scale of the maximum amplitude on a ridge line lies within a predefined scale range, and if furthermore the maximum amplitude exceeds a certain threshold it defines a peak. Their algorithm follows in its essentials our peak picking approach, whereas we provide a powerful peak picking approach without the costly determination of 33 wavelet scales. Furthermore, we extract the information of interest directly from the raw data and can therefore yield more accurate m/z positions and peak widths.

None of the proposed peak picking algorithms represents a general solution of the peak picking problem. All methods depend on a specific instrument type and provide only information for a certain analytical aim.

Chapter 8

Own contribution

We propose a wavelet-based peak picking technique suited for the application to the different kinds of mass spectrometric data arising in computational proteomics [Lange et al., 2006, 2005]. It solves the peak picking problem as defined in Section 6.2 and additionally extracts some useful information, which facilitates further analysis steps. The m/z values are accurately determined not only for well-resolved, but also for convoluted data using an asymmetric peak shape. It achieves this in real time and does not make assumptions about the underlying machine or ionization method (MALDI or ESI), which makes the algorithm robust for different experimental settings. In Chapter 9, we will show the performance of our peak picking algorithm on two different kinds of data: a low-resolution LC-ESI data set and high-resolution MALDI spectra. Compared to a vendor supplied standard algorithm, our algorithm delivers superior performance on the former and state of the art performance on the latter data set.
The independence of the underlying machine is achieved by addressing the problem from a signal-theoretic point of view, which tells us that spectral data such as MS measurements are of an inherently multiscale nature. Different effects, typically localized in different frequency ranges, add up to a result in the final signal (see Figure 6.4). As mentioned in Section 6.2, we will assume that the experimentally obtained signal s can be decomposed into three such contributions: a high-frequency noise term n, a low-frequency baseline or background term b, and the information i we are interested in, often referred to as the analytical signal [Tan and Brown, 2002], where i occupies a frequency range in between noise and baseline.

In Section 6.2, we described the peak picking problem and defined the characteristic features of mass spectral peaks. Compared to other approaches, our approach extracts additional information about a peak's shape, such that the fit of an average isotopic pattern to the peak data during the feature finding process is improved. In contrast to many established approaches to this problem, the algorithm presented here has been particularly designed to work well even on data of low resolution with strongly overlapping peaks. This is especially apparent when

separating, for example, charge two isotopic patterns with poor resolution, as the bombesin isotopic peaks in Figure 6.6. The peak picking approach directly exploits the multiscale nature of the measured mass spectrum. This becomes possible with the help of the Continuous Wavelet Transformation (CWT) (see Section 5.2). A main advantage of the CWT in contrast to other decomposition methods such as the Fourier transform, is the preservation of information about the localization of different frequencies in the signal in a near-optimal manner [Louis et al., 1997]. Using the CWT, we can split the signal into different frequency ranges or length scales that can be regarded independently of each other. This is demonstrated in Figure 8.1, where we have plotted the transformed signal of a typical region of a mass spectrum on different scales. Apparently, looking at the signal at the correct scale—in our case, a rough estimate of the typical peak width as depicted in panel B—effectively suppresses both baseline and noise, keeping only the contribution due to the analytical signal.

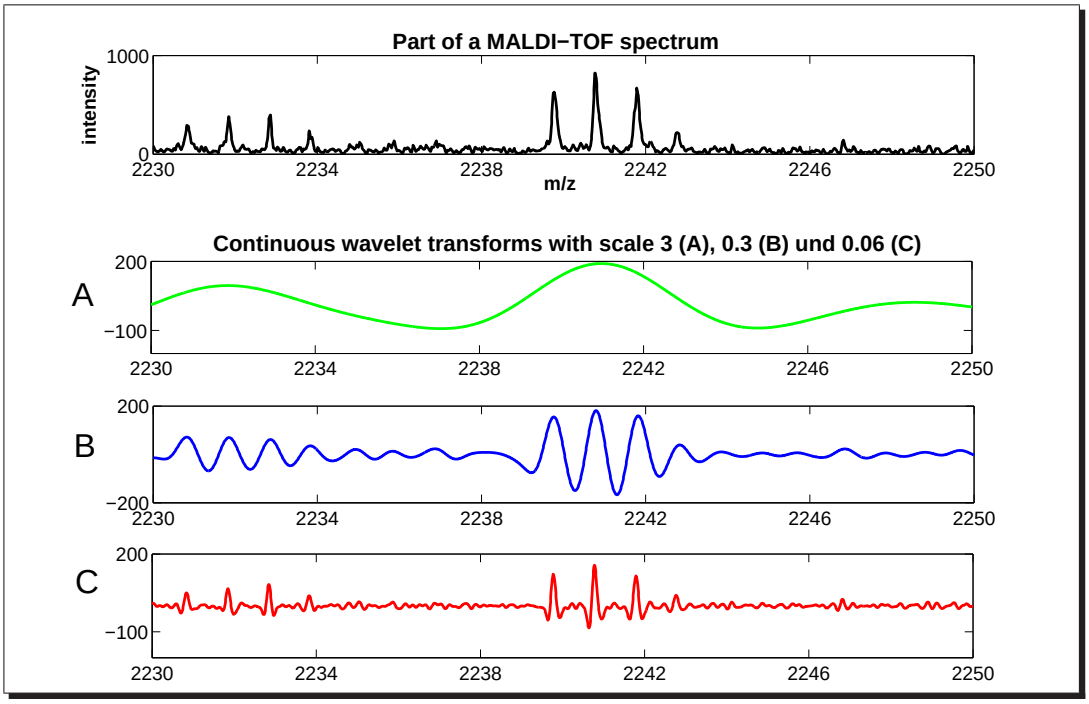

Figure 8.1: *The plot on top represents a mass interval of a MALDI-TOF spectrum between* 2230 *Th and* 2250 *Th. Plots A, B, and C show the continuous wavelet transform of the spectrum using a Mexican hat wavelet with fixed scale values a (A:* $a = 3$*, B:* $a = 0.3$*, C:* $a = 0.06$*).*

This decomposition allows us to determine each feature of a peak in the domain from which it can be computed best, i.e., either from the frequency range of the analytical signal *i*, the full signal *s*, or from a combination of both.

Our algorithm is a three-step technique that first determines the positions of putative peaks in

the wavelet-transformed signal and then fits a peak function to the original raw data in that region. In a third step, we use the CWT again to separate overlapping signals. For the optional fourth stage, we offer two techniques from non-linear optimization, which both improve the fit, either in a single mass spectrum, or in two-dimensional LC-MS data.

8.1 General schema of our peak picking algorithm

Our peak picking algorithm searches peaks in individual mass spectra, so in case of LC-MS data the mass spectral peaks are subsequently extracted from every scan. In a first step, the continuous wavelet transformation is computed of the whole scan. Starting from the maximum position in the wavelet transform, every peak centroid, its height, and its area can be estimated in the raw data. Using these parameters, we are able to represent the raw data peaks by typical analytical peak functions. We perform the fit of an asymmetric $sech^2$ and an asymmetric Lorentzian function.
Afterward, overlapping peaks may be separated by an efficient separation technique: in a first step, we estimate the number of convolved peaks in the continuous wavelet transform, and discriminate in the second step the peaks by a non-linear optimization technique.
At this stage of the algorithm, the fitted analytical description is typically in very good correspondence with the experimental signal. To further improve the quality of the fit, the correlation of the resulting peaks with the experimental data can be increased in a subsequent, optional optimization step. This is of particular importance in two cases: first, if neighboring peaks overlap strongly enough that they cannot be fitted well individually, and second, if the resolution of the experimental data is low.

The pseudocode of the algorithm is given in Figure 8.2. In the rest of this chapter we elaborate on the individual steps of peak detection, fitting of an analytical peak function, separation of overlapping peaks and the optional optimization of peak parameters.

8.2 Peak detection

Over the past decades the wavelet transformation has found a broad field of application, e.g., in signal processing, image processing, as well as in bioinformatics [Liò, 2003]. It is commonly used for denoising, baseline removal, and compression of chemometrics data [Alsberg et al., 1997]. Wavelets are used to transform the signal under investigation into another representation that presents the signal information in a more useful form. Mathematically speaking, the wavelet transform is a convolution of the wavelet function with the signal as shown in Equation 5.8. If the wavelet matches the shape of the signal well at a specific scale and location, then a large value of the transform is obtained. If, however, the wavelet and the signal do not

PEAK PICKING PHASE

Input: Raw data *experiment* consisting of one or several mass spectra
Output: A list *peak_list* of all mass spectral peaks picked in *experiment*

```
1:  // pick the peaks in each spectrum
2:  for all mass spectra s in experiment do
3:      w:=pretabulateWavelet()
4:      n := 0, peak_list:=[]
5:      repeat
6:          peak_number := 0
7:          cwt:= continuousWaveletTransformation(s)
8:          while getNextMaximumPosition(cwt, s, p̂) do
9:              h := intensity(p̂)
10:             // t_sne: threshold for signal to noise ratio, t_i: minimal height
11:             if (signalToNoise(p̂) < t_sne) ∨ (h < t_i) then
12:                 continue
13:             end if
14:             (x_l,x_r) := searchForPeakEndpoints(s, p̂)
15:             c := estimateCentroid(x_l, x_r)
16:             (A_l, A_r) := estimateTotalIntensity(x_l, x_r)
17:             (p, fwhm, asym, corr) := fitPeakShape(A_l, A_r, p̂, h)
18:             if ((corr > t_corr) ∧ (fwhm > t_fwhm)) then
19:                 push(p, peak_list)
20:                 peak_number := peak_number + 1
21:             end if
22:             removePeak(x_l, x_r, s)
23:         end while
24:     until peak_number = 0
25: end for
26: // optional separation of overlapping peaks
27: peak_list:=separateOverlappingPeaks(peak_list, experiment)
28: // optional improvement of the peak parameters by non-linear optimization
29: peak_list:=optimizeAllPeakParameters(peak_list, experiment)
```

Figure 8.2: *Pseudocode of the peak picking algorithm.*

correlate well then the transform value is small. The choice of wavelet depends on the type of signal that is being investigated. Short-duration (high-frequency) features are best investigated using narrow wavelets, while longer-lasting (low-frequency) features are more suited to wider wavelets. Changing the type of wavelet lets one zoom in on individual small-scale, high-frequency components or to pan out to pick up larger-scale, low-frequency components.

Wu et al. [2001] used the CWT with the *Mexican hat wavelet* as the analyzing function to separate overlapping voltammetric peaks in voltammetric spectra (voltammetry is an electroanalytical methods used in analytical chemistry that determines information about an analyte by measuring the current as the potential is varied). Figure 8.3 shows the Mexican hat wavelet with scaling a and translation b. The Mexican hat wavelet is also called *Marr wavelet* and defined as the negative of the second derivative of the Gaussian function

$$\psi(x) := (1-x^2)\exp\left(-\frac{x^2}{2}\right) = -\frac{d^2}{dx^2}\exp\left(-\frac{x^2}{2}\right). \tag{8.1}$$

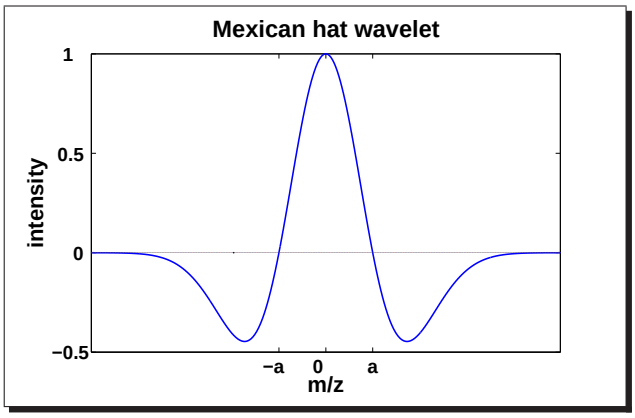

Figure 8.3: *The Mexican hat wavelet with scaling a and translation $b = 0$.*

Wu et al. chose the Mexican hat wavelet as defined in Equation 8.1 because of its simple symmetric form and the relation between voltammetric peaks and its wavelet transform. If the original peak can be described by a symmetric sech^2-function, Gaussian function, or Cauchy function, Wu et al. proved that the maximum position in the continuous wavelet transform (at a proper scale) corresponds to the maximum position of the original peak, and thus the peak positions can be located in the wavelet transform. To separate two overlapping peaks, they search for the first maximum in the wavelet transform. Assuming that the left half of the first peak is not interfered with by the second peak, it can be used to determine also its right half, which is covered by the left half of the second peak. Wu et al. use the symmetry of peaks and mirror the original signal at the maximum position of the first peak and subtract the reflected signal from the original signal. Thereby the contribution of the first peak on the signal

is removed and the second peak becomes visible in the iteration of the algorithm.
As we have seen in Section 6.1, the mass spectrometric peaks are asymmetric, but despite the skewness of the peaks, the maxima in the continuous wavelet transform at a proper scale correspond approximately to the peak positions in the original spectrum. Figure 8.1 shows that the mass spectral peaks can be detected in the wavelet transform at scale 0.06. As defined in Equation 5.7 in Section 5.2.3, the continuous wavelet transform of a signal $s \in L^2$ is defined as

$$W_s(a,b) = \langle s, \psi_{a,b} \rangle = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} s(x) \psi_{a,b}^* \left(\frac{t-b}{a} \right) dt \tag{8.2}$$

where * denotes complex conjugation. Using the Mexican hat wavelet defined in Equation 8.1 as a mother wavelet we get

$$W_s(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} s(x) \frac{d^2}{dt^2} \left(-\exp\left(-\frac{(t-a)^2}{2b^2} \right) \right) dt, \tag{8.3}$$

since ψ is a real-valued function and it holds $\psi^* = \psi$. The wavelet transform $W_s(a,b)$ as a function of b for fixed $a \neq 0$ can be interpreted as the "detail" contained in the signal at scale a, since we have seen in Equation 5.9 that the Fourier transform of a wavelet ψ is the transfer function of a bandpass filter and the convolution computes the wavelet transform with dilated bandpass filters.
Let us consider the wavelet transform $W_s(a,b)$ at a fixed scale $a \neq 0$ computed with the Mexican hat wavelet. Since the convolution and differentiation are linear systems [Smith, 1999] it holds with the scaling Theorem 5.2.1

$$s(x) \star \frac{d^2}{dt^2} \psi(x) = \frac{d^2}{dt^2} s(x) \star \psi(x) = \frac{d^2}{dt^2} (s(x) \star \psi(x)). \tag{8.4}$$

Hence, we may reorganize Equation 5.7 and get

$$W_s(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} -\frac{d^2}{dt^2} \left(s(x) \exp\left(-\frac{(t-a)^2}{2b^2} \right) \right) dt. \tag{8.5}$$

Therefore, the $W_s(a,b)$ is the second derivative of a "moving weighted average" of s performed with a translated and dilated Gaussian. The Gaussian filter extracts the frequency part of s with respect to a and the second derivative measures the concavity (second-order detail) of this "detail". If a function s is two-fold differentiable, a necessary condition for t_0 to be an extreme point is $s'(t_0) = 0$. If s also fulfills $s''(t_0) < 0$ then s has a local maximum at s_0. Accordingly, we can search for the local minima in the second derivative s'' to find the maxima in s. If we translate this fact into Equation 8.5, we can detect the mass spectral peaks in the signal by searching for local maxima in the negative second derivative of the interesting frequency part.

Figure 8.4 illustrates the procedure of peak detection. After a potential mass spectral peak is located in the continuous wavelet transform of the mass spectrum, the maximum position in $W_s(a,b)$ is used to find the peak's maximum position in the original spectrum. Using the

maximum position of a peak, its endpoints are determined. Furthermore, the centroid position and the height of the peak are estimated. In the following, this procedure will be described in more detail.

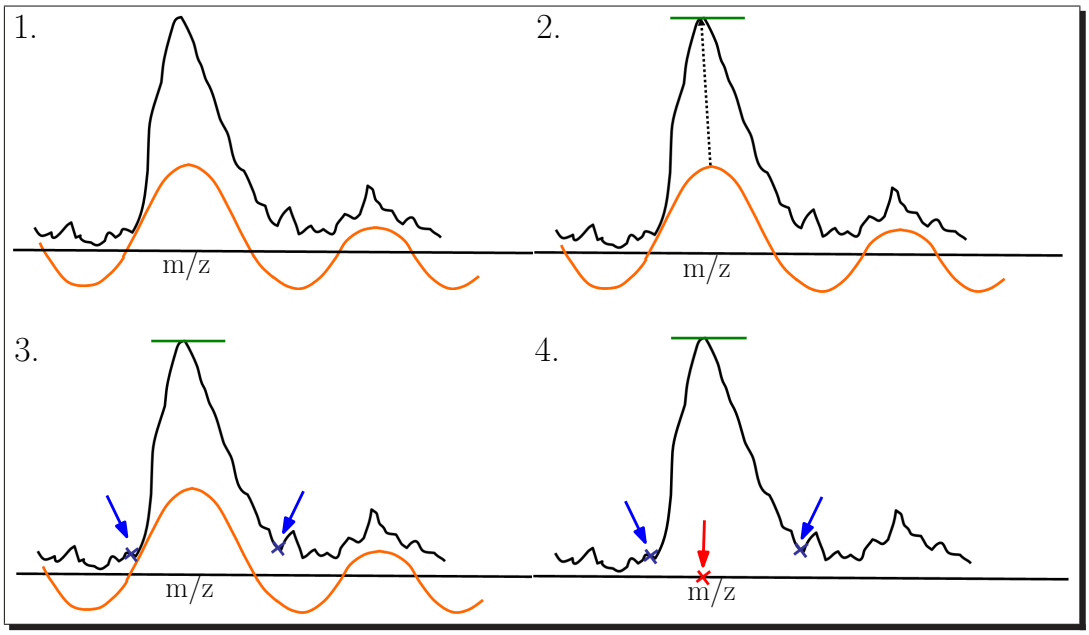

Figure 8.4: *Workflow of the peak detection. 1. Compute $W_s(a,b)$ with a fixed scale a and search for a maximum in $W_s(a,b)$, 2. Search for peak's maximum position, 3. Search for peak's endpoints, 4. Estimate the centroid.*

8.2.1 Detecting a peak in the continuous wavelet transform

Figures 8.1 and 8.5 show that we can detect the mass spectral peaks in $W_s(a,b)$ of the spectrum s using the Mexican hat wavelet ψ with a proper scaling factor a.

To extract the frequency range of mass spectral peaks, the scaling factor should correspond to a rough estimate of the typical peak width. With respect to the resolution of a instrument we can estimate a minimal mass spectral FWHM value $fwhm$. In the current version of the algorithm, we use $fwhm$ to determine the scale a. This works quite well if the mass range is relatively small ($500-1500$ Th) and the FWHM values of the peaks do not vary extremely between the peaks at low m/z values and the peaks at high m/z values. For greater mass ranges, the scale a should be adapted to FWHM values that grow with increasing m/z values.
We chose the scale parameter a of the wavelet ψ such that the FWHM value of the wavelet is

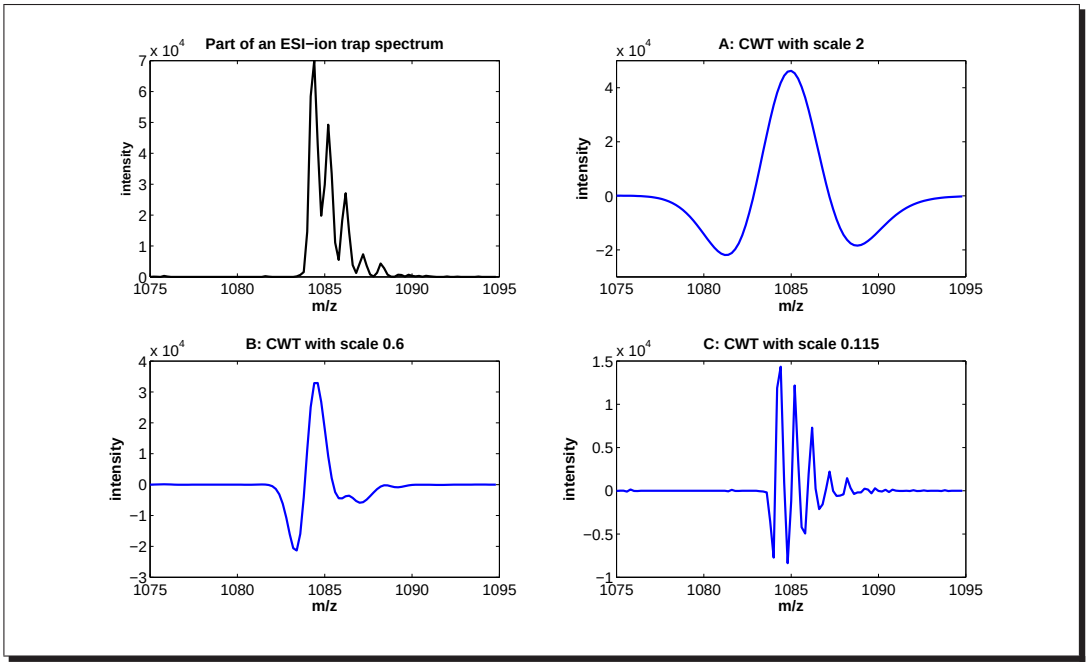

Figure 8.5: *The left hand side plot on top represents an isotopic pattern in an ESI-ion trap spectrum between* 1075 *Th and* 1095 *Th (figure taken from Zerck [2006]). Plots A, B, and C show the continuous wavelet transform of the spectrum using a Mexican hat wavelet with different scale values a (A:* $a = 3$*, B:* $a = 0.3$*, C:* $a = 0.06$*).*

$fwhm$. The FWHM value of the wavelet is derived by solving

$$\psi(x) = (1-x^2)\exp\left(-\frac{x^2}{2}\right) = 0.5 \tag{8.6}$$

for x (since the height of the wavelet is 1) with the commercial computer-algebra-system Maple (version 10). Maple solves Equation 8.6 using the Lambert W-Function (which is the inverse function of $f(W) = W\exp(W)$) and results in the two points $x_1 := -0.626a$ and $x_2 := 0.626a$. Accordingly, the FWHM of the wavelet with scale a is $1.252a$ and the desired peak FWHM $fwhm$ is achieved by a scaling of $a := \frac{fwhm}{1.252}$. A proper scaling factor can therefore be estimated if the resolution of the instrument is known.

Since ψ is described by only a few data points, the convolution of wavelet and signal can be computed very efficiently with pre-tabulated values of ψ. We tabulated the values of the wavelet in the beginning of the peak picking algorithm and determine the required points of ψ during the convolution with the spectrum at discretized translation values b by linear interpolation using the pre-tabulated values of ψ. Thereby the convolution of the wavelet with the signal is approximated by numerical integration. The runtime of the convolution is linear because the filter kernel of $\psi(x)$ contains only a small number of points with respect to the whole spectrum.

To detect the approximate positions of mass spectral peaks, we linearly scan the $W_s(a,b)$ for local maxima.

8.2.2 Searching for a peak's maximum and its endpoints

The maxima in the continuous wavelet transform approximatively represent maximum positions of the mass spectral peaks in the spectrum. Accordingly, we can find in the neighborhood of each maximum position p in the wavelet transform a corresponding maximum position $\hat{p}$ in the spectrum. To filter out chemical noise peaks (see Section 6.1) that have a frequency range similar to that of mass spectral peaks, we introduce an intensity-based threshold t_i. Furthermore, we filter out peaks with low signal-to-noise ratios. The signal-to-noise value of a peak is defined by the signal-to-noise value of the raw data point $\hat{p}$. We use a sliding window approach to estimate the noise level for each raw data point in a mass spectrum. The noise level is defined as the median intensity of all raw data points within the window. The algorithm is implemented using histogramming techniques, such that we achieve a fast estimation of the signal-to-noise values of all raw data points in a spectrum [Bielow, 2006]. If the maximum intensity and the signal-to-noise value at position $\hat{p}$ exceed the user-defined thresholds t_{sne} and t_i, we search for the endpoints of the peak at position $\hat{p}$.
Defining the "ends" of a peak shape becomes difficult when effects such as noise or overlapping of peaks have to be considered. In this case, we cannot expect that the peak's intensity drops below a given threshold before the next peak's area of influence is reached. To solve this problem, we start at the maximum position and proceed to the left and right until either a minimum is reached, or the value drops below a pre-defined noise threshold. A minimum might either be caused by the rising flank of a neighboring peak, or could be a mere noise effect. To discriminate between these two cases, we consider again the $W_s(a,b)$ in the neighborhood, where noise effects are typically smoothed out and peaks can be clearly discerned.

8.2.3 Estimating a peak's centroid

To reduce the effect of asymmetry in the determination of the peak position, we follow the advice from Lehmann [1995] to take only the most intense data points representing a MALDI-TOF mass spectral peak for the computation of its m/z value. We estimate a peak's m/z value, the so-called *peak centroid c*, as an intensity-weighted average using all consecutive set of points next to the maximum with intensity above 70% of the peak's height.

8.3 Peak fitting

We discover the shape of a peak by the fit of an analytical peak function, because shape information can be used in further analysis steps. The fit provides information about the quality of a raw peak. The better a raw peak can be described by a peak function, the less its shape is distorted by noise or other peaks and the more reliable this peak is. In the literature, several different analytical expressions have been proposed for the representation of mass spectral peaks. Since no universally accepted peak shape exists. We chose two common functions that describe the shape of mass spectrometric peaks very well. As we have seen in Section 6.1, imperfections of the mass analyzer often result in asymmetric mass spectral peaks. We take this into consideration and fit asymmetric peak functions to the data that specify the shape of mass spectral peaks precisely.

Figure 8.6 illustrates the procedure of peak fitting. For each detected raw peak, we determine an asymmetric peak function that has the same area, maximum position, and maximum intensity as the raw peak. We explain the asymmetric peak function by two halves of two symmetric peak functions. The left half of the first symmetric peak function has the same area as the left half of the raw peak and it describes the asymmetric peak function until the peak maximum position. Accordingly, to the right of peak maximum position the asymmetric peak function is defined by the right half of the second symmetric peak function.

8.3.1 Fit of an asymmetric Lorentzian and sech^2 peak function

In the current implementation, we fit two peak functions to the data, which are an asymmetric Lorentzian function ($\mathfrak{L}_{h,\lambda(x),\hat{p}}$) and an asymmetric sech^2 ($\mathfrak{S}_{h,\lambda(x),\hat{p}}$) function, but other peak shapes such as double Gaussian profiles [Strittmatter et al., 2003; Kempka et al., 2004] can be easily included. The asymmetric functions $\mathfrak{L}_{h,\lambda(x),\hat{p}}$ and $\mathfrak{S}_{h,\lambda(x),\hat{p}}$

$$\mathfrak{L}_{h,\lambda(x),\hat{p}}(x) = \frac{h}{1+\lambda^2(x)(x-\hat{p})^2} \tag{8.7}$$

and

$$\mathfrak{S}_{h,\lambda(x),\hat{p}}(x) = \frac{h}{\cosh^2(\lambda(x)(x-\hat{p}))} \tag{8.8}$$

where

$$\lambda(x) = \begin{cases} \lambda_l, & x \leq \hat{p} \\ \lambda_r, & x > \hat{p} \end{cases}. \tag{8.9}$$

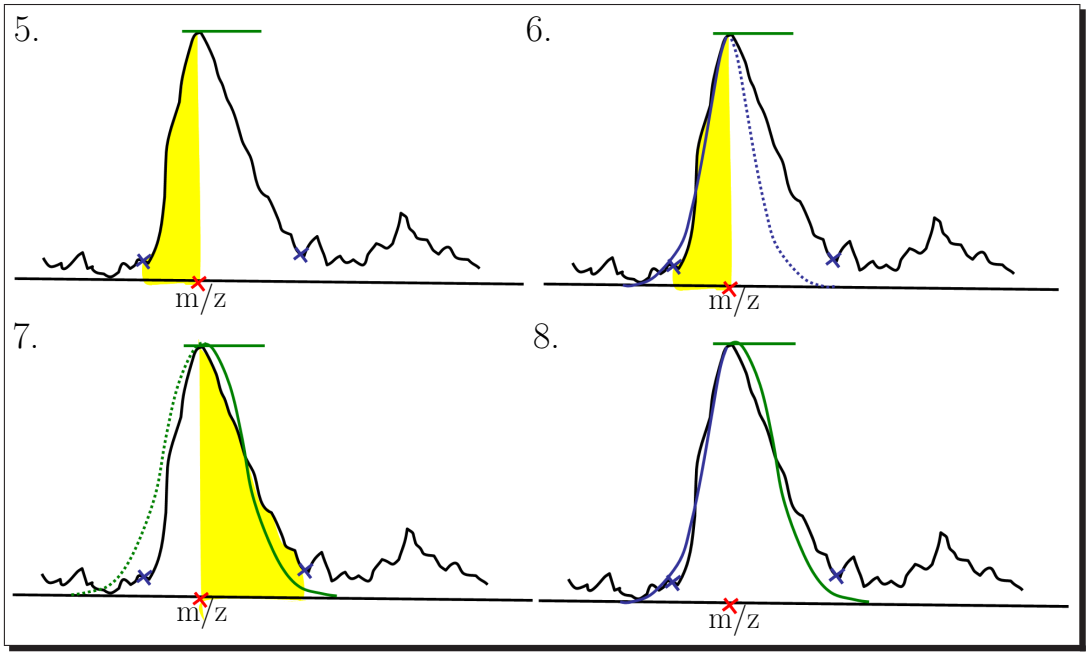

Figure 8.6: *Workflow of the second step of the peak picking algorithm. 5. Estimate the peak's left area, 6. Fit a symmetric peak function to the left, 7. Analogously fit a peak function to the right, 8. Two halves of the symmetric peak function define the resulting asymmetric peak shape.*

are defined by a height parameter h, parameters λ_l and λ_r for the left and right peak width, and a parameter for the peak position $\hat{p}$. A peak can be fitted to the raw data in several ways. In our implementation, we have chosen to use the peak's m/z value at maximum intensity and the area under the experimental signal. Fitting the area of the peak automatically introduces a smoothing effect, yields very good approximations to the original peak shape, and is extremely efficient, since the peak's width can be computed from its area in constant time for the functions considered here. Since the peaks are modeled as asymmetric functions, we integrate from the left endpoint x_l up to the peak maximum position $\hat{p}$ to obtain the left peak area A_l. Analogously, we compute the right peak area A_r between $\hat{p}$ and the right peak endpoint x_r. Let y_r be the intensity at x_r and y_l be the intensity at x_l. From these values, we can finally analytically compute the asymmetric Lorentzian or sech^2 function with position $\hat{p}$ and height h that has the same area A_l as the raw peak from $\hat{p}$ until the intensity value y_l and, and A_r between $\hat{p}$ and the intensity value y_r, respectively.

We describe the derivation of the analytical expression with respect to the fit of a Lorentzian function $\mathfrak{L}_{h,\lambda(x),\hat{p}}$. Assume we are given a mass spectral peak in a raw spectrum, defined by its position $\hat{p}$, the maximum intensity h, the peak endpoints x_l and x_r, the intensity values y_l, y_r at x_l and x_r, as well as the left and right area A_l, A_r. As we want to determine an asymmetric peak $\mathfrak{L}_{h,\lambda(x),\hat{p}}$ as defined in Equation 8.7, we have to fit two symmetric Lorentzian functions, one to

the left peak half, and another to the right peak half. To this end, the right $\mathfrak{L}_{h,\lambda_r,\hat{p}}$ peak function should have the same area from $\hat{p}$ to a $\hat{x}_r$ with $\mathfrak{L}_{h,\lambda_r,\hat{p}}(\hat{x}_r) = y_r$ and it should hold

$$A_r = \int_{\hat{p}}^{\hat{x}_r} \mathfrak{L}_{h,\lambda_r,\hat{p}}(x)\, dx. \tag{8.10}$$

Using the inverse function of $\mathfrak{L}_{h,\lambda_r,\hat{p}}$ we obtain

$$\hat{x}_r = \hat{p} + \frac{1}{\lambda_r}\sqrt{\frac{h}{y_r} - 1}. \tag{8.11}$$

Inserting Equation 8.11 into Equation 8.10, we get

$$A_r = \int_{\hat{p}}^{\hat{p}+\frac{1}{\lambda_r}\sqrt{\frac{h}{y_r}-1}} \mathfrak{L}_{h,\lambda_r,\hat{p}}(x)\, dx \tag{8.12}$$

$$= \frac{h}{\arctan \lambda_r^2 (x-\hat{p})}\Bigg|_{\hat{p}}^{\hat{p}+\frac{1}{\lambda_r}\sqrt{\frac{h}{y_r}-1}} \tag{8.13}$$

and we can conclude that

$$\lambda_r = \frac{h}{A}\arctan\sqrt{\frac{h}{y_r} - 1} \tag{8.14}$$

The computation of the width parameter λ_l of the left Lorentzian function $\mathfrak{L}_{h,\lambda_l,\hat{p}}$ is completely analogous.

The analytical expression for the width parameter λ_l and λ_r of an asymmetric sech^2 function is derived the same way. For $\hat{x}_r$ holds with $\mathfrak{S}_{h,\lambda_r,\hat{p}}(\hat{x}_r) = y_r$

$$\hat{x}_r = \hat{p} + \frac{1}{\lambda_r}\operatorname{arccosh}\sqrt{\frac{y_r}{h}}. \tag{8.15}$$

Since $\mathfrak{S}_{h,\lambda_r,\hat{p}}$ should have the same total intensity A_r from $\hat{p}$ to $\hat{x}_r$ we have to solve the definite integral

$$A_r = \int_{\hat{p}}^{\hat{p}+\frac{1}{\lambda_r}\operatorname{arccosh}\sqrt{\frac{y_r}{h}}} \mathfrak{S}_{h,\lambda_r,\hat{p}}(x)\, dx \tag{8.16}$$

to determine the width parameter λ_r of $\mathfrak{S}_{h,\lambda_r,\hat{p}}$

$$\lambda_r = \frac{h}{A}\sqrt{1 - \frac{y_r}{h}}. \tag{8.17}$$

The FWHM value $fwhm_{\mathfrak{L}}$ of $\mathfrak{L}_{h,\lambda(x),\hat{p}}$ is given by the half FWHM of $\mathfrak{L}_{h,\lambda_l,\hat{p}}$ plus the half FWHM value of $\mathfrak{L}_{h,\lambda_r,\hat{p}}$ (the FWHM value of the asymmetric sech^2 function is given analogously). Solving

$$\frac{h}{1+\lambda_l^2(x)(x-\hat{p})^2} = \frac{h}{2}, \tag{8.18}$$

we obtain

$$x = \frac{1}{\lambda}. \tag{8.19}$$

Accordingly, $fwhm_{\mathfrak{L}}$ is given by

$$fwhm_{\mathfrak{L}} = \frac{1}{\lambda_l} + \frac{1}{\lambda_r}. \tag{8.20}$$

The FWHM value of $\mathfrak{S}_{h,\lambda(x),\hat{p}}$ is given by

$$fwhm_{\mathfrak{L}} = \frac{\operatorname{arccosh}(\sqrt{2})}{\lambda_l} + \frac{\operatorname{arccosh}(\sqrt{2})}{\lambda_r} \tag{8.21}$$

$$= \frac{\ln(\sqrt{2}+1)}{\lambda_l} + \frac{\ln(\sqrt{2}+1)}{\lambda_r}. \tag{8.22}$$

8.3.2 Examination of the best fitting function

In the previous subsection, we used the peak position $\hat{p}$, the maximum intensity h, the intensities at the peak endpoints x_l and x_r, as well as the left and right area A_l, A_r of each mass spectral peak in the mass spectrum to determine an asymmetric Lorentzian function and an asymmetric sech^2 function. Both functions are representations of the mass raw peak and have the same area as the original raw spectral peak, but in most cases one of the analytical peak shapes defines the original peak shape more precisely. To determine the "best" fitting peak function we perform a correlation test based on the fact that if two variables vary together there is a lot of covariation or correlation.
Suppose we are given the n raw data points $\{e_1, \ldots, e_n\}$ representing a mass spectral peak. Let $\mathrm{m/z}(e_i)$ be the m/z value and $\mathrm{int}(e_j)$ the intensity of the j-th raw data point. Furthermore, let $p \in \{\mathfrak{L}_{h,\lambda(x),\hat{p}}, \mathfrak{S}_{h,\lambda(x),\hat{p}}\}$ be either an asymmetric Lorentzian function or an asymmetric sech^2 function.
The average intensity $\bar{i}$ of the raw peak and the average intensity $\bar{p}$ of the fitted peak function are given by

$$\bar{i} := \frac{1}{n}\sum_{j=1}^{n} \mathrm{int}(e_j) \qquad \bar{p} := \frac{1}{n}\sum_{j=1}^{n} p(\mathrm{m/z}(e_j)).$$

The *coefficient of determination* r^2, which is the squared correlation coefficient r, developed by Pearson in 1895 [J. L. Rodgers, 1988] is then given by

$$r^2 := \frac{\sum_j^n (\mathrm{int}(e_j) - \bar{i})^2 (p(\mathrm{m/z}(e_j)) - \bar{p})^2}{\sum_j^n (\mathrm{int}(e_j) - i)^2 \sum_j^n (p(\mathrm{m/z}(e_j)) - \bar{p})^2}. \tag{8.23}$$

The correlation coefficient r^2 has a value that ranges from zero to one, and is the fraction of the variance in the two variables that is shared. For example if $r^2 = 0.8$, then 80% of the variance is shared between the peak function p and the raw data points of the original peak.

By means of the correlation coefficients estimated for the $\mathfrak{L}_{h,\lambda(x),\hat{p}}$ and the $\mathfrak{S}_{h,\lambda(x),\hat{p}}$, we take the function that represents the raw peak best. If the correlation coefficient of both functions is lower than a certain threshold t_{corr}, and the peak shape cannot be sufficiently described either by a Lorentzian or a sech^2 function, we reject the peak as a mass spectral peak.

8.4 Separation of overlapping peaks

A low resolution of the mass analyzer as well as a high charge state of the measured compound may result in a high overlap of mass spectral peaks. Hence, broad or extremely asymmetric peaks in a mass spectrum are often an indicator for the overlap of several peaks. Consider, for example, the charge two isotopic pattern of bombesin in Figure 8.7. The first three iso-

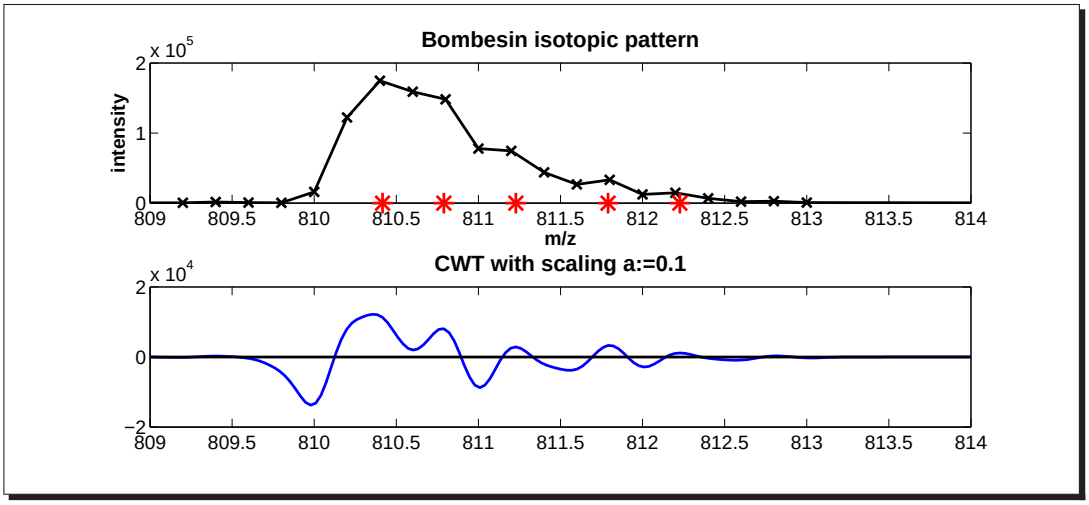

Figure 8.7: *Top: Isotopic pattern of a doubly charged bombesin measured with ESI-ion trap. Bottom: Continuous wavelet transform of the isotopic pattern with scale* $a := 0.1$.

topic peaks were not well resolved by the mass analyzer (here an ion trap) and additional noise prevents the occurrence of three maxima in the measured spectrum. All peak picking algorithms that detect the peak positions by searching for maxima in the spectrum will fail to pick five individual peaks. Figure 8.7 shows that our approach with the idea to detect the peak positions in the continuous wavelet transform $W_s(a,b)$ with fixed scale a can solve the problem in principle; for highly convoluted peak patterns, a further modification is required that will be described below. All five approximate peak positions are represented by maxima in $W_s(a,b)$ (using scale $a := 0.1$) at positions 810.416 Th, 810.790 Th, 811.228 Th, 811.790 Th, and 812.227 Th. The average distance between adjacent maxima in $W_s(a,b)$ is 0.45 Th and agrees well with the theoretical peptide mass rule for isotopic patterns of charge two peptides

$d := \frac{1.00235 \text{ Th}}{2} = 0.501175$ Th (see Section 6.1).
As described in Section 8.2.1 to Section 8.3.2 our basic peak picking algorithm (see Figure 8.2, line 1-25) will detect a maximum in the continuous wavelet transform with a fixed scale a and afterward search for the corresponding maximum in the original signal. Going back to the original raw data allows for an accurate determination of the position and the ion count (total ion count and maximum ion count). Furthermore, it enables the representation of the raw peak by an analytical peak function. But in case of a highly convoluted peak pattern, this approach will fail to detect the individual peaks and result in a broad peak at positions 810.51 Th and two narrower peaks at 811.6 Th and 812.2 Th as illustrated in Figure 8.8.

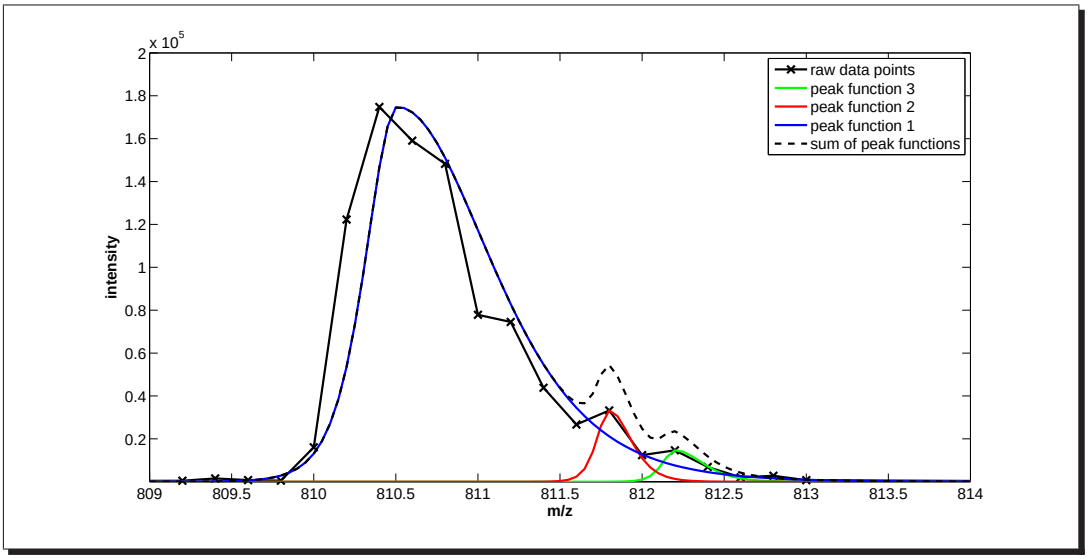

Figure 8.8: *Charge two isotopic pattern of bombesin and the three peak functions determined by the basic peak picker. The peak positions are* 810.51 *Th,* 811.8 *Th, and* 812.2 *Th. The dotted line shows the sum of the three peak functions.*

Every deisotoping algorithm will have problems to discover the right charge and the monoisotopic mass of bombesin with respect to the three peaks. To solve the problem of overlapping peaks, we developed a sophisticated separation technique that uses the continuous wavelet transform to determine the number of convolved peaks and estimates the peak parameters by a non-linear optimization technique in the raw mass spectrum.

After our basic peak picking procedure, we determine peaks in the method `separateOverlappingPeaks` that likely represent an convolved peak pattern and separate the overlapping peaks. A broad or asymmetric peak is identified by its FWHM value and its symmetric value. The symmetric measure $sym \in [0,1]$ of a peak is defined by $sym := \frac{\lambda_l}{\lambda_r}$ if $\lambda_l < \lambda_r$ and $sym := \frac{\lambda_r}{\lambda_l}$ if $\lambda_l \geq \lambda_r$ whereby λ_l and λ_r are the left and right width parameter of the peak function. Each peak with a FWHM value greater than a user defined t_{FWHM} or a

symmetric value less than a user-defined threshold t_{sym} is labeled as too broad or asymmetric and is examined more closely in the next step. The pseudocode of the separation procedure is shown in Figure 8.9.

A broad or asymmetric peak *peak* that has one or two neighboring peaks (on the left and the right hand side) lying within the peptide mass rule for charge one pattern $t_d \approx 1.1$ Th is assumed to be part of an isotopic pattern. If either the distances to the two adjacent peaks are dissimilar or if the FWHM values of the neighboring peaks are much smaller than the FWHM of *peak* we "deconvolve" *peak*. Furthermore, we also separate broad or asymmetric peaks that have no neighboring peak within t_d, since we assume those peaks represent an overlapping isotopic peak pattern.
The method that determines the number of overlapping peaks in the continuous wavelet transform as well as the algorithm that estimates the parameters of the convolved peaks are described in more detail in the next two subsections.

8.4.1 Determining the number of overlapping peaks

In Section 8.2 we have seen that, if we have a rough estimate of the frequency range we are interested in, the continuous wavelet transform with a fixed scale a can be used to localize this information. Figure 6.3 shows the capability of the CWT using the Mexican hat wavelet to localize the approximate positions of the convolved mass spectral peaks. Given a broad or asymmetric peak function, we go back to the raw data and compute the continuous wavelet transform $W_s(a,b)$ of the original raw peak that is represented by the raw data points within the endpoints x_l and x_r. Thereby, we use the same scale as in the basic peak picking step. Subsequently, we take the positions of the maxima in $W_s(a,b)$ as initial estimates of the hidden peak positions. Maxima in $W_s(a,b)$ that lie close to the peak endpoints x_l and x_r are disregarded since they are often caused by side effects.

8.4.2 Discriminating overlapping peaks

In the previous section, we determined the number $k \in N^+$ of convolved peaks with respect to the continuous wavelet transform of a broad or asymmetric raw data peak given by m data points $\{e_1,\ldots,e_m\}$. Each raw data point $e_i := (x_i, y_i)$ (with $i = 1,\ldots,m$) is defined by its m/z position x_i and an intensity value y_i.
We now search for the k asymmetric sech^2 peak functions $\mathfrak{S}_i$ with $i = \{1,\ldots,k\}$ that describe the convolved raw peak best. For a true separation, we need to fit the sum of all k peaks $\mathfrak{S}_{h_i,\lambda_{l_i},\lambda_{r_i},\hat{p}_i}$ to the experimental raw signal $\{e_1,\ldots,e_m\}$. Hence, our analytical peak model M

PEAK SEPARATION PHASE

Input: The raw data *raw_data*, the determined list *peak_list* of all mass spectral peaks picked in *raw_data*

Output: A list *peak_list* of the mass spectral peaks after the separation of overlapping peaks

```
1:  for all peaks peak in peak_list do
2:      fwhm:=getFWHM(peak)
3:      asym:=getAsymmetryValue(peak)
4:      // search for broad and asymmetric peaks
5:      if (fwhm > t_FWHM) ∨ (asym < t_asym) then
6:          (d_l, d_r):=getDistanceToNeighbors(peak,peak_list)
7:          // peak has two neighbors with a distance less than t_d
8:          if (d_l < t_d) ∧ (d_r < t_d) then
9:              // dissimilar distances to the left and right adjacent peak
10:             if dissimilarDistances(d_l, d_r) then
11:                 {peak_1, ..., peak_m}:=separatePeak(peak,raw_data)
12:                 replace({peak_1, ..., peak_m},peak,peak_list)
13:             end if
14:         else
15:             // peak has only one neighbor peak_x with distance d_x less than t_d
16:             if (d_l < t_d) ∨ (d_r < t_d) then
17:                 if satifiesPeptideMassRule(d_x) then
18:                     fwhm_x:=getFWHMNeighbor(peak_x)
19:                     if dissimilarFWHMValues(fwhm, fwhm_x) then
20:                         {peak_1, ..., peak_m}:=separatePeak(peak,raw_data)
21:                         replace({peak_1, ..., peak_m},peak,peak_list)
22:                     end if
23:                 else
24:                     {peak_1, ..., peak_m}:=separatePeak(peak,raw_data)
25:                     replace({peak_1, ..., peak_m},peak,peak_list)
26:                 end if
27:             end if
28:             // peak has no neighbor with a distance less than t_d
29:         else
30:             {peak_1, ..., peak_m}:=separatePeak(peak,raw_data)
31:             replace({peak_1, ..., peak_m},peak,peak_list)
32:         end if
33:     end if
34: end for
```

Figure 8.9: *Pseudocode of the peak function* `separateOverlappingPeaks`.

for the k peaks is given by

$$M(a,x) := \sum_{i=1}^{k} \mathfrak{S}_{h_i,\lambda_{li},\lambda_{ri},\hat{p}_i}(x) \tag{8.24}$$

where the i-th peak of $\mathfrak{S}$ depends on four parameters $\{h_i, \lambda_{li}, \lambda_{ri}, \hat{p}_i\}$ $(i = 1,\ldots,k)$ with the peak position $\hat{p}_i$, the height h_i, and the left and right width parameter $\lambda_{li}, \lambda_{ri}$. Since we assume that the convolved peaks are part of the same isotopic pattern, we use the same left and right width parameter for all peaks. Hence, the parameter vector $a \in \mathbb{R}^{2+2k}$ of M is given by $a := (\lambda_l, \lambda_r, h_1, \hat{p}_1, \ldots, h_k, \hat{p}_k)^T$. We now fit the peak model M to the data $\{(x_1,y_1),\ldots,(x_m,y_m)\}$ by solving a non-linear least squares problem: Find a local minimizer $a^* \in \mathbb{R}^{2+2k}$ (see Definition 5.3.2) for

$$F(a) := \frac{1}{2}\sum_{i=1}^{m}(f_i(a))^2 \tag{8.25}$$

with

$$f_i := y_i - M(a,x_i) = y_i - \sum_{j=1}^{k} p_j(x_i). \tag{8.26}$$

Starting from an initial guess of the parameter vector $a^0 := (\lambda_l^0, \lambda_r^0, h_1^0, \hat{p}_1^0, \ldots, h_k^0, \hat{p}_k^0)^T \in \mathbb{R}^{2+2k}$, we search for the optimal parameter vector a^* that minimizes the sum of squared residuals in Equation 8.25. The initial peak positions $\hat{p}_i^0$ are given by the k maxima in the continuous wavelet transform, the initial height h_i^0 by the intensity in the raw data at position $\hat{p}_i^0$, and the λ_l^0 and λ_r^0 values can be defined by the user. We find a^* using the Levenberg-Marquardt [Marquardt, 1963] algorithm (for more details, see Section 5.3) implemented in the GSL [Galassi et al., 2006]. The Levenberg-Marquardt algorithm is a powerful heuristic that is based on the steepest descent method and the Gauss-Newton method. In each iteration, the algorithm settles for a steepest descent like or a Gauss-Newton like step by means of local curvature information (the ratio between the actual and predicted decrease in function value). After each iteration, the convergence criteria are checked: is the maximal number of iterations reached or if either the absolute or relative error is small enough to characterize the location of the minimum?

The values for the maximal number of iterations and the threshold for absolute and relative error can also set by the user. The GSL offers the possibility to handle additional parameters that might avoid undesirable effects such that large shifts of peaks or negative peak width and height parameters. Hence, we introduce a penalty term for height and width parameters that fall below certain thresholds. Furthermore, we force the distance between the separated peaks to meet the peptide mass rule by penalizing too small or too large distances.

The dotted line in Figure 8.10 shows the model function $M(a^*,x)$ with respect to the localized minimizer a^* resulting from the optimization step.

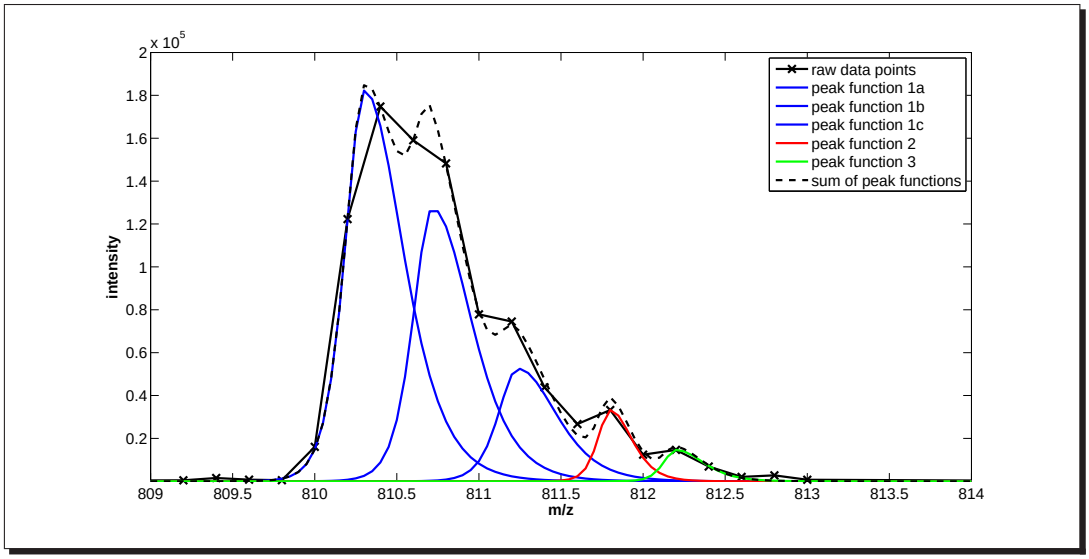

Figure 8.10: *Charge two isotopic pattern of bombesin and the five peaks resulting from the basic peak picker plus the separation method for overlapping peaks, which deconvolve the first broad peak (see Figure 8.8) into three individual peaks. The peak positions are* 810.303 *Th,* 810.717 *Th,* 811.237 *Th,* 811.8 *Th, and* 812.2 *Th. The dotted line shows the sum of the five peak functions.*

8.5 Optimization of all peak parameters

The peaks computed so far typically yield a reasonable approximation of the true signal, especially for well-resolved, clearly separated peaks. We tried to further improve accuracy and perform an additional (optional) optimization step of all picked peaks in a spectrum. In the basic peak picker (see Figure 8.2, line 1-25), each of the peaks has been fitted independently of the others and only during the separation of overlapping peaks we fit the sum of convolved peaks to the experimental signal. In this step, we want to optimize the parameters of all picked peaks in the spectrum by minimizing the sum of squared residuals between the determined peak functions and the original raw signal. Our peak model M is now given by all peak functions p_i picked in the spectrum,

$$M(a,x) := \sum_{i=1}^{k} p_{h_i,\lambda_{l_i},\lambda_{r_i},\hat{p}_i}(x) \tag{8.27}$$

whereby p_i can either represent an $\mathfrak{L}$ or an $\mathfrak{S}$ peak function (compare Section 8.3). Hence, the model M depends on $4k$ parameters and the parameter vector is defined by $a := (h_1,\lambda_{l1},\lambda_{r1},\hat{p}_1,\ldots,h_k,\lambda_{lk},\lambda_{lk},\hat{p}_k)^T \in \mathbb{R}^{4k}$. Since the number of peaks in a spectrum and thereby the number of the parameters can be very high, we decompose the optimization problem into smaller subproblems. After sorting all peak functions with respect to their positions we linearly search for connected peaks that are afterward fit simultaneously. Thereby, two

peaks are connected if the distance between the peak positions is smaller than a certain threshold.
We use the Levenberg-Marquardt algorithm to find a local minimizer a^* for the function defined in Equation 8.25. For a group of k connected peaks p_i, the initial parameter vector $a^0 := (h_1^0, \lambda_{l1}^0, \lambda_{r1}^0, \hat{p}_1^0, \ldots, h_k^0, \lambda_{lk}^0, \lambda_{rk}^0, \hat{p}_k^0)^T \in \mathbb{R}^{4k}$ is given by the four peak parameters of each p_i $(i = 1, \ldots, k)$.

We again use the additional parameters provided by the GSL to introduce penalty terms for height and width values that fall below certain thresholds. Furthermore, we penalize large changes of position parameters during an iteration.
The dotted line in Figure 8.11 shows the model function $M(a^*, x)$ with respect to the localized minimizer a^* resulting from the optimization step.

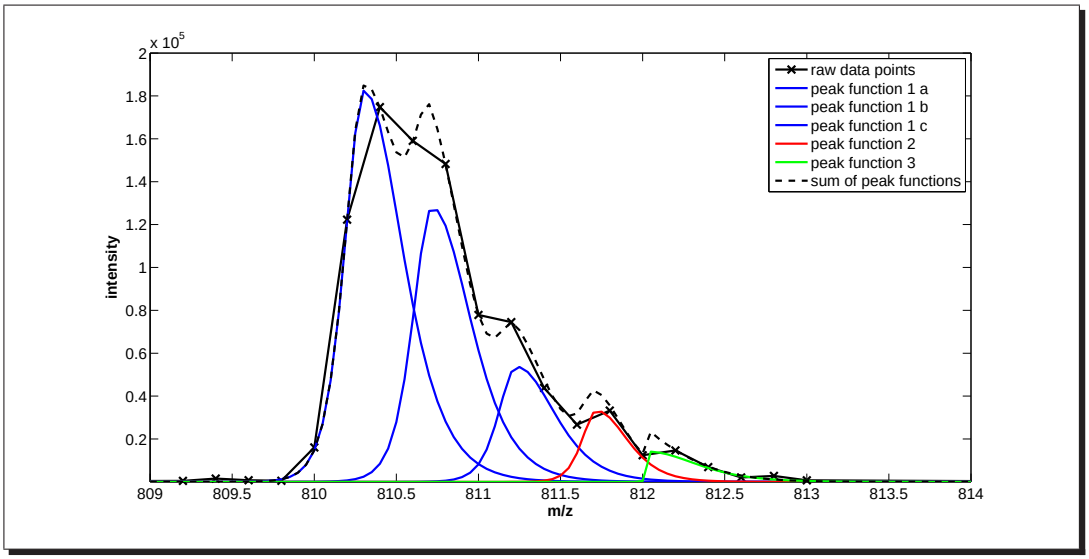

Figure 8.11: *Optimization of all peak parameters: Charge two isotopic pattern of bombesin and the five peaks resulting from the basic peak picker plus the separation method for overlapping peaks and the optimization of all peak parameters. Note the slight differences to Figure 8.10: The peak positions are* 810.304 *Th,* 810.718 *Th,* 811.231 *Th,* 811.722 *Th, and* 812.025 *Th. The dotted line shows the sum of the five peak functions.*

8.5.1 The PeakPicker TOPP tool

We provide an application for "The OpenMS Proteomics Pipeline (TOPP)" [Kohlbacher et al., 2007] application called *PeakPicker* for the extraction of peaks in mass spectra that implements the algorithm proposed in Chapter 8. The input and output format of spectra is `mzData` (see Figure 8.12).

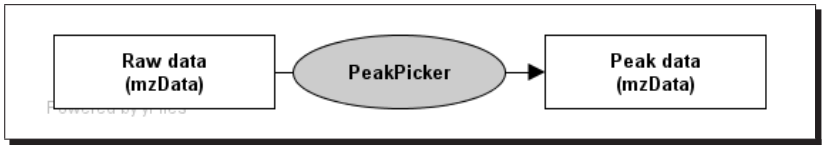

Figure 8.12: *Peak picking with the PeakPicker tool.*

All parameters are provided by an XML-based control file. The usage of the tool is described in the TOPP documentation and an example is given in the TOPP tutorial.
The PeakPicker application, as all other TOPP tools, is based on the OpenMS library. Figure 8.13 shows the class diagram of our peak picking classes in UML format. The classes are described in the OpenMS documentation and examples of use can be found in the OpenMS tutorial.

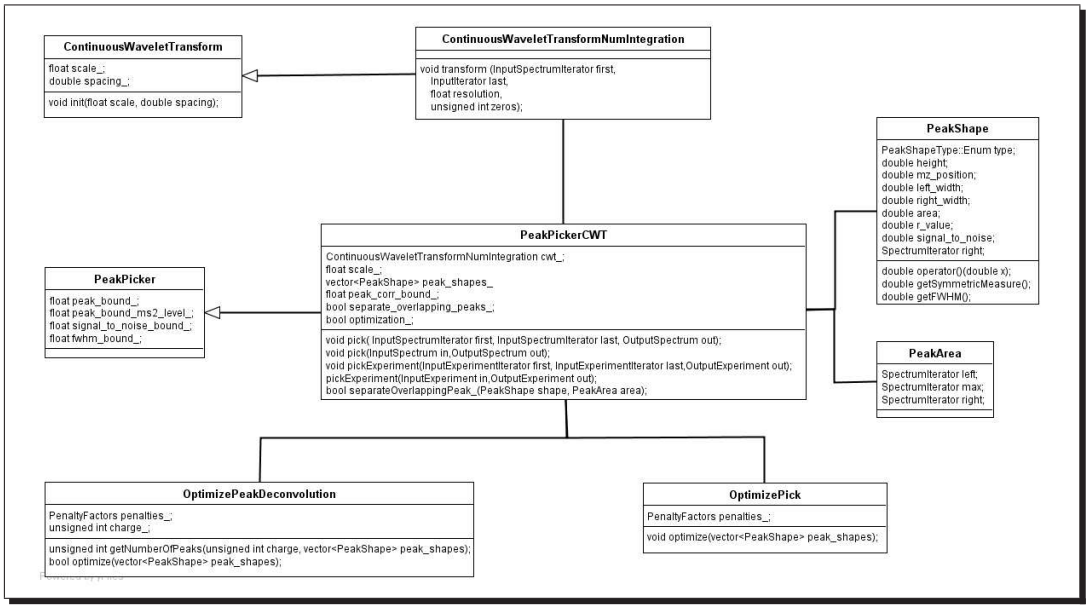

Figure 8.13: *UML class diagram of the main classes for peak picking.*

Chapter 9

Experiments

The qualitative assessment of a peak picking scheme is a non-trivial problem and its solution by a straight-forward and general approach is still missing. Obviously, an algorithm that solves the problem should compute the peak's centroid, height, and area as accurately as possible while featuring a high sensitivity and specificity. To determine the accuracy of, e.g., a peak's centroid, the correct mass value is needed, and thus peak picking algorithms are typically tested against a spectrum of known composition, e.g., a standard peptide mixture or the tryptic digest of a certain protein. Comparing the features of the peaks found in the spectrum with the theoretical values gives a measure of the algorithm's capabilities, typically expressed as the average absolute and relative deviation (measured in ppm). Unfortunately, these results are heavily affected by the quality of the experimental data, and additional issues such as the calibration. Consequently, peak picking algorithms are typically tested against particularly well-resolved spectra, and internal calibration methods are employed. This usually results in high mass measurement accuracy, but the quality of the peak picking algorithms cannot be judged independently of the quality of the calibration scheme. From a user's perspective, on the other hand, obtaining similarly well-resolved spectra is often infeasible, and internal calibration is not always an option. Thus, we have decided to demonstrate the capabilities of our approach on both LC-MS data measured by an ion trap with low resolution, containing severely overlapping isotope patterns, as well as on highly resolved MALDI-TOF spectra.

As described in Chapter 7, most peak picking algorithms are designed for a specific data type and, furthermore, they often are not freely available. Li et al. [2005] and Bellew et al. [2006] propose algorithms for the determination of mass spectral peaks, but those methods are closely connected with their 2D feature detection procedures. Thus, they are not appropriate for the comparison to our peak picking approach.
Hence, we decided to use the vendor-supplied software on the same spectra in both experiments to provide a fair means of comparison.

9.1 Sample preparation and MS analysis

Peptide mix ESI: A peptide mix (peptide standards mix #P2693 from Sigma Aldrich) of nine known peptides (bradykinin (*F*), bradykinin fragment 1-5 (*B*), substance P (*H*), [Arg^8]-vasopressin (*E*), luteinizing hormone releasing hormone bombesin (*G*), leucin enkephalin (*A*), methionine enkephalin (*C*), oxytocin (*D*)). Sample concentration was 0.25 ng/μl, injection volume 1.0 μl. LC separation was performed on a capillary column (monolithic polystyrene/-divinylbenzene phase, 60 mm x 0.3 mm) with 0.05% trifluoroacetic acid (TFA) in water (eluent A) and 0.05% TFA in acetonitrile (eluent B). Separation was achieved at a flow of 2.0 μl/min at 50 °C with an isocratic gradient of 0–25% eluent B over 7.5 min. Eluting peptides were detected in a quadrupole ion trap mass spectrometer (Esquire HCT from Bruker, Bremen, Germany) equipped with an electrospray ion source in full scan mode (m/z 500-1500).

Peptide mix MALDI: The MALDI matrix solution was prepared as a CHCA thin layer by ultrasonicating an excess of CHCA in 90% tetrahydofurane, 0.1% trifluoroacetic acid (TFA). A PolyK-mixture with 6.4 mg/ml polylysine in 1% TFA was deposited onto the matrix and dried. Afterward, the samples were washed by depositing 2 μl of 1% TFA and 1 mM n-octylglucopyranoside, and immediately aspirated. Peptide samples (with 19 known peptides: bradykinin (*A*), angiotensin II (*B*) and I (*C*), substance P-methylester (*D*), substance P-methylester (ox.) (*E*), fibrinopeptide A (*F*), Glu1-fibrinopeptide A (*G*), bombesin (*H*), bombesin (ox.) (*I*), renin substrate (human) (*J*), ACTH clip 1-17 (*K*), ACTH clip 1-17 (ox.) (*L*), ACTH clip 18-39 (*M*), ACTH clip 3-24 (*N*), ACTH clip 3-24 (ox.) (*O*), ACTH clip 1-24 (*P*), ACTH clip 1-24 (ox.) (*Q*), somatostatin (*R*), and Insulin B chain (ox.) (*S*)) were prepared using the CHCA surface affinity preparation, previously described in [Gobom et al., 2001].
Mass analysis of positively charged peptide ions was performed on an Ultraflex II LIFT MALDI-TOF/TOF mass spectrometer (Bruker Daltonics, Bremen, Germany), equipped with a SmartBeam solid-state laser. Positively charged ions in the m/z range 500-4500 Da were analyzed automatically in the reflector mode. Altogether, 100 spectra were recorded, where each was the sum of 800 single-shot spectra acquired at two different locations of each MALDI sample.

9.2 Mass accuracy and separation capability in low resolved LC-MS measurements

To assess the performance of our peak picking scheme on a low resolved LC-MS run on on the peptide mixture (dataset *Peptide mix ESI*), we determined how often each peptide was found in the expected retention time interval, and whether the corresponding isotope patterns were discovered and separated. Furthermore, we computed the resulting relative errors of the monoisotopic peak's centroid compared to the theoretical monoisotopic mass. An isotopic

pattern is defined to be discovered if it lies within a predefined RT range and is given by at least three consecutive peaks. The distance between the isotopic peaks should be similar to the theoretical distance defined by the peptide mass rule (see page 51). Furthermore, the absolute distance between the observed and the theoretical peak centroid positions may not exceed a certain threshold. The same analysis was performed with the Bruker software *Data Analysis 3.2*, using the Apex algorithm recommended for ion trap data. The resolution of the data set is critically low ($R_{FWHM} \approx 2300$ around m/z 800) with a Δm value of 0.2 Th, implying that each peak is represented by as little as 3–6 data points, and instead of a sophisticated calibration, we only allowed for a constant mass offset to keep the number of fit parameters as small as possible. Using recommended signal-to-noise settings in the Bruker software turned out to miss a large number of the isotopic patterns. Therefore, we decided to perform two tests against the Bruker software, one with the recommended setting II_b (minimal FWHM 0.2 Th, minimal signal-to-noise ratio 1, minimal intensity 500), and one with a significantly reduced signal-to-noise threshold and peak bound II_a (minimal FWHM 0.1 Th, minimal signal-to-noise ratio 0.1, minimal intensity 100), leading to a total number of peaks comparable to the peaks I_* (minimal FWHM 0.2 Th, minimal intensity 500) found by the peak picking method described in Lange et al. [2006]. We compare the peak lists I_*,II_a and II_b with the peak lists determined by the current basic peak picking algorithm I_a (minimal FWHM 0.08 Th, minimal signal-to-noise ratio 6, minimal intensity 200). Furthermore, we evaluate the peak lists resulting from the additional separation step I_b and the optimization step I_c. The results of these tests are shown in Table 9.1. For each peptide, we estimated the average relative error of the monoisotopic position (the theoretical monoisotopic position is given by m_{theo}) and counted the number of scans, in which the peptide was discovered, and the total number of isotopic peaks associated with the measured peptide ions (shown in brackets). The "true" number of scans and peaks for each peptide was manually determined by an expert and is given by $man.op$.

Considering the resolution of the raw data, and the lack of sophisticated internal calibration, the mass accuracy that was obtained in these experiments is remarkable. Particularly important is the behavior on highly convoluted charge two isotopic patterns: as can be seen from the number of correctly identified and separated patterns shown in Table 9.1, our algorithms, both the former version I_* and the current enhanced method (I_a, I_b, and I_c), successfully deconvolute significantly more of these patterns than the established approaches. The basic peak picking approach I_a was actually able to resolve the isotopic pattern of all charge one peptides in the expected scans and missed only a small number of isotopic peaks due to their low signal-to-noise ratios. Considering the total number of peaks in I_* and I_a, we significantly increased the sensitivity of our peak picking algorithm by the incorporation of a robust signal-to-noise estimator. The additional separation of overlapping peaks I_b discriminates many of the highly convolved charge two peak patterns, see, for example, the isotopic pattern of doubly charged bombesin (peptide F) in Figure 8.10. The optimization of all peak parameters yields a minor improvement in mass accuracy for the charge two pattern and discovers some more isotopic

Table 9.1: *Evaluation of dataset* Peptide mix ESI. *In the table, I_* denotes the results of our peak picking method described in Lange et al. [2006], I_a of our current peak picking algorithm, I_b the current peak picking algorithm with the separation of overlapping peaks, and I_c represents the results of the current peak picking method along with the separation of overlapping peaks and the optimization of peak parameters. Method II_a denotes the Apex algorithm with reduced thresholds, and II_b the Apex method with default settings. The number of discovered and separated scans for each peptide is given by #occ.scans and #occ.peaks denotes the total number of separated isotopic peaks corresponding to the peptide within the scans.*

			rel. err. [ppm] *#occ.scans* (*#occ.peaks*)					
peptide	***z***	**m_{theo} [Da]** *man. op.*	**I_***	**I_a**	**I_b**	**I_c**	**II_a**	**II_b**
A	1	***555.269***	37	**31**	**31**	35	72	38
		14 (52)	**14** (57)	**14** (44)	**14** (44)	**14** (44)	**14** (68)	**14** (51)
B	1	***572.307***	16	**12**	**12**	19	48	16
		38 (117)	29 (88)	**38** (118)	**38** (118)	**38** (118)	39 (163)	29 (88)
C	1	***573.226***	30	**25**	**25**	26	40	28
		15 (56)	**15** (62)	**15** (51)	**15** (51)	**15** (51)	**15** (84)	**15** (60)
D	1	***1006.437***	39	29	29	65	-	**7**
		11 (52)	**11** (52)	**11** (46)	**11** (46)	**11** (46)	0 (0)	5 (18)
E	1	***1083.422***	40	38	38	55	-	**12**
		8 (36)	7 (35)	**8** (36)	**8** (37)	7 (31)	0 (0)	2 (8)
F	2	***1059.561***	-	-	146	**107**	-	-
		19 (73)	0 (0)	0 (0)	4 (16)	**7** (28)	0 (0)	0 (0)
G	2	***1182.557***	83	77	86	**64**	-	-
		18 (71)	10 (33)	9 (30)	14 (50)	**15** (54)	0 (0)	0 (0)
H	2	***1347.712***	40	**35**	57	48	-	-
		13 (52)	8 (30)	8 (28)	**13** (50)	**13** (50)	0 (0)	0 (0)
I	2	***1619.799***	48	**37**	78	64	-	109
		16 (74)	7 (34)	8 (33)	13 (60)	**14** (64)	0 (0)	1 (3)
total # occ. peaks			60485	15043	18958	18958	77459	22092

patterns. However, the peak positions of the charge one peptides are more precisely defined by the centroid (Section 8.2.3) than by the position of the fitted peak function.

In addition, it should be mentioned that the data collection of a mass spectrometer is a time consuming process; therefore our algorithm runs in real time and can be applied online. On the LC-MS spectra of about 100 MB of data, the former peak picking algorithm I_* took several seconds on a PC with dual 3 GHz CPU, while the following optimization run lasted for about 1 to 5 minutes, depending on the number of iterations performed. The runtimes of the current

peak picking algorithm, measured as absolute CPU time in seconds, are shown in Table 9.2. The algorithm I_* of Lange et al. [2006] that we presented there is an earlier version of our

Table 9.2: *Runtimes of the current peak picking algorithm on dataset* Peptide mix ESI*. In the table, I_a denotes our current peak picking algorithm, I_b the current peak picking algorithm with the separation of overlapping peaks, and I_c represents the current peak picking method with separation of overlapping peaks and the optimization of peak parameters.*

	$\mathbf{I}_a$	$\mathbf{I}_b$	$\mathbf{I}_c$
CPU time in seconds	11.07	36.20	60.68

current peak picking approach I_a. The main loop in I_* was less optimal and in the modified version I_a we avoid the computation of the continuous wavelet transform subsequent to each detection of a peak, and update the wavelet transform not until we processed all maxima in the wavelet transform. This leads to a speed-up in runtime and I_a takes only 11 s and enabling the optional separation method the whole runtime is with 36 s far below a minute. The optimization of all peak parameters additionally takes only half a minute (allowing for 100 iterations). The applicability of the proposed scheme is not restricted to low-resolution data nor to ESI data. To demonstrate this, we performed our peak picking algorithm on a well-resolved MALDI-TOF data set and present the results in the following.

9.3 Mass accuracy in high resolution MALDI-TOF measurements

To prove that the performance of our approach is independent of the underlying instrument type and the different analysis aims, we will demonstrate the performance of our peak picking algorithm on the high-resolution MALDI-TOF/TOF data set *Peptide mix MALDI*. Due to the good resolution of the mass analyzer the mass spectral peaks are well separated in all spectra. To this end we only compared the accuracy and precision values of our approach with the accuracy and precision measurements of the Centroid algorithm implemented in the vendor supplied *flexAnalysis 3.0* software.

Prior to the peak picking process in the m/z dimension, we performed a sophisticated calibration procedure similar to Gobom et al. [2002] on the 100 time-of-flight spectra. To avoid systematic errors and yield comparable results, we used the same peak picker for the calibration process and the subsequent peak picking step in m/z. The calibration procedure is shortly summarized in the following section.

9.3.1 Spectra calibration

For external calibration, first the monoisotopic signals from the polylysine polymers in the mass range between 737 and 4096 Da were labeled in the calibrant spectra, using a peak picking algorithm. Afterward, the labeled time-of-flight values were converted to m/z values using the calibration constants of the instrument. Subsequently, we determined the relationship between the time-of-flight dimension and the m/z dimension by fitting a quadratic function to the TOF values of the PolyK peaks and their expected masses. The remaining systematic error between the expected masses and the calculated m/z values was estimated by fitting a cubic spline. This error function together with the quadratic function defines the final calibration function, which was furthermore used to convert the flight times of ions detected in other samples to m/z values. Subsequently, an internal correction was performed for each sample to eliminate the sample position-dependent errors. For this correction, the relative errors of the m/z values determined for two reference MH^+ ions (peptide *C* and *M*) were used to determine the constants in a first-order equation. This equation was then used for an internal correction of the other externally determined m/z values in the same sample.

After the calibration procedure, we picked the peaks in the resulting 100 mass spectra using our basic peak picking approach I_a (minimal FWHM 0.07 Th, minimal signal-to-noise ratio 6, minimal intensity 400) and with the additional optimization step I_b, but no separation of overlapping peaks as in I_c in the previous section. Furthermore, we used the Centroid algorithm of the *flexAnalysis* software; once, II_a with parameters (minimal FWHM 0.07 Th, minimal signal-to-noise ratio 6, minimal intensity 400) similar to those used in I_a, and once we used a standard parameter set II_b (minimal FWHM 0.1 Th, minimal signal-to-noise ratio 10, minimal intensity 0) determined for MALDI data. Using the resulting peak lists, we computed the average relative error for each of the 19 known peptides in the 100 spectra to measure the accuracy of the different peak picking algorithms. Additionally, we determined the precision of each peak picker given by the average standard deviation of the relative error. The results are given in Table 9.3.

Since peptides *C* and *M* were used for the internal calibration procedure, their measured and calibrated values do always coincide with the theoretical m/z values. Our peak picking algorithm as well as the Bruker Centroid algorithm II_a achieved remarkable accuracies. Furthermore, our basic peak picking algorithm I_a yielded a slightly better average accuracy with 1.369 ppm than the Bruker algorithm II_a, which resulted in 1.935 ppm. However, the average precision of II_a was slightly better than the average precision of our algorithm. Using similar parameters for the Bruker peak picking method II_a our algorithm achieved comparable accuracy and precision values. In spite of that, the standard settings determined for MALDI spectra in II_b resulted in a clear worsening of accuracy and precision, which was mainly caused by measurements of the peptides *O*, *Q*, and *S*. The total number of peaks determined by II_a and II_b was restricted to 100 for each spectrum, whereas our method I_a (and I_d, respectively)

Table 9.3: *Evaluation of dataset* Peptide mix MALDI. *In the table, I_a denotes the results of our current peak picking algorithm and I_b represents the results of the current peak picking method along with the optimization of all peak parameters. Method II_a denotes the Centroid method with parameters similar to those used in our method, and the results in II_b are based on a standard parameter set determined for MALDI spectra.*

		average rel. err. [ppm] / standard deviation of rel. err. [ppm]			
peptide	**m_{theo} [Da]**	**I_a**	**I_b**	**II_a**	**II_b**
A	***757.399***	3.902 / 12.941	6.723 / 14.672	16.349 / 10.989	0.721 / 10.980
B	***1046.542***	0.334 / 6.898	2.614 / 9.347	0.387 / 4.578	0.272 / 4.553
C	***1296.685***	0.000 / 0.000	0.000 / 0.000	0.000 / 0.000	0.000 / 0.000
D	***1347.735***	2.056 / 4.082	1.743 / 5.388	1.755 / 3.121	1.758 / 3.122
E	***1363.730***	1.875 / 5.611	4.662 / 6.902	1.225 / 3.783	1.225 / 3.785
F	***1536.692***	2.614 / 6.647	2.565 / 7.719	2.011 / 4.409	2.038 / 4.412
G	***1570.677***	1.887 / 4.849	4.056 / 4.954	1.894 / 4.713	1.935 / 4.717
H	***1619.822***	2.501 / 5.519	1.279 / 6.497	3.343 / 4.283	3.288 / 4.278
I	***1635.817***	3.392 / 7.666	0.602 / 6.892	3.170 / 5.030	3.030 / 5.424
J	***1759.939***	0.313 / 5.804	0.796 / 5.454	0.049 / 4.225	0.006 / 4.219
K	***2093.086***	0.003 / 5.302	0.039 / 5.104	0.350 / 4.152	0.347 / 4.144
L	***2109.081***	0.084 / 5.058	0.238 / 4.557	0.845 / 4.331	0.828 / 4.299
M	***2465.198***	0.000 / 0.000	0.000 / 0.000	0.000 / 0.000	0.000 / 0.000
N	***2682.493***	0.420 / 4.175	2.056 / 3.591	0.560 / 2.369	0.561 / 2.347
O	***2698.487***	1.487 / 5.670	3.078 / 4.168	1.115 / 3.720	5.849 / 51.670
P	***2932.588***	0.544 / 7.174	3.569 / 4.893	0.518 / 5.205	0.551 / 5.199
Q	***2948.583***	1.859 / 6.992	4.739 / 6.503	0.118 / 5.709	52.437 / 124.066
R	***3147.471***	2.341 / 8.089	2.788 / 7.084	2.167 / 6.732	1.921 / 6.777
S	***3494.651***	0.393 / 11.987	6.350 / 9.719	0.902 / 10.280	26.578 / 82.540
	total	1.369 / 6.024	2.521 / 5.971	1.935 / 4.612	5.439 / 17.186

resulted in average in 185 peaks per spectrum.

The optional optimization step in I_b did not improve the accuracy of the detected peaks. We have seen in Section 9.2 that if the mass spectral peaks are well separated, the centroid position represents a more accurate estimate of the "true" m/z value than the peak position resulting from the non-linear optimization technique.
In addition to the high accuracy and precision of our approach the runtimes in Table 9.4 indicate its applicability to high-resolution mass spectra. On the 220 MB of data (100 spectra), our basic peak picking approach I_a took only 8.47 s (measured as absolute CPU time) on a PC with dual

3.2 GHz CPU. However, the optional optimization of all peak parameters requires 14.89 s. Note that we used different computers for the experiments and that runtime also depends on the number of peaks in the data, so that these numbers cannot be compared across experiments.

Table 9.4: *Total runtimes of the current peak picking algorithm on the 100 spectra of dataset* Peptide mix MALDI. *In the table, I_a denotes our basic peak picking algorithm, and I_d the basic peak picking algorithm together with the optimization of all peak parameters.*

	$\mathbf{I}_a$	$\mathbf{I}_b$
CPU time in seconds	8.47	14.89

Chapter 10

Discussion and conclusion

We have presented a wavelet-based peak picking technique suited for the application to the different kinds of mass spectrometric data arising in computational proteomics. In contrast to many established approaches to this problem, the algorithm presented here extracts all information that can be used for any kind of experimental setup. Besides an accurate m/z and FWHM value, our approach determines the two different quantity values of a peak: maximum intensity and total ion count. Furthermore, the curvature of each raw data peak is extracted by the fit of an analytical peak function. Our algorithm has been particularly designed to work well even on low-resolution data with strongly overlapping peaks. This is especially apparent when isotopic peaks, for example of charge two isotopic patterns, with poor separation arise in mass spectra (e.g., the LC-MS dataset discussed above). Here, the good performance of our algorithm can be attributed to two of its unique features: the ability to determine the position of a peak even if it overlaps heavily with another one, which is due to the use of the wavelet transform, and the optional non-linear optimization to determine the optimal peak parameters.

Applied to two real data sets a high-quality MALDI-TOF spectrum of a peptide mixture, our algorithm yields a high degree of accuracy and precision and compares very favorably with the algorithms supplied by the vendor of the mass spectrometers.
On the high-resolution MALDI spectra as well as on the low-resolution LC-MS data set, it achieves a fast runtime of only several seconds.

The results of most peak picking algorithms depend on meaningful parameter settings. In the current version of the *PeakPicker* tool, at least four parameters have to be adapted to the input data set. These essential parameters are the minimal expected FWHM value of a mass spectral peak, a minimal intensity of a peak, a minimal signal-to-noise value, as well as the scale for the continuous wavelet transform. As we have shown in Section 8.2.1, we can estimate a proper scale given the FWHM threshold. To facilitate the process of the parameter optimization, this process could be automatized in the next version of the *PeakPicker*. Assume we extracted a

representative number of peaks in one or multiple mass spectra with a default scale and the FWHM, signal-to-noise, and intensity threshold set to zero. Given the initial peak set we can compute a histogram of the FWHM values and determine a proper FWHM, signal-to-noise, and intensity parameter with respect to it. To emphasize the separation of noisy and "true" mass spectral peaks in the histogram we may weight each FWHM value by the signal-to-noise, correlation, and intensity value.

The peak picking algorithm is implemented in the freely available OpenMS framework. Based on the peak picking classes in OpenMS we also implemented the easy-to-use TOPP application `PeakPicker`.

Part III

Map alignment

Chapter 11

Computational geometry preliminaries

Computational geometry deals with the algorithmic aspects of geometric problems. Typical problems in computational geometry are, for example, the intersection of line segments or the point locations, which are motivated by the prevalent use of geometric objects in computer graphics and computer aided design. For a deeper insight into this research area we recommend Chapter 8 of Mehlhorn [1984]. In the following we will shortly describe two fundamental data structures in computational geometry that can be used to solve the closest point problem. We will also present the k-nearest neighbor search based on the definitions in Mehlhorn [1984] and Mehlhorn and Näher [1999].

11.1 Voronoi diagram

The *Voronoi diagram* for a two dimensional point set $P = \{p_1, \ldots, p_n\}$ is a partition of the plane into $|P|$ polygonal regions, one for each point $p_i \in P$. Given a metric $d : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$, the *Voronoi region of a point* p_i, defined as

$$VR(p_i) := \{y \in \mathbb{R}^2 : d(p_i, y) \leq d(p_j, y), \forall j \in \{1, \ldots, n\}\},$$

consists of all points which are closer to p_i than to any other point in P. The Voronoi diagram $VD(P)$ of P is then defined as the union of the Voronoi regions $VR(p_i)$ for $1 \leq i \leq n$. In the Voronoi diagram holds, that for each vertex v in $VD(P)$, there are at least three points $p_i, p_j, p_k \in P$ such that $d(v, p_i) = d(v, p_j) = d(v, p_k)$. Aurenhammer [1991] provides an extensive survey of Voronoi diagrams and their applications. In Figure 11.1 the dashed lines show a Voronoi diagram of ten points.

11.2 Delaunay triangulation

Consider the set of all triangles formed by the points in a point set P such that their circumcircle is empty. The set of edges of these triangles gives the *Delaunay triangulation* $D(P)$ of P. The solid lines in Figure 11.1 show the Delaunay triangulation of ten points and the green circle illustrates the "empty circle" property by means of one triangle.

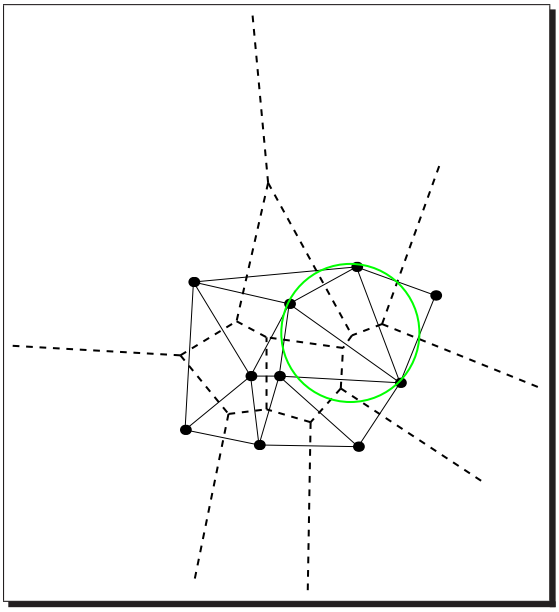

Figure 11.1: *The dotted lines show the Voronoi diagram; the solid lines show the Delaunay triangulation of the points. The green circle illustrates the "empty circle" property of the Delaunay triangulation by means of one triangle.*

The planar Voronoi diagram and the Delaunay triangulation are duals in a graph-theoretical sense. Given a Voronoi diagram it is straightforward to find those triangles. If one connects each p_i to all points in neighboring cells, then the resulting triangulation fulfills the above mentioned conditions.

11.3 k-nearest neighbors

Given a point set $P = \{p_1, \ldots, p_n\}$ and a metric $d : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$, the *k-nearest neighbors* of a point $p_i \in P$ are the k points $\{p_{j_1}, \ldots, p_{j_k}\}$, which have minimum distances to p_i with $d(p_{j_1}, p_i) \leq d(p_{j_2}, p_i) \leq \ldots \leq d(p_{j_k}, p_i) \leq d(p_l, p_i)$ and $l \in \{1, \ldots, n\} \setminus \{j_1, \ldots, j_k, i\}$.

To efficiently answer $k-$*nearest* neighbor searches, Voronoi diagrams are optimal in theory.

In practice, other data structures that are less efficient in theory still seem to perform quite well. Delaunay triangulation turns out to be a very powerful data structure for storing dynamic sets of points under range and nearest neighbor queries [Mehlhorn and Näher, 1999].

Chapter 12

Introduction to LC-MS map alignment

The quantitative information in an LC-MS map can be used in numerous applications. The spectrum ranges from additive series in analytical chemistry [Gröpl et al., 2005], over analysis of time series in expression experiments [Bisle et al., 2006; Niittylä et al., 2007], to applications in clinical diagnostics [Vissers et al., 2007], in which we want to find statistically significant markers for detecting certain disease states. All these applications have in common that the same peptides in different measurements have to be related to each other. For example, Myoglobin, a low molecular mass heme protein, is a biochemical marker for myocardial necrosis associated with myocardial infarction. To quantify the concentration of Myoglobin in a test blood sample, several measurements are made with known amounts of spiked Myoglobin. The change in ion counts for Myoglobin over these measurements allows for the estimation of the initial Myoglobin concentration [Gröpl et al., 2005]. The underlying assumption is that the measured m/z and retention time of a peptide stay roughly constant. As with every laboratory experiment, this only holds true to a certain extent.

In particular, the retention time often shows large shifts and possibly distortions when different runs are compared, but the m/z dimension might also show (typically smaller) distortions. The overall change in RT and m/z is called *warp*. Leaks, pump malfunctions, and changes in column temperature or mobile phase result in distorted elution patterns and can even cause changes in the elution order of peptides. For example, in one measurement peptides A, B, and C may elute in the order $A-B-C$, however, in the second measurement they elute in order $C-B-A$. This scenario is not unlikely if the retention times of A, B, and C are similar [Snyder and Dolan, 2007]. The shift in RT makes the assignment of similar peptides difficult since the relative shift of two maps to each other is not known in advance. But it is crucial to correct for those shifts and to consider time order changes. Otherwise it is hard or even impossible to find for a peptide in the first map the corresponding partner in the second map. The correction of the shift in RT and m/z is called *dewarping* according to the time warping problem of [Sakoe and Chiba, 1976] in speech processing. The advent of high-throughput quantitative proteomics made an

efficient solution to this problem an important task.

Several approaches have been presented in the literature and we will give an overview in Section 13.2. In Chapter 14, we will lay out our own solution in detail. In the following, we will first introduce a general distance measure for LC-MS maps. Based on this measure, we will develop a problem definition for multiple raw and feature map alignment.

12.1 LC-MS map alignment problems

The estimation of suitable mappings between multiple LC-MS maps can be either positioned at the beginning or at the end of a comparative proteomics data analysis pipeline. Both alternatives have their advantages and disadvantages. The comparison of raw maps places the correction of the RT and m/z dimensions at the beginning of an analysis pipeline, whereas the comparison of feature maps positions the estimation of a suitable mapping at the end of the pipeline prior to the statistical analysis. Feature maps have a much smaller data amount than raw maps and therefore allow for much faster dewarping algorithms. However, signal preprocessing, peak picking, and feature finding algorithms may also introduce errors, and thereby the quality of the feature maps strongly depends on the reliability of these algorithms. The correction of RT and m/z dimensions on the raw data level enables the search for differentially expressed peptides directly in the raw maps using multiway data analysis methods (e.g., PARAFAC [Bro, 1997]). These approaches avoid errors introduced by peak picking and feature finding algorithms, but they tend to have high runtime and problems with time order changes. Our solution, however, works equally well on both raw and feature maps by transforming the estimation of a suitable mapping between LC-MS maps into a well-known problem in computational geometry. We consider the elements of an LC-MS raw or feature map as two-dimensional point sets, given by the RT and the m/z positions of the elements. This reduces to the *point pattern matching problem*: Given two finite point sets M (the *model*) and S (the *scene*) we want to know how much they resemble each other [Alt and Guibas, 1996]. In the point matching problem the point sets underwent a certain transformation, which we want to recover. This transformation should map the corresponding points of the two sets close together; by this, it discovers the correspondences between M and S.

The *point pattern matching problem* can be divided into the *exact point pattern matching problem (EPMP)* and the *approximative point pattern matching problem (APMP)*. The EPMP assumes two point sets of equal size and searches for a transformation that maps the points of one set exactly onto the points of the other set. Since the RT and m/z dimensions of an LC-MS map are afflicted by measurement errors, and the positions of corresponding elements in two LC-MS maps will hardly ever be identical, the APMP is better suited to our problem: Given two point sets M and S search for that transformation that maps each point of M close to another point in S.

Our LC-MS map alignment problem constitutes a special case of the APMP: the *partial APMP*. Consider two LC-MS maps, where the 2D positions of the elements (which can be raw data points or features) in the two maps define two 2D point sets M and S in the plane. In the partial APMP M and S share only a fraction of common points. This is a realistic assumption for LC-MS maps, where even two LC-MS maps resulting from repeated measurements do not necessarily have identical elements.
To solve the partial APMP, we have to find a transformation $T : M \rightarrow \hat{M}$ that maps M onto S such that the dewarped point sets $\hat{M}$ and S become most similar. In our case, most similar means that common elements in S and dewarped $\hat{M}$ have nearby positions. Determining pose and correspondence between two sets of points in space, is, in other words, to transform one point set so that it best matches another point set in whole or in part. This is a fundamental problem in computer vision and a number of algorithms were developed to solve it [Alt and Guibas, 1996; Veltkamp, 2001]. Most point pattern matching algorithms are not general and are designed for a specific similarity measure $s : M \times M \rightarrow \mathbb{R}$ between two point sets. These approaches are defined by a similarity measure, a transformation $T : M \rightarrow \hat{M}$, and an optimization strategy to determine the parameters of the transformation maximizing the similarity measure. Four more general approaches, which are used to solve the partial APMP and could also be used to dewarp LC-MS maps, are described in Section 13.1. Accordingly, we can either solve the LC-MS map alignment by the adaptation of one of the general approaches mentioned in Section 13.1, or by an algorithm that optimizes a certain similarity measure. Our own contribution in Chapter 14 builds on both ideas and proposes a multiple LC-MS map alignment algorithm using an adapted pose clustering approach, and suggests the implementation and application of a specific distance function for LC-MS maps.
In the following, we will develop the mentioned distance function for LC-MS maps, which can be used for LC-MS raw as well as feature maps, because it depends only on the 2D positions of the elements and their intensity values.

12.2 A distance function *dsim* for LC-MS maps

At first we consider a similarity of LC-MS/MS maps. In LC-MS/MS maps some of the elements are annotated with reliable peptide identifications and thereby a part of the correspondence between the maps is already given. These corresponding elements give information about the extent of the distortions in both the RT and the m/z dimension and can be used to discover the correspondence of the remaining elements without annotations. Corresponding elements in two maps with similar 2D positions point at comparable RT and m/z dimensions, whereas common elements with different positions indicate a considerable shift in RT and m/z. The more the 2D positions of common elements vary, the greater the distance between the maps and the more dissimilar the maps are. Therefore, we measure the similarity using the distance of corresponding elements in the Euclidean space $\mathbb{R}^2$. The RT dimension is in general more

distorted than the m/z dimension, hence a weighted Euclidean metric should be used instead of the standard Euclidean distance. Instead of evaluating the distance of corresponding elements, we can also evaluate the similarity of elements with similar coordinates. If the positions of common elements vary significantly between different maps, an element's nearest neighbor in the other map will not have the same annotation. Instead of the sum of distances between corresponding elements, we can also count the number of corresponding elements that have similar coordinates and are nearest neighbors. This approach requires a one-to-one assignment of elements in two maps, that we will give in the following definition.

Definition 12.2.1: Given two LC-MS maps $M := \{m_1, \ldots, m_k\}$ and $S := \{s_1, \ldots, s_l\}$ and an $\varepsilon > 0$. The matching function $match : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{B}$ with $\mathbb{B} = \{0,1\}$ is defined as follows: Two elements $m_i \in M$ and $s_j \in S$ are matched if their positions lie within an ε-environment in a weighted Euclidean metric $d : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$, and s_j is nearest neighbor of m_i and vice versa:

$$match(m_i, s_j) := \begin{cases} 1, & d(m_i, s_j) < \varepsilon \text{ and} \\ & \forall m_r \in M \setminus \{m_i\}, s_t \in S \setminus \{s_j\} : \\ & d(m_i, s_j) \leq d(m_i, s_t) \text{ and} \\ & d(m_i, s_j) \leq d(m_r, s_j) \\ 0, & \text{otherwise} \end{cases}$$

For the annotated elements we could verify each match using the identification of the elements. The total number of matched elements with identical identifications indicates the similarity of two LC-MS/MS maps.
The match function allows for an assignment of unannotated elements in two LC-MS maps and thereby can also be used in a similarity measure for LC-MS maps. Although the lack of annotations prevents the verification of the matching, we can use the intensity of the elements as an additional similarity term instead. A matching of elements with similar intensities should be rewarded, whereas a matching of two elements with extremely different intensities should be penalized. The evaluation of the matching using the elements' ion counts is a sensible assumption if the majority of peptides is not differentially expressed, which is usually the case. It should be noted that the comparison of intensities in different maps requires an intensity normalization of the maps [Katajamaa et al., 2006; Radulovic et al., 2004; Wang et al., 2007].
The matching function in Definition 12.2.1 indicates the similarity of matched elements' positions, and the ion counts of two feature maps. Hence, we are now able to define a distance function or dissimilarity measure for LC-MS maps:

Definition 12.2.2: Given LC-MS maps $M := \{m_1, \ldots, m_k\}$ and $S := \{s_1, \ldots, s_l\}$ and $\varepsilon > 0$. Furthermore, $(\mathrm{RT}(m_i), \mathrm{m/z}(m_i))$ is the 2D position of the element m_i and $int(m_i)$ its ion count. The distance or dissimilarity $dsim : M \times S \to \mathbb{R}$ of M and S is given by:

$$dsim(M,S) := \max\{k,l\} - \sum_{i=1}^{k}\sum_{j=1}^{l} match(m_i, s_j) \frac{|d(m_i, s_j) - \varepsilon|}{\varepsilon} \frac{\min\{int(m_i), int(s_j)\}}{\max\{int(m_i), int(s_j)\}}.$$

Given two maps $M := \{m_1, \ldots, m_k\}$ and $S := \{s_1, \ldots, s_l\}$, the codomain of the distance measure is $[0, \ldots, \max\{k, l\}]$.
For all maps M, S and X *dsim* satisfies the following conditions

- $dsim(M,S) \geq 0$ (non-negativity).
- $dsim(M,S) = 0,$ if and only if $M = S$ (identity).
- $c(dsim(M,X) + dsim(X,S)) \geq dsim(M,S)$ for some constant $c \geq 1$ (relaxed triangle inequality).
- $dsim(M,S) = dsim(S,M)$ (symmetry).

Similarity measures for partial matching, giving a small distance $dsim(M,S)$ if a part of M matches a part of S, in general do not obey the triangle inequality and it therefore makes sense to formulate a weaker form, the relaxed triangle inequality [Veltkamp, 2001]. Another useful property of $dsim(M,S)$ is the symmetry, which guarantees that the order in which the maps are compared does not matter.

As an explanatory example, Figure 12.1 shows two feature maps, "feature map 1" and "feature map 2", which share a fraction of common features. "Feature map 1" depicts data from a real measurement. 80% of the data points were copied to "feature map 2" after their RT positions had been warped by an affine transformation $T := 1.1x + 30$. Additionally, random points were added to the bounding box. Since the RT dimension is usually more distorted than the m/z dimension we use a weighted Euclidean metric given by $d(m,s) = \sqrt{w_1^2(m_{\mathrm{RT}} - s_{\mathrm{RT}})^2 + w_2^2(m_{\mathrm{m/z}} - s_{\mathrm{m/z}})^2}$ with $w_1 := 1$ and $w_2 := 10$. Furthermore, we allow for an error of 22 s and 0.2 Th and yield $\varepsilon \approx 30$. Due to the shift, the distance between the two maps is relatively large and shows up in the maximum *dsim* value of 195. Even with $\varepsilon := 100$ (corresponding to an error of 0.2 Th and 98 s) the *dsim* value of 190 indicates a large dissimilarity of the maps. In Figure 12.1 on the right hand side "feature map 1" and the dewarped "feature map 2" are shown. The common 80% of the features have now similar positions and the *dsim* value of 30 indicates relatively similar maps.

This general distance function can be used for every type of LC and MS based experiment. Furthermore, it is also independent of the processing state of the maps, because it uses only the 2D positions and intensities of the elements. We will now use *dsim* to define the multiple LC-MS raw and feature map alignment problem.

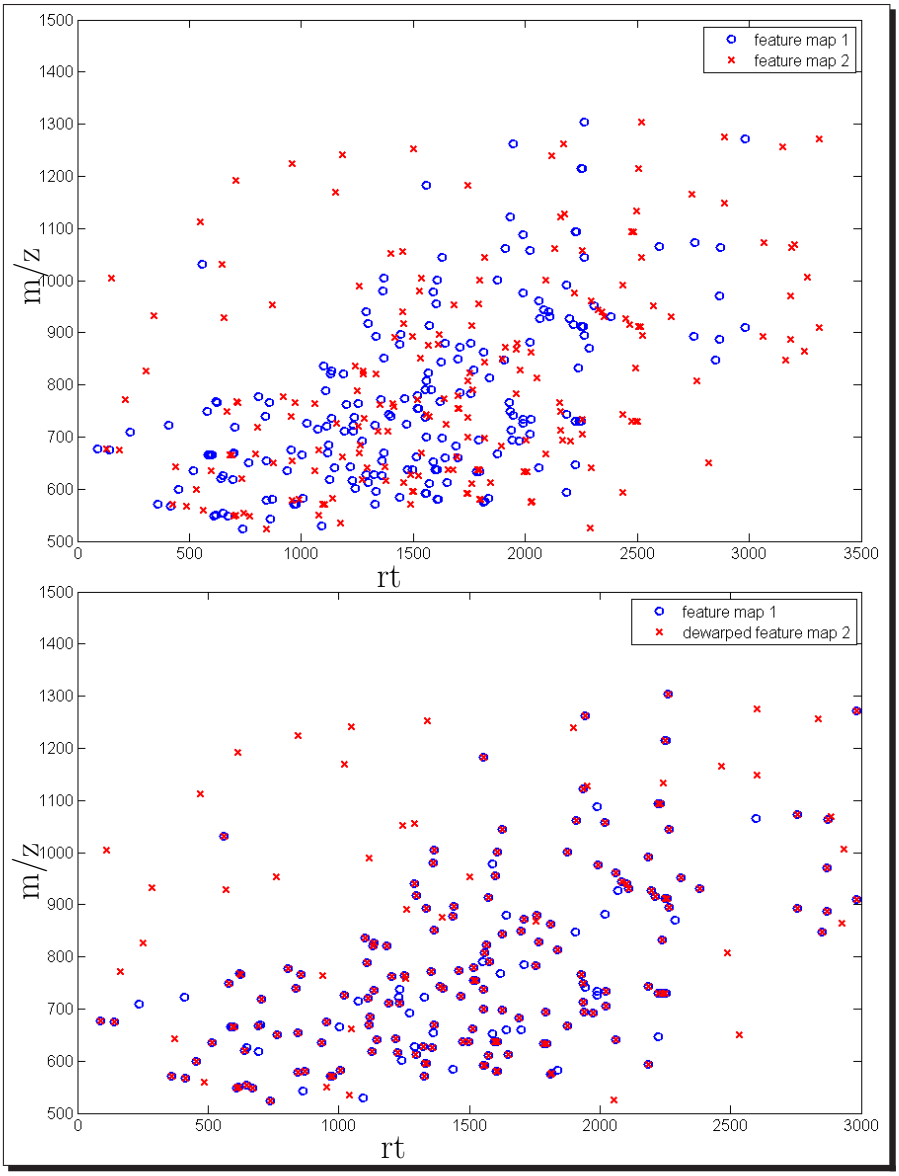

Figure 12.1: *Top: Two LC-MS feature maps are shown. "Feature map 1" as well as "feature map 2" contain 195 features. The two feature maps share 156 common features, but the RT positions of these features are shifted in "feature map 2" by an affine transformation* $T := 1.1x + 30$*. The dsim value of the two dissimilar feature maps is* 195 *using* $\varepsilon = 30$ *(allowing for an error of* 0.2 *Th in m/z and* 22 *s in RT) and even with* $\varepsilon = 100$ *(allowing for an error of* 0.2 *Th in m/z and* 98 *s in RT) the two maps have a large distance of* 190*. Bottom: "feature map 1" and the dewarped "feature map 2" are shown. The dsim value of these two feature maps is only* 30 *for both* $\varepsilon = 30$ *and* $\varepsilon = 100$*.*

12.3 Multiple raw and feature map alignment problem

To enable the comparison of raw or feature maps, we have to correct for the shift in RT and m/z, such that corresponding elements get similar 2D positions. The optimal transformation would already solve the raw map alignment problem, because the assignment of corresponding elements is directly done by the following multiway data analysis methods [Bro, 1997]. However, when dealing with feature maps, the assignment of corresponding features is a requirement for the following comparative analysis.

The retention time warp as well as the warp of the m/z dimension are continuous functions but a detailed description of their shape has not been specified in the literature yet. The shift in the m/z dimension can be defined as a monotonically increasing function and the m/z positions of corresponding elements in two different maps are typically very similar. However, the type of function representing the distortion in RT is more difficult to characterize. Due to possible changes in the elution order of peptides, the monotonicity cannot be stringently assumed. Jaitly et al. [2006] propose that the flow rate variability from experiment to experiment introduces a global linear trend, whereas gradient noise, or to some extent other types of variations between analyses, e.g., temperature changes, variations in solvent composition, or changes to the stationary phase may introduce local distortions. Any computational approach to the multiple LC-MS raw and feature map problem should overcome the inherent variability in the time and m/z axis and transform all maps onto a comparable coordinate system.

We define the *Multiple LC-MS Raw Map Alignment Problem (MRMAP)* as follows:

Multiple LC-MS Raw Map Alignment Problem:
Given k LC-MS raw maps $M_1, \ldots, M_k$ of size $l_1, \ldots, l_k$.
Find k continuous transformations $T_1, \ldots, T_k$ with $T_i : M_i \rightarrow \hat{M}_i$ ($i \in \{1, \ldots, k\}$), and $T_1 := \mathrm{id}$,
such that the sum of pairwise distances $\sum_{i=1}^{k} \sum_{j=1}^{k} dsim(\hat{M}_i, \hat{M}_j)$
between the dewarped maps $\hat{M}_1, \ldots, \hat{M}_k$ is minimal.

A feature map alignment should not only correct the inherent variability in the time and m/z axis, but also assign corresponding features to allow for the subsequent statistical comparative analysis.
The correspondence information of all detected peptides in multiple maps is stored in a so-called *consensus map*. A consensus map consists of a number of *consensus features*, each of which groups together corresponding elements across multiple maps. All features constituting a consensus feature should represent the same charge state of an ionized peptide. Each feature should be assigned to only one consensus feature and each consensus feature should contain at most one feature of each map.
Given k maps a consensus feature may consist of a single feature, if no other map contains the same charge state of the ionized peptide, or represent up to k features of different maps.

Definition 12.3.1: Let $M_1, \ldots, M_k$ be k LC-MS maps of size $l_1, \ldots, l_k$ and f_{ij} the j-th feature of map i

- A tuple $c := (\mathrm{RT}(c), \mathrm{m/z}(c), int(c), \{f_{st} : s \in \{1, \ldots, k\} \text{ and } t \in \{1, \ldots, l_s\}\})$ is called *consensus feature*, if it fulfills the following properties
  - If $f_{jk} \in c_i$ then $f_{jk} \notin c_r$ with $i \neq r$ (uniqueness of features).
  - If $f_{jk} \in c$ and $f_{st} \in c$ then $j \neq s$ (uniqueness of consensus features).
- The minimum and maximum RT position of all combined elements along with the minimum and maximum m/z position define the *bounding box* of a consensus feature.
- The set of all consensus features defines a *consensus map* $C := \{c_1, \ldots, c_n\}$ of the maps $M_1, \ldots, M_k$ with $(\max\{l_1, \ldots, l_k\} \leq n \leq \sum_{i=1}^{k} l_i)$.

A consensus map C represents a partition of the set M_{all}, which contains the elements of all maps $M_1, \ldots, M_k$. The consensus features are the disjoint subsets in M_{all} and each consensus feature may contain at most one feature of each map. In the alignment of feature maps we want to create meaningful partitions and to avoid consensus maps where each feature represents a singleton consensus feature. Corresponding features should be grouped in only one consensus feature instead of being split in multiple subsets. Therefore, we define a convex quality measure, the *size*, for a consensus feature.

Definition 12.3.2: The *size* of a consensus feature $c := (\mathrm{RT}(c), \mathrm{m/z}(c), int(c), \{f_1, \ldots, f_n\})$ is given by $size(c) := \binom{n}{2}$.

Figure 12.2 illustrates the idea of size. The grouping of all five elements to only one consensus feature leads to a size of ten, whereas the two consensus features of size three and two achieve sizes of three and one and lead to a total size of four.

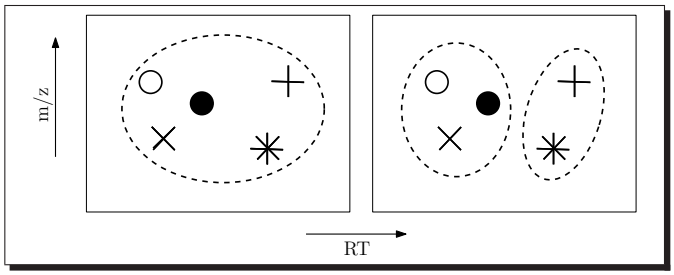

Figure 12.2: *The consensus feature on the left hand side has a size of ten. Composing the five elements to two consensus features of size three and two leads to sizes of three and one.*

We use the distance function *dsim* and the total size of consensus maps to define the *multiple feature map alignment problem (MFMAP)*:

Multiple LC-MS Feature Map Alignment Problem:

Given k LC-MS raw maps $M_1, \ldots, M_k$ of size $l_1, \ldots, l_k$.

Find k continuous transformations $T_1, \ldots, T_k$ with $T_i : M_i \rightarrow \hat{M}_i$ ($i \in \{1, \ldots, k\}$), and $T_1 := \mathrm{id}$, such that the sum of pairwise distances $\sum_{i=1}^{k} \sum_{j=1}^{k} dsim(\hat{M}_i, \hat{M}_j)$ between the dewarped maps $\hat{M}_1, \ldots, \hat{M}_k$ is minimal and the consensus map $C = \{c_1, \ldots, c_n\}$ has a maximum total size $\sum_{i=1}^{n} \mathrm{size}(c_i)$.

Chapter 13

Related work

Both the multiple raw map and the multiple feature map alignment problem can be generalized to a 2D point pattern matching problem. This problem is common in computer vision and many other fields [Brown, 1992]. In the following Section 13.1 we will briefly introduce the point pattern matching problem and describe some general approaches to solve its.
Section 13.2 gives an overview of existing algorithms for the alignment of multiple raw or feature maps. Several of these algorithms will be performance evaluated in detail in Section 15.3.

13.1 General approaches for point pattern matching problems

Many algorithms match two point sets with respect to a predefined similarity measure; Veltkamp [2001] provides a good survey of matching algorithms along with a description of the used similarity measures. In the following, we will describe the basic ideas of four generic popular approaches for the partial matching problem, which are *generalized Hough Transformation*, *pose clustering* [Ballard, 1981; Stockman et al., 1982; Olson, 1994], *geometric hashing* [Wolfson and Rigoutsos, 1997], and *alignment* [Huttenlocher and Ullman, 1987]. These methods belong to the class of *voting schemes* and offer appropriate solutions for our LC-MS alignment problem. They serve as methods for pattern recognition, where the *pose*—the position and orientation (with respect to the image coordinate system) of a given shape is searched in an image. Given two point maps M and S, if we consider M as the *model* or shape, which we want to detect in the *scene* or image S, our partial matching problem looks like a pattern recognition problem. The question to answer is "Is there a transformed subset of M that matches a subset of S?".

We will explain the basic ideas of pose clustering, geometric hashing, and alignment on a simple example given two sets of points $M := \{m_1, \ldots, m_k\}$ and $S := \{s_1, \ldots, s_l\}$, which are

related by a general affine transformation $T_\alpha(x) := Ax + t$ [Brown, 1992] with parameters $\alpha := \{A, t\}$ with $A \in \mathbb{R}^{2\times 2}$, $t \in \mathbb{R}^2$. The scaling matrix A, and the translation vector t are defined by

$$A := \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \in \mathbb{R}^{2\times 2} \text{ and } t := \begin{pmatrix} t_1 \\ t_2 \end{pmatrix} \in \mathbb{R}^2.$$

Given M and S, we want to detect a warped version of $T_\alpha(M)$ in the image S.

Generalized Hough Transformation. Ballard [1981] laid the foundation of methods to detect arbitrary 2D shapes undergoing transformations such as translation, scaling, and rotation by the generalization of the Hough Transform. The generalized Hough Transformation (GHT) is a brute-force technique where a parametric equation of the shape is no longer required. The shape can be of any complex form and is only described by the orientation of the shape points, e.g., the gradient in an edge representation, along with the orientation of the points relative to a given shape's reference point. In a so-called R-generation phase, this information is stored in a hash table R, with the gradient orientation of the points as the key values. In the following object detection phase, the position of the shape in an unknown image can be determined using the R-table. Therefore, each point x in the image is considered as a point of the shape. Using the gradient orientation of x, the hypothetical reference position can be determined with the R-table. Each possible reference point position x_r, y_r is stored in a 2D accumulator array $A(x_r, y_r)$ and a maximum will occur at the reference point in A where the shape exists in the image. The complexity of the GHT is $O(kl)$. To allow for the search for an affine transformed version of M in S, the R-table as well as the accumulator array have to be expanded by 6 dimensions $a_{11}, a_{12}, a_{21}, a_{22}, t_x, t_y$. This leads to a large increase in runtime to $O(klAT)$ where A is the number of discrete scaling matrices and T the number of discrete translation vectors. In order to ensure precise transform parameters, small intervals should be used, but GHT quickly becomes infeasible.

Pose clustering. Pose clustering is a specialized form of GHT. In contrast to GHT, a pose clustering method does not compute all possible transformed forms of a shape and compares them to an image, but it computes only those transformations that correspond to hypothesized matches between shape points and image points. To solve for the six parameters of an affine transformation, three shape points m_1, m_2, m_3 (each given by its two-dimensional position $m_i := (m_{i,1}, m_{i,2})$) and three image points s_1, s_2, s_3 are needed. Using the system of six linear equations $s_{i,j} := a_{i,1}m_{j,1} + a_{i,2}m_{j,2} + t_i$ the parameters that map the m_i onto s_i can be uniquely determined.
Two point sets $M := \{m_1, \ldots, m_k\}$ and $S := \{s_1, \ldots, s_l\}$ yield $6\binom{k}{3}\binom{l}{3}$ distinct matching triples along with just the same number of hypothesized poses of the shape. Matching corresponding points onto each other yields the correct transformation, which would indicate the position and scaling of the shape in the image. In theory the correct matches will yield transformations

close to the correct pose of the shape in the image. Correct matches and thereby the correct pose appear $\binom{k}{3}$ times if each point of M can be matched onto a point in S. In practice, due to localization errors in detected features the estimates are not exactly correct but the cluster in the parameter space should still be easy to detect. Most pose clustering algorithms find clusters by histogramming the poses in the multidimensional transformation space. Searching for an affine transformation in this method, each pose is represented by a single point in the 6D pose space. The pose space is discretized into bins and the poses are histogrammed in these bins to find large clusters. Clustering techniques are more precise than histogramming, but in most cases they lead to unacceptably high runtime [Olson, 1997]. Even in case of histogramming techniques, an accurate pose clustering results in an immense pose space for the fineness of discretization and the runtime is $O(k^3 l^3)$.
Stockman et al. [1982] reduce the pose space by a coarse-to-fine clustering where the pose space is quantized in a coarse manner and the large clusters found in this quantization are then histogrammed in a more finely quantized pose space. A problem that can arise with this technique is that the largest clusters in the first clustering step do not necessarily correspond to the largest clusters in the entire pose space. Grimson and Huttenlocher [1990] show that for cluttered images, an extremely large number of bins would need to be examined due to saturation of the coarse histogram.
Olson [1998] shows a considerable improvement in both speed and accuracy of object recognition with his approach. He divides the recognition problem into smaller subproblems, whereby randomization is used to limit the number of subproblems. Furthermore, he introduces a simple grouping mechanism that locates pairs of points that are likely to belong to the same object and matches only these possible matching points. He achieves a runtime of $O(kl^2)$ and space complexity of $O(kl)$.

Geometric Hashing. Geometric Hashing [Wolfson and Rigoutsos, 1997] is an indexing technique and proceeds in two steps comparable to GHT. In a first step, the preprocessing phase, possible forms of the shapes are extracted. In contrast to GHT, Geometric Hashing does not work with all transformations, but only with that subspace of transformations constrained by the data points. Therefore, each triplet (m_1, m_2, m_3) of shape points is used to construct an orthonormal basis and the point positions $\tilde{m}_i$ of all other shape points m_i are calculated in respect to the new basis. The basis triple (m_1, m_2, m_3) is inserted at each quantized value $\tilde{m}_i^q$ in a hash table. In the second step, the recognition phase, the preprocessed forms of the shape are recognized in the image. Each triplet (s_1, s_2, s_3) of image points is used to compute an orthonormal basis and all residual image points s_i are transformed relative to the new coordinate frame. The transformed and quantized values $\tilde{s}_i^q$ are used to determine whether the image point matches any shape point in a certain orthonormal basis. Histogramming of the hash-table entries along with the actual image basis then discovers the basis of the shape and the image achieving the best matching. This basis can be used to find all corresponding point pairs and

recover an affine transformation that results in the best least square match between the point pairs. Geometric hashing is a popular method due to the recognition of multiple shapes in an image even in case of partial occlusion but it requires a large memory to store the hash table. The runtime of the preprocessing phase is $O(k^4)$ and for recognition is in worst case $O(l^4)$, but in typical real-world applications, the runtime often is linear.

Alignment. The alignment approach is similar to the pose clustering approach. It iteratively computes a certain affine transformation for each triple (m_i, m_j, m_k) of shape points and each triple (s_l, s_m, s_n) of image points. In contrast to pose clustering, the transformation parameters are not voted for, but the shape is mapped into the image by applying the transformation to the shape points. This allows the search for corresponding points of the shape in the image and the total number of common elements is used to evaluate the transformation parameters. If the validation of a transformation exceeds a certain threshold the iteration is finished. The alignment method achieves a runtime of $O(k^4 l^3)$ consisting of $O(k^3 l^3)$ for the determination of all transformations and $O(k)$ for the verification phase of each transformation. In case of noisy image point sets Grimson et al. [1991] showed that the probability of false matches using the alignment approach is substantially smaller than for the Geometric Hashing approach, but the high runtime of this method makes it inapplicable for large point sets.

13.2 Multiple LC-MS map alignment algorithms

The computational challenges in LC-MS map alignment have recently moved into the focus of the bioinformatics community and several alignment algorithms have already been developed. In the following we will review the existing algorithms for multiple raw LC-MS map alignment [Bylund et al., 2002; Prakash et al., 2006; Prince and Marcotte, 2006; Listgarten et al., 2007; Listgarten and Emili, 2005] and multiple feature LC-MS map alignment [Radulovic et al., 2004; Katajamaa et al., 2005; Li et al., 2005; Zhang et al., 2005; Jaitly et al., 2006; Bellew et al., 2006; Smith et al., 2006; Wang et al., 2007].

Multiple LC-MS raw map alignment algorithms. Many of the algorithms for raw LC-MS map alignment [Bylund et al., 2002; Prakash et al., 2006; Prince and Marcotte, 2006] are based on two standard non-parametric approaches, namely dynamic time warping (DTW) [Sakoe and Chiba, 1976] and correlation optimized warping (COW) [Nielsen et al., 1998]. Both approaches align time series by stretching or shrinking the time axis. DTW has its origin in speech processing and computes a non-linear mapping of one signal onto another by minimizing the distances between time series. COW is comparable to DTW, but it computes a piecewise linear transformation by dividing the time series into segments and then performing a linear warp within each segment to optimize overlap while constraining segment boundaries.

The parameters for the best linear transformation are determined by maximizing the sum of correlation coefficients between data segments in pairs of samples. Both techniques appeared first in the alignment of chromatograms [Tomasi et al., 2004] and were afterward extended to the case of two-dimensional LC-MS data [Bylund et al., 2002; Prakash et al., 2006; Prince and Marcotte, 2006].
The approach of Bylund et al. [2002] is based on the idea of the traditional COW algorithm. The pairwise alignment of two LC-MS maps is determined by using representative subsets of extracted ion chromatograms of the complete maps (typically taken from the middle of the two maps). To allow for the largest time shift at the end of the chromatogram and the rejection of end portions in both maps which are not related, Bylund et al. used variable segment boundaries. Furthermore, Bylund et al. show that the sum of correlation coefficients as well as the sum of covariance coefficients are sensible scoring functions, which yield the best set of segment boundaries and therefore the optimal set of linear transformation by maximizing the total score via Dynamic Programming. The evaluation of the method shows the necessity of an alignment. Bylund et al. compares the amount of variance in base raw chromatograms explained by the two principal components determined by PCA, which was 70% before alignment and 98% afterward. Similarly, explained variance went from 60% to 97% with a seven-component parallel factor analysis (PARAFAC [Bro, 1997]—a generalization of PCA to three-way data), indicating a reduction in the major sources of sample variation.

Prakash et al. [2006] and Prince and Marcotte [2006] describe an extension of DTW and differ mainly in the similarity function they maximize. Prakash et al. [2006] introduce a score based on a normalized dot product of the mass spectra to lower the influence of noise peaks. The fuzzy dot product is based on the similarity measure proposed by Stein and Scott [1994], which exploits the mass resolution. To avoid high similarity scores for noisy spectra, the scoring function is expanded by an additional term. The maximal score is determined by a global alignment using a modified version of the Needleman-Wunsch algorithm [Needleman and Wunsch, 1970] and the optimal path, the so-called *signal map*, is the mapping of mass spectra in the two experiments that lead to the maximal score.

Prince and Marcotte [2006] verify the applicability of DTW for the alignment of LC-MS raw data and, in a comprehensive study, show that the best scoring function for the similarity of MS spectra besides covariance, dot product, and Euclidean distance, is the Pearson correlation coefficient. Furthermore, the penalization of gaps should prevent the stray from the optimal warping path, which occurs without any gap penalty [Prakash et al., 2006]. Prince and Marcotte introduce a bijective one-to-one spectra mapping by interpolation of the warping path yield during DTW.

Listgarten and Emili [2005] propose a Continuous Profile Model (CPM) for the alignment of multiple raw LC-MS maps using their total ion chromatograms (TICs). Each observed TIC or time series represents a noisy transformation of a canonical time series, the *latent trace*. The time points of the latent trace are a series of hidden states in a HMM, which are augmented by scale states that allow for intensity scaling. Due to mapping both in time and scale states the alignment procedure maps not only all time points in the TICs onto hidden states in the HMM, but also normalizes the intensities at the same time. The latent trace is determined by unsupervised learning with a Dynamic-Programming-based Expectation-Maximization algorithm. After the training phase, the model is used for the simultaneous alignment of multiple TICs. The proposed time consuming HMM-based alignment is reduced to TICs of repeated measurements.
In Listgarten et al. [2007] the CPM model is expanded by the m/z dimension. Instead of taking the total ion count at each time point of an LC-MS map into account, the model now uses the intensity of four m/z bins at each time point. The authors note that a greater number of bins would increase the runtime whereas too few m/z bins would result in a loss of quality of the alignment algorithm. The normalization is no longer regulated by scaling states, but performed by adding a new parameter vector to the model to speed up the runtime. Although Listgarten et al. declare that the alignment algorithm is no longer restricted to replicated data sets, a high similarity of the aligned samples is assumed and the algorithm is evaluated on data sets that differ in only three peptides.

In general, raw map alignment methods tend to produce more accurate warping functions, but they are computationally expensive and therefore often not applicable for the multiple alignment of many samples. Moreover, algorithms that compute an alignment using time warping cannot accommodate for reversals in the retention time of peptides. If in one measurement peptides $A, B,$ and C appear in the order $A-B-C$ and in the second measurement in order $C-B-A$. This scenario is not unlikely if the retention times of A, B and C are similar [Snyder and Dolan, 2007]. Prakash et al. [2006] assume that time order changes do not appear, whereas Prince and Marcotte [2006] address the problem of DTW-based algorithms dealing with time order changes, since these algorithms preserve the temporal order of the peptides. Thereby these methods are only suitable for the determination of the warping function, but not for the mapping of corresponding elements. To assign the correct peptides in different maps, some further processing steps have to be applied, which extract additional useful peptide information e.g., the charge state.

Multiple LC-MS feature map alignment algorithms. In contrast to raw map alignment methods there exist also a great number of approaches for aligning processed LC-MS data sets [Radulovic et al., 2004; Katajamaa et al., 2005; Li et al., 2005; Zhang et al., 2005; Jaitly et al., 2006; Bellew et al., 2006; Smith et al., 2006; Wang et al., 2007]. The feature map

alignment methods can be organized into algorithms, which

- estimate linear or non-linear (typically piecewise linear) dewarping functions and use these transformations to compute a consensus map [Radulovic et al., 2004; Jaitly et al., 2006; Li et al., 2005; Zhang et al., 2005; Bellew et al., 2006]; or
- compute the consensus map directly without the correction of RT and m/z [Wang et al., 2007; Katajamaa et al., 2005].

Furthermore, some of the algorithms compute the final consensus map by

- aligning all maps in a progressive or starwise manner [Radulovic et al., 2004; Katajamaa et al., 2005; Li et al., 2005; Zhang et al., 2005]; or
- assigning corresponding features in all maps simultaneously [Wang et al., 2007; Jaitly et al., 2006; Bellew et al., 2006].

And finally, some of the methods

- use only the 2D positions of the features [Radulovic et al., 2004; Katajamaa et al., 2005; Jaitly et al., 2006; Li et al., 2005],
- whereas other incorporate the ion count, charge, or other feature information [Wang et al., 2007; Zhang et al., 2005; Bellew et al., 2006].

Radulovic et al. [2004] propose an multiple feature map alignment algorithm that is embedded in a software framework for biomarker discovery. The final consensus map, which is called *mother-pamphlet*, is computed in two steps. Using one map as reference map, all other maps are successively aligned to this reference map. First, all best piecewise transformations, which transform the elements of each map onto the coordinate system of the reference map, are determined using a Monte Carlo optimization technique. The similarity score, which is maximized, provides information about the feature overlap between two maps. Finally, the corresponding features are assigned using a "wobble" function that determines the nearest adjacent features in the other maps. Radulovic et al. admit that the proposed alignment is very time consuming and takes the most time during their analysis pipeline. To improve the runtime, they recommend a progressive alignment strategy.

The multiple feature map alignment algorithm of Katajamaa et al. [2005] is also embedded in a software package for the analysis of LC-MS data, called *MZMine*. The simple alignment approach does not estimate any dewarping transformations. The consensus map, which they call *master raw list*, is successively generated. Starting with one map as the initial master raw list, the elements of all other maps are added to the steadily growing master list. Elements lying

within a given RT and m/z window are grouped together to consensus features. This simple alignment strategy is fast, but highly error prone. It assumes only a slight shift in RT and m/z and will fail if the RT dimensions of different maps are additionally scaled.

The multiple feature map alignment algorithm *LCMSWARP* of Jaitly et al. [2006] is developed as a part of an accurate mass and time tag data analysis pipeline [Smith et al., 2002]. The alignment algorithm is based on two steps. In a first step, a reference map is chosen and a piecewise linear warping function for each map with respect to the reference map is estimated. To this end, all maps are broken up into a number of RT segments similar to the COW approach. The number of segments of the reference map and the other maps differs to allow for a scaling of the RT dimension. The best piecewise transformations are determined by maximizing the sum of matching scores of all segment pairs via Dynamic Programming. The matching score assesses the number of assigned features, whereby two features are grouped together if the Mahalanobis distance is smaller than a predefined error bound. The feature matches are used to discover a recalibration function. This function should correct for the error in m/z and allow for a rematching of features. Rematched feature pairs are used to estimate the final transformation in RT using a natural regression spline.
This first step of the algorithms computes an initial alignment, which is further improved in a second step. The determined piecewise transformations are used to dewarp the maps with respect to a reference map and a final consensus map is computed by a two step complete (or single-linkage) clustering approach using again the Mahalanobis metric.

Li et al. [2005] developed a multiple feature map alignment algorithm embedded in a software suite called *SpecArray*. The proposed algorithm computes all pairwise alignments and combines them to a final consensus map. To correct the distortion in RT a retention time calibration curve (RTCC) is iteratively computed for each pairwise alignment. To this end, features with similar m/z values are paired together to construct an original feature pairs set. The retention times of the paired features are used to estimate a retention time calibration curve by minimizing the root mean square distance of the features' RT positions to the monotonic function. Afterward, pairs with a small pairing score are removed and the reduced set of feature pairs is again used to estimate a RTCC. The two steps are repeated until only the pairs with a high pairing score remain and each feature in one map is paired with at most one feature in the other map. The final RTCC curve and the distance of peptides in m/z is used to select likely and unique feature pairs from the original set of feature pairs. The combination of all pairwise alignments yields the final consensus map, or the so-called *super list*. The determination of all pairwise alignments results in a high runtime and makes the algorithm inapplicable for the comparison of a high number of feature maps.

Zhang et al. [2005] propose a heuristic algorithm *XAlign* for the alignment of multiple feature maps. *XAlign* computes in a first step a so-called gross-alignment, where the algorithm corrects a systematic shift in RT. In the second step, a final consensus map, the so-called *micro*

alignment, is determined. The gross-alignment algorithm aligns multiple maps in a starwise manner, whereby the reference map is chosen in the following way. For all predefined RT and m/z windows the most intense features of each map are determined. If a window contains features from all maps, the features are called significant and their intensity weighted average mean RT position is calculated. The map with the minimal difference of all its significant features to the averaged RT positions is chosen as the reference map. Afterward, all other maps are dewarped with respect to the reference by estimating a straight line that minimizes the mean absolute deviation of the RT positions of significant features. In the micro-alignment phase features yielding a high correlation coefficient are successively grouped together and establish the final consensus map.

The multiple feature map alignment algorithm of Bellew et al. [2006] is part of an LC-MS analysis platform called *msInspect*. Before a consensus map, a so-called *peptide array*, is determined the algorithm corrects the non-linear distortions of the RT dimension of all maps in a starwise manner with respect to a certain reference map. Bellew et al. assume that the distortion in RT is explained by a global linear trend plus a remaining non-linear component. The overall non-linear warp for each pairwise alignment is estimated iteratively. In the first step, the linear trend is estimated using the most intense features with similar m/z positions. This initial model of the RT transformation is used to iteratively determine a non-linear transformation using smoothing-spline regression methods from the previous model. After dewarping all maps, a global alignment is performed by applying divisive clustering, whereby the tolerances in RT and m/z of assigned features are user-supplied. The quality of the alignment is defined by the number of clusters that include at most one feature from each map. The algorithm of Bellew et al. optionally offers the automatic choice of the optimal RT and m/z tolerances using the quality of clustering.

The approach of Smith et al. [2006] simultaneously aligns multiple feature maps. The algorithm is also part of a software package, which is called *XCMS*. In a first step, an initial feature matching is determined by grouping all features across the maps with similar m/z positions. Using a kernel density estimator, groups, which contain features with different retention times are split into smaller subgroups. Each group that contains features from fewer than half the maps are eliminated. This gives a coarse matching of features into reasonable groups. To correct for the RT distortion, the median RT and the deviation of the median for every feature in each group are calculated. A local regression fitting method, called loess, uses the deviations in RT within each group to compute a non-linear transformation. This function is used to correct the retention times of all features in the original feature maps and is followed again by matching. To enhance the precision of the final consensus map, the matching/alignment procedure can be repeated in an iterative fashion.

Wang et al. [2007] propose a statistical approach, called *PETAL*, which simultaneously uses feature and raw data information to align LC-MS maps. The algorithm uses not only the

2D position of a features, but also RT range, charge state, and the isotopic distribution of the feature. The features' isotopic distribution scaled to unit total ion count is called *element spectrum vector*. Assume we are given a peptide library that contains all possible features in the multiple maps. The peptide library as well as the features of an individual feature map are represented as a linear combination of the scaled versions of the element spectrum vectors. An assignment of corresponding features is done by means of the similarity of element spectrum vectors. Maximum similarity is determined by fitting a least square regression model penalized by the L_1 norm, with an additional penalization term to prevent the matching of peptides with a great deviation in RT. By varying the scaling factors of the element vectors of the peptide library and searching for the minimum sum of squared distances in the L_1 norm between the element vectors in the aligned map and the scaled element vectors of the peptide library, the abundance of each peptide of the library in the map can be determined. Because the peptide library is usually not known in advance, Wang et al. propose a method to determine a peptide library. Starting with all features of the maps to be aligned, the algorithm selects a proper subset of all features in a backward-stepwise strategy. To extract those features for the peptide library which are contained in preferably many maps, all features are clustered using a sparse regression approach called Elastic net [Zou and Hastie, 2005].
Given the peptide library, all maps can be aligned simultaneously with respect to the peptide library. The proposed method is very time consuming, because the generation of the peptide library takes $O(nk)$ (where n is the number of maps and k the total number of features in all maps). It is more suited for applications in which the peptide library is already given (e.g., peptide "accurate mass tags" [Smith et al., 2002]).

Chapter 14

Own contribution

As mentioned in Section 12.1, the multiple raw map alignment problem (MRMAP) and the multiple feature map alignment problem (MFMAP) can be solved by using efficient point pattern matching approaches. The multiple maps can either be superimposed by the maximization of a specific similarity measure for LC-MS maps, or by an algorithm based on one of the general approaches described in Section 13.1. We will treat both attempts at a solution, and at first propose a fast implementation of our own similarity measure *dsim* along with its area of application. Furthermore we will describe in detail a fast and accurate algorithm [Lange et al., 2007] for the MRMAP and the MFMAP based on the general pose clustering approach. The performance of this algorithm will be evaluated in the following chapter.

14.1 Implementation and applications of *dsim*

The distance function *dsim* as defined in 12.2.2 can be used in several ways. The dissimilarity measure *dsim* could find an interesting and promising application in a progressive alignment approach. It could be used to generate a distance matrix, which includes all pairwise dissimilarities of multiple maps. This matrix defines the generation of a guide tree (heuristic "phylogenetic tree"). The progressive alignment approach starts with the alignment of the most similar LC-MS maps in the hope that the fewest errors are made. Then, progressively, more and more LC-MS are aligned to the already existing alignment. The guide tree can be built by a Neighbor-Joining method [Saitou and Nei, 1987].
Another application could be the superposition of LC-MS maps. Assume we are given two LC-MS maps M and S, which share a fraction of common elements, and the points of M are shifted by a transformation T. The partial APMP would be solved by the determination of the correct transformation parameters, which allow for the superposition of M and S. The distance measure *dsim* could be used in a specific algorithm which determines the correct transforma-

tion parameters by minimizing the distance between two maps. Although we will present a fast implementation of *dsim*, the method proposed in Section 14.2 is an even faster solution for the MRMAP and the MFMAP.

The comparison of hundreds or thousands of maps requires an efficient implementation of the *dsim* measure. Assume we are given two LC-MS maps M and S, the distance measure *dsim* requires the computation of the nearest neighbors of each point of M in S and vice versa. The nearest neighbor of a 2D point in a point set can efficiently be determined in data structures such as Voronoi diagrams or Delaunay triangulations. The Computational Geometry Algorithms Library (CGAL) [Overmars, 1996; Fabri et al., 1996] implemented a 2D point set class `Point_set_2` based on a Delaunay triangulation, which offers efficient nearest neighbor searches and range queries. This data structure can be used to implement an approach for the computation of *dsim*. The construction of the Delaunay triangulation for a point set of size n has a runtime of $O(n \log n)$. For the computation of *dsim* we have to compute a Delaunay triangulation for both LC-MS maps. Afterward, for each element in M we have to determine its nearest neighbor in S and vice versa. The nearest neighbor is determined in constant time using the Delaunay triangulation. Finally, we only have to check which nearest neighbors correspond, if the distance between them is smaller than a given threshold, and sum up the intensity and position dependent similarity value. The total runtime of *dsim* is thus $O(n \log n)$ and the distance between two feature maps of size 195 takes about 40 ms on a typical PC.

A heuristic speed-up by a constant factor can be achieved if we lay a grid onto both maps. The grid size should approximately correspond to the maximum distance in RT and m/z we expect for nearest neighbors. To avoid boundary effects, which can occur using fixed grid cells, we search for the neighbors of each point within a grid cell in M in the corresponding grid cell in S and its surrounding grid cells. The construction of the grid cell takes linear time. However, the search of the neighbor depends on the number of points in each cell. The worst case, where all elements lie within only one grid cell, can be ruled out due to the nature of LC-MS maps and the number of points within each cell can be assumed to be limited by a constant.

14.2 Multiple LC-MS map alignment

In this section we propose a fast and accurate algorithm [Lange et al., 2007] for the *Multiple Raw Map Alignment Problem (MRMAP)* and the *Multiple Feature Map Alignment Problem (MFMAP)*. The solution of the MRMAP is a partial solution of the MFMAP, because the correct superposition of all maps does not only solve the MRMAP but also facilitates the search of common elements in multiple feature maps. The mapping in a star-wise manner of all maps onto a certain reference map leads us to the desired superposition. Because we want to use the superposition algorithm for both raw and feature maps, we design it to be independent of the element type. We use only an element's RT position, m/z position, and ion count, which are

the three characteristics that raw data points and features have in common. For the solution of the pairwise dewarping we developed a powerful pose clustering approach; the algorithm for the search of consensus features is also based on efficient data structures.

In the following sections, we will lay out in detail our approach to multiple LC-MS map alignment and will prove certain properties of our algorithm. In Section 14.2.1, we describe the fast and accurate algorithm for the pairwise map alignment. This algorithm is an improved version of the pose clustering method described in Section 13.1. Section 14.2.2 shows how pairwise map alignment and the resulting transformation can be used to solve for the MRMAP. The method described in Section 14.2.3 expands the pairwise alignment by a search for corresponding elements in two maps. In Section 14.2.4 we show how pairwise map alignment and the search for common elements is combined to an algorithm for the MFMAP. Section 14.2.5 describes the TOPP application, which implemented the algorithms for multiple raw and feature map alignment.

14.2.1 The superposition phase

The Multiple Raw Map Alignment problem searches for a set of transformations that maps all elements of the LC-MS maps onto comparable RT and m/z dimensions such that common elements are shifted closer together. The determination of the correct set of parameters of the underlying warping functions would also allow for the grouping of corresponding elements, which represents the actual solution of the MFMAP. Hence, both problems need the optimal set of transformations.
We developed a star-like progressive multiple alignment approach, which yields the set of transformation functions for both problems. The multiple dewarping approach is based upon pairwise alignments. Given two maps we define the estimation of the transformation that maps one map onto the other as the *superposition phase*. After an initial, coarse transformation is found using pose clustering, results are refined by landmark matching and a final linear regression technique.

Efficient pose clustering for LC-MS data

Given two 2D point sets M and S, the point pattern matching methods described in Section 13.1 determine an affine transformation T such that $T(M)$ best matches S. Depending on the processing stage, LC-MS maps may contain up to 10^8 elements; the straightforward application of these approaches thus is intolerable.
In the following subsection, we will show how we developed an adapted pose clustering algorithm, which accurately solves for the partial point pattern matching problem of LC-MS maps in feasible time. Our pose clustering determines the transformation which maps a maximum

number of points in M close to points in S. Since we use the same metric as in the *dsim* distance function, and also incorporate the similarity of intensities, our approach indirectly maximizes the *dsim* measure of M and S.

Pose clustering is a voting schema and the correct transformation parameters are determined by histogramming. The matching of triples (m_1, m_2, m_3) onto triples (s_1, s_2, s_3) uniquely defines the six unknown parameters of an affine transformation

$$T_\alpha(x) := \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} t_1 \\ t_2 \end{pmatrix}.$$

The parameters of transformation are recorded in a 6D grid and each point matching $(m_i, m_j, m_r), (s_t, s_u, s_v)$ yields a vote for their transformation parameters. In the end the correct transformation is given by the maximum number of votes, because the matching of corresponding tuples will always result in the correct transformation, whereas the transformations of other non-matching tuples are more or less randomly distributed.

Olson [1998] shows that besides a speed-up in runtime, the accuracy of pose clustering is improved by limitations of the pose space. He constricts the pose space by computing only poses for triples in the shape point set and image point set that are possible real matches. We develop a similar approach and use the characteristics of LC-MS measurements to limit the pose space and, on the other hand, improve the runtime.

We develop a similar approach and introduce four improvements exploiting the characteristics of LC-MS measurements. Given two point sets of size k and l, the general pose clustering approach solving for an affine transformation has a runtime of $O(k^3 l^3)$. The four improvements described in the next four subsections limit the pose space and reduce the number of false positives. The first three improvements achieve a remarkable speed-up of pose clustering and achieve a total runtime of $O(kC^2)$ with a constant $C \ll l$. The fourth improvement reduces the false-positive rate.
Figure 14.1 summarizes the stepwise improvement of runtime.

$$O(k^3 l^3) \xrightarrow{\text{Improvement 1}} O(k^2 l^2) \xrightarrow{\text{Improvement 2}} O(k^2 C^2) \xrightarrow{\text{Improvement 3}} O(kC^2)$$

Figure 14.1: *The runtime of the pose clustering approach after the incorporation of each improvement.*

Pose clustering improvement 1—Nature of the warp in RT and m/z dimension. Due to the fact that the RT and the m/z are based on the measurements of two different analysis techniques, the uncertainties in measurement are independent. The mass spectrometer may be well calibrated and thereby the error in m/z may be small, but the mobile phase of the LC column

may change during measurement; this could result in drifts in the retention time of the measured compounds. We can therefore address the warp in both dimensions independently. In case of a well-calibrated mass spectrometer, the shift in m/z should be minimal and can be described by an affine transformation. Jaitly et al. [2006] notice that it is expected from the central limit theorem that even after correcting for global trends of dead time and flow rate changes, effects of less understood factors can result in the observed elution times being normally distributed around an ideal elution time. Also, we observed that an affine transformation is frequently sufficient for the RT dimension. Figure 14.2 and Figure 14.3 show the results of an experiment, which supports this observation of an affine transformation in RT.

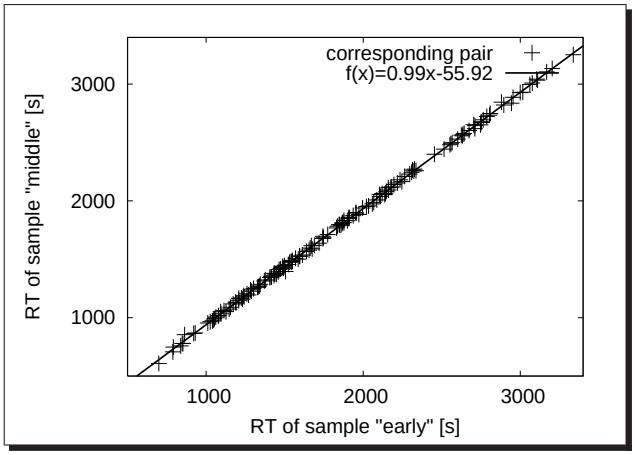

Figure 14.2: *Retention times of corresponding peaks (*159 *verified common identifications) in two LC-MS raw maps of the Mycobacterium smegmatis experiment (see Section 15.3). Sample "early" is a protein profiling of Mycobacterial smegmatis in early exponential phase, whereas "middle" states a protein profiling in middle exponential phase.*

We selected a set of high-confidence peptides in two LC-MS samples of Mycobacterium smegmatis. Corresponding peptides in both samples, verified by common identifications, were matched and manually validated. An affine correction was applied to the RT coordinates yielding a Pearson correlation coefficient of 0.999. Figure 14.2 shows the corresponding pairs distributed over the whole RT axis of the experiment and the fitted affine warp. For each pair, we plot difference versus mean RT. As can be seen in Figure 14.3, the error in retention time remaining after correction is scattered around zero.

Although we could compute transformations using higher-order functions it is doubtful whether they are necessary or even practical since there is the potential of overfitting. Furthermore, it should be noted that each additional parameter will expand the pose space by one dimension and therefore increase both runtime and memory space. However, our definition allows also for non-affine functions.

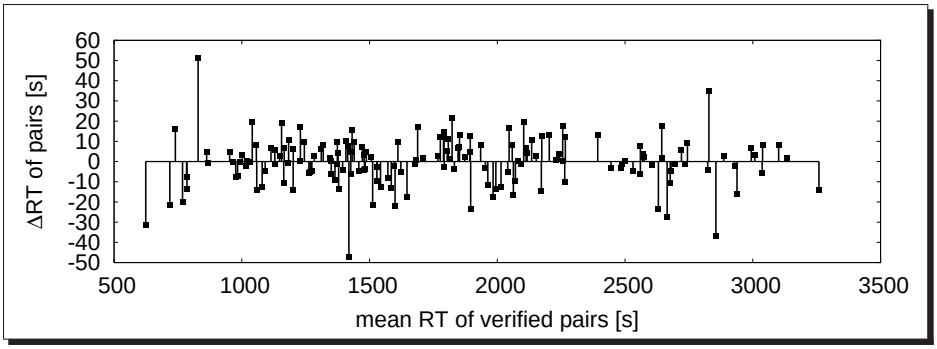

Figure 14.3: *The plot shows the remaining differences in retention time after a suitable affine dewarping function has been applied to the time standard of the "early" and "middle" sample. For each pair of retention times (s_i, m_i), we plot $s_i - m_i$ (vertically) against $(s_i + m_i)/2$ (horizontally). The figure shows that the remaining error after affine dewarping is almost independent of the retention time. The affine transformation used for dewarping was calculated by a linear regression of all retention time pairs.*

We define the warps in RT and m/z by one-dimensional affine transformations $T_{\mathrm{RT}}(e) := a_{\mathrm{RT}} e_{\mathrm{RT}} + b_{\mathrm{RT}}$ and $p_{\mathrm{m/z}}(e) := a_{\mathrm{m/z}} e_{\mathrm{m/z}} + b_{\mathrm{m/z}}$. The two-dimensional warping function p, which transforms the positions e_{RT} and $e_{\mathrm{m/z}}$ of an element e into $\tilde{e}_{\mathrm{RT}}$ and $\tilde{e}_{\mathrm{m/z}}$ is given by

$$p(e_{\mathrm{RT}}, e_{\mathrm{m/z}}) := (\tilde{e}_{\mathrm{RT}}, \tilde{e}_{\mathrm{m/z}}) = \begin{pmatrix} a_{\mathrm{RT}} & 0 \\ 0 & a_{\mathrm{m/z}} \end{pmatrix} \begin{pmatrix} e_{\mathrm{RT}} \\ e_{\mathrm{m/z}} \end{pmatrix} + \begin{pmatrix} b_{\mathrm{RT}} \\ b_{\mathrm{m/z}} \end{pmatrix}.$$

This special case of two independent affine transformations limits the pose space to only four instead of six dimensions. To solve for the four unknown parameters in p, two points m_1, m_2 of one map along with two points (s_1, s_2) of the other map are needed. Using the system of four linear equations

$$\begin{aligned} s_{1,\mathrm{RT}} &= a_{\mathrm{RT}} m_{1,\mathrm{RT}} + b_{\mathrm{RT}} \\ s_{2,\mathrm{RT}} &= a_{\mathrm{RT}} m_{2,\mathrm{RT}} + b_{\mathrm{RT}} \\ s_{1,\mathrm{m/z}} &= a_{\mathrm{m/z}} m_{1,\mathrm{m/z}} + b_{\mathrm{m/z}} \\ s_{2,\mathrm{m/z}} &= a_{\mathrm{m/z}} m_{2,\mathrm{m/z}} + b_{\mathrm{m/z}} \end{aligned}$$

the parameters that map m_1 onto s_1 and m_2 onto s_2 can be uniquely determined. With k elements in one LC-MS map M and l elements in another LC-MS map S we yield $2\binom{k}{2}\binom{l}{2}$ distinct matching tuples $((m_i, s_r), (m_j, s_t))$ that result in the same number of hypothesized poses of M in S. If the model point set M is completely contained in S and thereby all points k points of M have a corresponding point in S then the correct pose is supported by $\binom{k}{2}$ matching tuples. Even in case of partial matching when only fk (f is the fraction of the model points that appear in S) points of M have a corresponding point in S the correct pose is supported by $\binom{fk}{2}$ matching tuples and they form a cluster in the pose space. The introduction of a special affine transformation reduces the complexity of pose clustering to $O(k^2 l^2)$.

Pose clustering improvement 2—Hypothesized correspondence in m/z. Given two LC-MS element maps M and S. Even in case of insufficient calibrated mass spectrometers, the deviation of corresponding elements (m_i, s_j) in m/z position should be smaller or equal than the mass spectrometer's precision $\sigma_{\text{m/z}}$

$$|m_{i\,\text{m/z}} - s_{j\,\text{m/z}}| \leq \sigma_{\text{m/z}}.$$

To allow for errors which can occur during processing of the maps we define an error tolerance $\varepsilon_{\text{m/z}} > \sigma_{\text{m/z}}$ in m/z and take advantage of the precision of mass spectrometric measurements and limit the pair of matching tuples $((m_i, s_k), (m_j, s_l))$

$$|m_{i\,\text{m/z}} - s_{k\,\text{m/z}}| \leq \varepsilon_{\text{m/z}} \text{ and } |m_{j\,\text{m/z}} - s_{l\,\text{m/z}}| \leq \varepsilon_{\text{m/z}}$$

to those that meet the above condition with and are likely to originate the correct pose. Figure 14.4 illustrates the hypothesized correspondence of an element. The arrows indicate the potential corresponding elements in the other map, which lie in between a certain $\varepsilon_{\text{m/z}}$-environment.

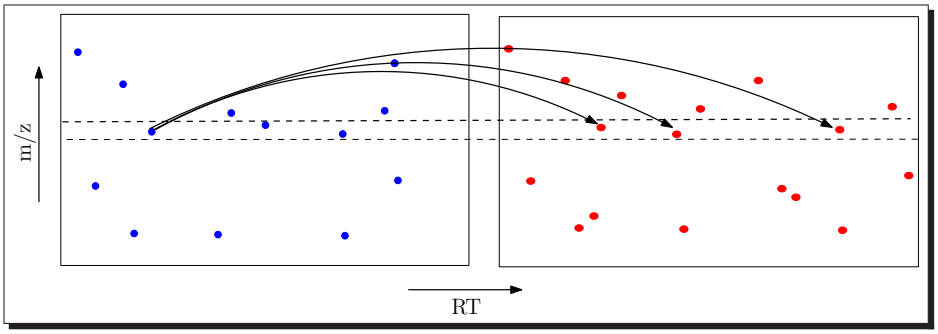

Figure 14.4: *The blue points represent a point set M and the red points form point set S. Potential partners of a point $m \in M$ in S have to lie in between a certain error bound, shown by the dotted lines. The arrows indicate the hypothesized partners of m.*

In the worst case, the complexity remains $O(k^2 l^2)$. But this case, where all points in M and S lie within $\varepsilon_{\text{m/z}}$, is unrealistic and under real circumstances, the number of hypothesized partners of each point in M is typically constant in the number of points and bounded by a constant C. The runtime becomes then $O(k^2 C^2)$ with $C \ll l$. The approach of hypothesized correspondence in m/z has the same effect as the grouping technique of Olson [1998] and does not only improve the complexity but also the accuracy of pose clustering by elimination of a large number of false positive poses.

Pose clustering improvement 3—Decomposition of the problem. To further improve the complexity of our algorithm we propose a similar decomposition technique as described in

Olson [1997]. The idea behind the following theorem is that if we take a model point m_1 of M, which has a corresponding partner in S, and compute all matching tuples $((m_1,s_i),(m_j,s_n))$ that include m_1 then we will already yield a cluster of size $k-1$ at the correct pose in pose space. This avoids computing all the possible matching tuples and reduces complexity from $O(k^2)$ to $O(k)$. The matching of m_1 onto s_1 and m_2 onto s_2 is called a group match $\gamma := \{(m_1,s_1),(m_2,s_2)\}$. A subset $\theta(\gamma)$ of the pose space Θ can achieve the matching of m_1 onto s_1 and m_2 onto s_2 within some error bound

$$\theta(\gamma) \equiv \{p \in \Theta : \|p(m_i) - s_i\| \leq \varepsilon, \text{ for } 1 \leq i \leq 2\}.$$

Theorem 14.2.1: The following statements are equivalent for each pose $w \in \Theta$:

1. There exist $g = \binom{k}{2}$ distinct group matches that pose p brings into alignment up to the error bounds. Formally, $\exists \gamma_1, \ldots, \gamma_g$ s.t. $w \in \theta(\gamma_i)$ for $1 \leq i \leq g$.

2. There exist k distinct point matches $\pi_1, \ldots, \pi_k$ with $\pi_i = (m_i, s_j)$ that pose p brings into alignment up to the error bounds: $\exists \pi_1, \ldots, \pi_k$ s.t. $w \in \theta(\{\pi_i\})$ for $1 \leq i \leq k$.

3. There exist $k-1$ distinct group matches sharing one point match that pose p brings into alignment up to the error bounds: $\exists \pi_1, \ldots, \pi_k$ s.t. $\theta(\{\pi_1, \pi_i\})$ for $2 \leq i \leq k$.

Proof. We will prove in a circular fashion that $1 \Leftarrow 2$, $2 \Leftarrow 3$, and $3 \Leftarrow 1$. Therefore, the three statements must be equivalent.
$1 \Leftarrow 2$: Each of the group matches is composed of a set of two point matches. The fewest point matches from which we can choose $\binom{k}{2}$ group matches is k. The definition of $\theta(\gamma)$ guarantees that each of the individual point matches of any group match that is brought into alignment are also brought into alignment. Thus, each of these k point matches must be brought into alignment up to the error bounds.
$2 \Leftarrow 3$ Choose a point match that is brought into alignment. Form all of the $k-1$ group matches composed of this point match and each of the additional point matches. Since each of the point matches is brought into alignment, each of the group matches composed of them also must be from the definition of $\theta(\gamma)$.
$3 \Leftarrow 1$ There are k distinct point matches that compose the $k-1$ group matches, each of which must be brought into alignment. Any of the $\binom{k}{2}$ distinct group matches that can be formed from them must therefore also be brought into alignment. □

If we knew in advance an element of the model map M that has a partner in the element map S we only had to bin all $(k-1)\binom{C}{2}$ possible poses and could achieve a runtime of $O(kC^2)$. Unfortunately, we do not know anything about correspondence in the two maps and have to find a common element of M and S by chance. If we randomly choose a point of M that has a corresponding element in S we will find the correct pose in pose space, but how many trials are

required until we choose a correct element? We will derive an upper bound of not choosing a correct point in M in t trials if fl model points are present in S. The probability for a randomly chosen element to be correct is $\frac{fl}{k}$ and to be wrong $(1-\frac{fl}{k})$. Thereby, the probability to choose t wrong elements in t trials is $p=(1-\frac{fl}{k})^t$. If we require the probability of a false negative to be less than δ we have:

$$(1-\frac{fl}{k})^t \quad \leq \quad \delta.$$

Solving for t leads to:

$$\begin{aligned} t\ln(1-\frac{fl}{k}) &\geq \ln(\delta) \\ t &\geq \frac{\ln(\delta)}{\ln(1-\frac{fl}{k})} \end{aligned}$$

Using the approximation of $\ln(1+x)\approx x$ for $x\to 0$ we get a first order approximation of t as a lower bound of t

$$t \quad \geq \quad \ln(\frac{1}{\delta})(\frac{k}{fl}) = O(\frac{k}{l}).$$

From this follows that we have to evaluate at least $O(\frac{k}{l})$ model points to expect at least one correct model point in t trials. Each model point can be matched onto C hypothesized element partners in the other map. For the choice of the second model and scene element, $O(kC)$ possibilities remain. For $k\approx l$ we achieve a complexity of $O(kC^2)$ and only if $k\gg l$ the complexity remains $O(k^2C^2)$.

Pose clustering improvement 4—Incorporation of intensity information To reduce the number of false positive clusters in pose space we can further incorporate the elements' intensity values. By a simple normalization using the total ion count of a map the elements' intensities in different maps become comparable. The ion count of corresponding elements in maps resulting from repeated measurements should be almost identical and even in case of maps containing differentially expressed peptides, the majority of peptides and their intensities should be usually similar. We exploit this property and multiply each vote by a weight indicating a level of confidence in the mapping before we histogram it. We give matching points with similar intensity a higher vote than the matching of points with varying ion counts.

We could easily incorporate other constraint such as equal charge state of matched peptides, to prevent histogramming of unrealistic candidate transformations. We disregarded these possible extensions because we want to stay independent of the element type and our algorithm should work for raw, and feature maps.

Landmark matching

Although the pose clustering approach yields a suitable estimation of the warp in RT and m/z, the initial pose estimation can still be improved upon. We use the initial estimate of the underlying warp in RT and m/z to detect reliable element pairs in two maps, which are likely to represent corresponding elements and use these pairs as "landmarks". Landmarks are element pairs that are likely to represent common elements.
Even in case of an RT warp that would be better approximated by a higher order polynomial than by an affine transformation, the RT warp usually has a prominent linear trend and is globally smooth. If we determined an adequate estimate of the correct transformation the application of the initial transformation should map corresponding elements closer together and some of them should even become nearest neighbors.

Adaptation of the Euclidean metric. Because in an LC-MS map the element's RT position is affected by a much larger measurement error than the m/z position, we cannot use the Euclidean metric to determine a nearest neighbor. A typical uncertainty in the retention time measurement lies between 10–30 s (the remaining errors in RT of the corresponding elements in Figure 14.3 yield a mean of 10^{-13} and a standard deviation of 12.33 s). However, the mass accuracy is in the ppm range. In our case, differences in m/z are much less tolerable, and should be weighted more heavily, than differences in RT. We adapted the Euclidean metric to this purpose by introducing scaling factors w_1 and w_2 for the RT and m/z positions.

Definition 14.2.1: The adapted Euclidean distance $\widetilde{\mathrm{euc}}(m,s)$ between two elements $m := \begin{pmatrix} m_{\mathrm{RT}} \\ m_{\mathrm{m/z}} \end{pmatrix}$ and $s := \begin{pmatrix} s_{\mathrm{RT}} \\ s_{\mathrm{m/z}} \end{pmatrix}$ is defined by

$$\widetilde{euc}(m,s) := \sqrt{w_1^2(m_{\mathrm{RT}} - s_{\mathrm{RT}})^2 + w_2^2(m_{\mathrm{m/z}} - s_{\mathrm{m/z}})^2}.$$

with $w_1, w_2 \in \mathbb{R}$.

Search for landmarks. The initial transformation is precise enough that at least a subset of corresponding elements becomes nearest neighbors in respect to the adapted Euclidean metric as defined in 14.2.1. To ensure reliable matching pairs, the hypothesized element pairs should fulfill two conditions:

1* Two elements can be matched only if, for each of them, the other one is the nearest neighbor within a given error bound in the other map, and

2* the distance to the second-nearest neighbor is significantly larger than the distance to the nearest one.

The second condition necessitates not only the determination of an element's nearest neighbor but also of its second nearest neighbor. Given two LC-MS maps $M := \{m_1, \ldots, m_k\}$ and $S := \{s_1, \ldots, s_l\}$ whereby the elements of M are already dewarped. By searching only the 2-nearest neighbors of each m_i in S and not vice versa we can accelerate runtime greatly, but we also have to relax the first and second condition mentioned above to:

1 Two elements m_i, s_j can be matched only if s_j is nearest neighbor of m_i and all other points m_r for which s_j is nearest neighbor have a larger adapted Euclidean distance to s_j, and

2 the adapted Euclidean distance to the second-nearest neighbor of m_i is significantly larger than the distance to s_j. Furthermore, m_r also has s_j as nearest neighbor and has the second-smallest distance to s_j of all points that have s_j as nearest neighbor. The distance of m_r and s_j is significantly larger than the distance between s_j and m_i.

Figure 14.5 illustrates condition 1 and 2. The nearest neighbor of a point in the other map is indicated by an arrow, and the second nearest neighbors by an arrow with a dotted line. If two points are linked by a left right arrow, and condition 2^* holds, the two points form a pair, which is highlighted by a solid rectangle.

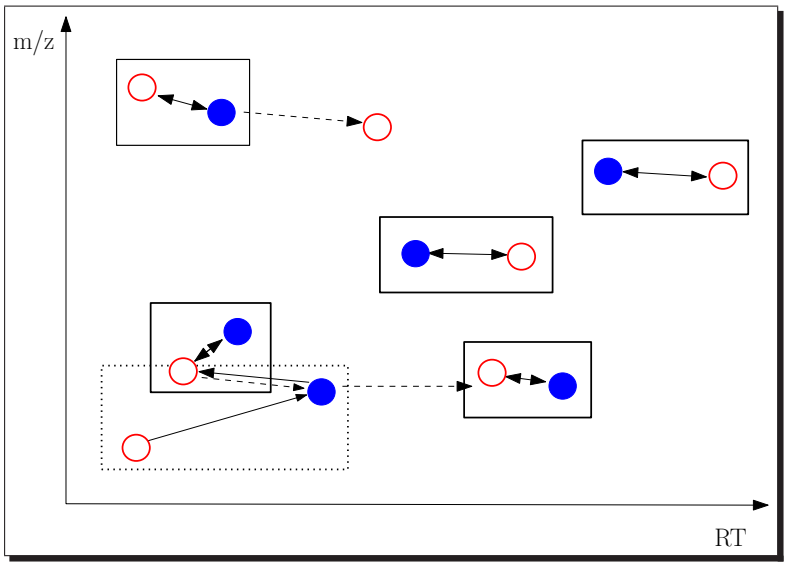

Figure 14.5: *The empty circles represent a point set M and the filled circles form point set S. If the distance to the nearest neighbor of a point is significantly small enough the point and its nearest neighbor are linked by a solid arrow. The second nearest neighbor, with a significantly small enough distance is indicated by a dotted arrow. Pairs that fulfill condition 1^*, 2^*, 1, and 2 are framed by a solid rectangle, whereby pairs meeting only 1 and 2 are highlighted by a dotted rectangle.*

We will now show that a set of matching pairs that obeys condition 1^* and 2^* is a subset of all matching pairs that fulfill condition 1, and 2.

Theorem 14.2.2: An algorithm that searches for matching pairs that meet condition 1 and 2 also yields all matching pairs that fulfill condition 1^*, and 2^*.

Proof. Assume condition 1^* and 2* hold for m_i and s_j and condition 1 and 2 are violated. If condition 1^* holds m_i is nearest neighbor of s_j and vice versa and all other points m_r have a greater adapted euclidean distance to s_j than m_i, which obeys condition 1. The condition 2^* implies that the distance between m_i and the second nearest neighbor of m_i is significantly larger than the distance between m_i and s_j, which meets the first part of condition 2. Furthermore, the distance between s_j and m_i is significantly smaller than the distance between s_j and any other point m_r, which has s_j as a nearest neighbor, which fulfills the second part of condition 2. Thereby, given condition 1^* and 2^* condition 1 and 2 hold, which contradicts the initial assumption. □

The relaxed conditions expand the number of hypothesized element pairs, which obey condition 1* and 2* by pairs m_i, s_j for which at least s_j is the nearest neighbor of m_i. Furthermore, the distance between m_i and s_j is sufficiently small such that all other points m_r, which have a smaller distance to s_j are already paired with an s_t, which lies closer to m_r than s_j. Such pairs m_i, s_j can also be treated as hypothesized element pairs and are appended to the list of landmarks. The dotted rectangle in Figure 14.5 illustrates such a pair.

The set of matching pairs meeting condition 1 and 2 can be determined by first searching for the 2-nearest neighbors for all m_i in S using a Delaunay triangulation of S for the adapted Euclidean metric as defined in 14.2.1. A Delaunay triangulation $D(S)$ of S is created in $O(l \log l)$ time and needs $O(l)$ space [Mehlhorn and Näher, 1999; Boissonnat et al., 2000]. Besides the Delaunay triangulation of S we need a lookup table L, which stores for each s_i a list of points in M choosing s_i as nearest neighbor. Using $D(S)$ and L the determination of landmarks is performed as follows: For each m_i we use $D(S)$ to find its nearest s_r and second nearest neighbor s_t in S. If the distance between s_r and s_t is large enough we append m_i to the list of s_r in L. In the end, L has to be processed to determine the matching pairs:

1. If $|l_i| = 0 \Rightarrow s_i$ has no matching partner in M.
2. If $|l_i| = 1$ and $l_i = [m_j] \Rightarrow (s_i, m_j)$ is a hypothesized element pair.
3. If $|l_i| > 1$ and $l_i = [m_1, \ldots, m_n]$ with $m_j < m_{j+1}$ for $1 \leq j < n$ and the adapted Euclidean distance of s_i and m_1 is significantly larger than the distance between s_i and $m_2 \Rightarrow (s_i, m_1)$ is a hypothesized element pair.

The total runtime of landmark search takes $O(l \log l)$ for the creation of $D(S)$ and the nearest neighbor search of all m_i. The runtime of a nearest neighbor search of a point m_i in $D(S)$ is constant. It consists of the insertion of m_i in $D(S)$, search for the two nearest nodes of m_i in $D(S)$, and deletion of m_i in $D(S)$).

Improvement of the initial warp by linear regression. In a second step, we can refine the estimated warp even further. The landmarks obtained in the previous step are used to obtain the final transformation by linear regression. We again assume an affine transformation, which maps M onto S but at this point any other type of transformation can be estimated using the matching pairs.
The linear regression method calculates the translation and scaling factors, which minimize the sum of the squared deviations of the pairs in RT and m/z

$$\begin{aligned}\{a_{\mathrm{RT}}, b_{\mathrm{RT}}\} &= \arg\min_{a,b\in\mathbb{R}} \sum_i (s_{\mathrm{RT}} - (a_{\mathrm{RT}} * \mathrm{RT}(m_i) + b_{\mathrm{RT}}))^2 \\ \{a_{\mathrm{m/z}}, b_{\mathrm{m/z}}\} &= \arg\min_{a,b\in\mathbb{R}} \sum_i (s_{\mathrm{m/z}} - (a_{\mathrm{m/z}} * \mathrm{RT}(m_i) + b_{\mathrm{m/z}}))^2\end{aligned}$$

with $2 \leq i \leq \min(k,l)$.

While the final transformation given by $T(e) := Ae + t$ with the scaling matrix

$$A := \begin{pmatrix} a_{\mathrm{RT}} & 0 \\ 0 & a_{\mathrm{m/z}} \end{pmatrix} \in \mathbb{R}^{2\times 2} \text{ and translation vector } t := \begin{pmatrix} b_{\mathrm{RT}} \\ b_{\mathrm{m/z}} \end{pmatrix} \in \mathbb{R}^2$$

will typically not differ much from the initial transformation, it is guaranteed to be at least locally optimal.
Moreover, it renders our algorithm robust to small changes in the parameter settings applied for pose clustering, such as histogramming bin size, and m/z tolerance.

Piecewise defined transformation. Our multiple alignment approach allows not only for the determination of a globally defined affine warp, but also for an affine warp that is piecewise defined.
Considerable problems with the chromatogram during an LC-MS measurement can result in significant distortion of the RT dimension and sometimes the variability of corresponding elements' retention times then is better approximated by a piecewise affine function. Particularly, for the multiple alignment of raw maps a precise estimation of the shift is more important than in multiple feature map alignment where corresponding elements can typically be found even with a less restrictive estimate of the warp.
To compute an initial piecewise transformation with pose clustering, given two maps M and S, we first partition the model map M into segments $M_1, \ldots, M_m$. Afterward, we follow the approach as described in Section 14.2.1 for each M_i and S. Thereby, the partition should ensure that each segment M_i contains a number of common elements of M and S; otherwise, we will find only false positive poses during histogramming. The transformations $T_i(e) := A_i e + b_i$ (with $A_i \in \mathbb{R}^{2\times 2}$, $b_i \in \mathbb{R}^2$, and $1 \leq i \leq m$), which define the initial piecewise transformation T

$$T(e) = \begin{cases} T_1(e) & e \in M_1 \\ \vdots & \\ T_m(e) & e \in M_m \end{cases}$$

are improved by linear regression as described in Section 14.2.1.
It has to be noted that this procedure results in a piecewise affine transformation that is not guaranteed to be continuous at the boundaries of M_i. Without the knowledge of the warp the co-domains of T_i cannot be limited and each M_i has to be mapped onto the almost whole map S. But again we can use the pose clustering approach in Section 14.2.1 to achieve a globally defined affine transformation in a first step. The substitution of the linear regression in Section 14.2.1 by a linear spline regression as described by Ertel and Fowlkes [1976] will yield a continuous defined piecewise affine transformation.

Final algorithm

In Section 14.2.1 we developed an efficient algorithm for pairwise dewarping, the so-called superposition phase. Figure 14.6 shows the pseudocode of our algorithm, which implements the effective pose clustering approach described in Section 14.2.1 followed by the procedure in Section 14.2.1 to find the optimal affine warp $T(x) := Ax + t$ (with $A \in \mathbb{R}^{2\times 2}, \in \mathbb{R}^2$) in RT and m/z, which shifts common elements in two maps closer together.

A first estimate of the correct warp parameters A and t can be recovered from the data using the pose clustering approach as described in Section 14.2.1. Following the paradigm of pose clustering, we find the initial warp by a voting scheme. Consider the set of solutions of the local superposition problem for all pairs of pairs of data points $\big((m_1,s_1),(m_2,s_2)\big)$. In the space of all affine transformations (which is spanned by the parameters A and t) the correct transformation shows up as an accumulation point (or cluster), whereas the local solutions for non-matching pairs of pairs are more or less randomly distributed over the (A,t) plane. An example is shown in Figure 14.7.

We use the centroid of the accumulation point as a guess for the optimal transformation. These initial parameters are optimized afterward.

The algorithm records the candidate transformations in a hash table. The hash table itself is implemented as a sparse matrix, and the vote of a candidate transformation is distributed among its four neighboring discretized positions in the hash table in such a way that by taking their weighted average we will retrieve the original parameters.

In its simplest form, the voting scheme could iterate over all pairs of pairs of features and then search for the accumulation point using the hash table. However this leads to an $\Omega(k^2 l^2)$ algorithm, which is potentially very slow or even infeasible for $k,l \geq 1000$, as is often the case in real applications.

Fortunately, the set of candidate transformations is highly restricted for LC-MS maps and the incorporation of three of the four improvements described in Section 14.2.1 leads to a total runtime of $O(k^2C^2)$, where C is the number of potential matching partners of each s_i in S. The

SUPERPOSITION PHASE

Input: Reference map $S := \{s_1, \ldots, s_k\}$, and map $M := \{m_1, \ldots, m_l\}$
Output: Affine transformation T_{final} such that $T_{\text{final}}(M) \approx S$.

```
// preprocessing
normalizeIntensities(S)
normalizeIntensities(M)
for all elements m in M do
   partner_list_m = searchForPartners(S, m)
end for
// find initial transformation by pose clustering
for all elements m_1 in M do
   for all elements m_2 ≠ m_1 in M do
      for all partners s_1 of m_1 in partner_list_m1 do
         for all partners s_2 ≠ s_1 of m_2 in partner_list_m2 do
            T_α=computeTransformation((m_1, s_1), (m_2, s_2))
            if isAdmissible(T_α) then
               hashTransformationParameters(α)
            end if
         end for
      end for
   end for
end for
T_initial = estimateInitialTransformation()
// find corresponding element pairs
M_dewarped = dewarp(M, T_initial)
computeDelaunayTriangulation(S)
pair_list = findElementPairs(S, M_dewarped)
// compute final transformation
T_final = linearRegression(pair_list)
```

Figure 14.6: *Pseudocode of the superposition phase.*

implementation of improvement 3 is work in progress, but in the current version of the algorithm we also reduce the number of pairs considered in each M_i during the superposition phase. We observe that it is sufficient to consider only pairs of points in M_i that lie close together in m/z. This is a reasonable assumption since local distortions are frequently dominating and leads also to the same runtime as improvement 3 with $O(kC^2)$.

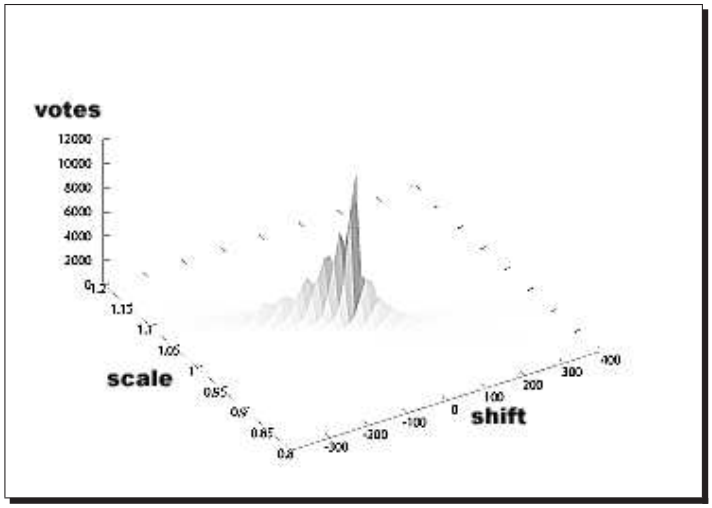

Figure 14.7: *Histogram of the transformation hash table used for aligning two M. smegmatis samples in middle exponential phase. The accumulation point stands out clearly. The minor "ripples" are artifacts due to the discretization of positions during the re-sampling.*

Once we have computed and hashed all potential poses the accumulation point can be found in the hash map, and we estimate the parameters of the transformations using a weighted average over a small neighborhood of it, to compensate for discretization errors and random fluctuations present in the data.

Finally, we apply the initial transformation to the model and search for landmarks, which meet condition 1 and 2 on page 130. To determine the matching pairs we use the `Point_set_2` class of the Computational Geometry Algorithms Library (CGAL) [Overmars, 1996; Fabri et al., 1996] and its fast near k nearest neighbors search based on a Delaunay triangulation.

Given a list of landmarks we obtain the final transformation by linear regression.

14.2.2 Application to LC-MS raw maps

In Section 14.2.1 we developed an efficient algorithm for pairwise dewarping, the so-called superposition phase. Given two LC-MS maps M and S, a first estimate of the correct transformation that maps common elements in M and S onto each other within some error bound is estimated using an adapted pose clustering approach. Afterward, the initial transformation is improved by linear regression and results in a final affine estimate of the warp in RT and m/z.
We use the pairwise dewarping in an algorithm for solving the MRMAP. We are given a set of element maps $\{M_0, \ldots, M_n\}$. First, we select the map with the highest number of elements. It is used to initialize the reference map M_{ref}. The other maps are successively aligned to the ref-

erence map. Thus we perform a star-like progressive multiple alignment based upon pairwise alignments. The superposition phase results in the optimal affine warp, which shifts common elements of each map M_i and the reference map M_{ref} close together. Applying these warps to the other maps we transform all elements onto the coordinate plane of the reference map and solve the MRMAP. Figure 14.8 illustrates the workflow of our multiple raw map alignment algorithm.

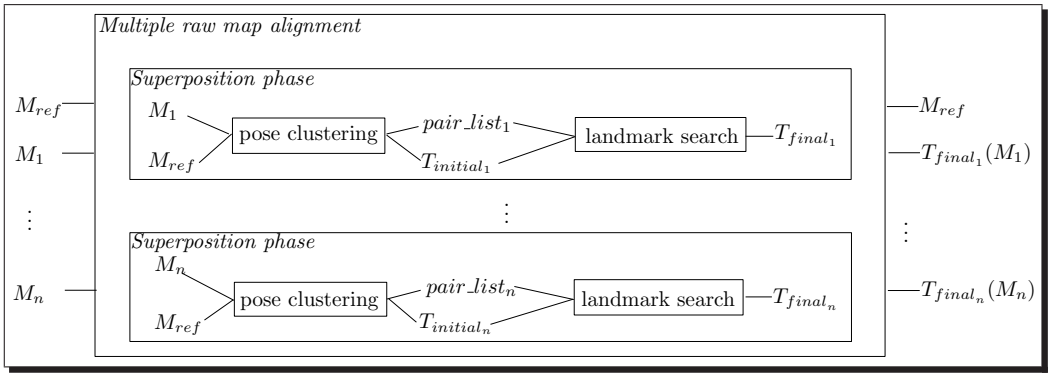

Figure 14.8: *Workflow of the multiple raw map alignment approach. Given $n+1$ maps, our algorithm dewarps n maps with respect to a chosen reference map.*

The pseudocode of the algorithm for multiple alignment of raw LC-MS maps is shown in Figure 14.9.

MULTIPLE ALIGNMENT OF RAW MAPS

Input: List of element maps $map\_list = M_1, \ldots, M_n$
Output: List of dewarped element maps $map\_list = \widetilde{M}_1, \ldots, \widetilde{M}_n$

```
// multiple alignment
// choose map with highest number of elements as reference map
ref = indexOfReferenceMap(M_1, ..., M_n)
M̂_ref = reduceElements(M_ref)
M̃_ref = M_ref
for all maps M_i in map_list (with i ≠ ref) do
    M̂_i := reduceElements(M_i)
    T_i := superpositionPhase(M̂_ref, M̂_i)
    M̃_i:= dewarp(M_i, T_i)
end for
```

Figure 14.9: *Pseudocode of the multiple raw map alignment.*

14.2.3 The consensus phase

Using the superposition phase we can not only solve the MRMAP, but also facilitate the search of corresponding elements in the MFMAP. Assume we have two maps M and C, which are mapped onto the same coordinate system, such that corresponding elements lie within a given error bound. Let M be an element map and C a consensus map. Assigning the elements of M to the consensus features in C we can use again the algorithm proposed on page 130. We consider M and C as two-dimensional point sets given by the elements' RT and m/z positions.
The elements of C are represented by their consensus RT and m/z positions. A consensus RT position is the weighted mean of the RT positions of all combined features, whereby the features' ion counts serve as weights. The consensus m/z position is determined in the same way.
Common elements (c_1, m_1) of M and C should meet conditions 1 and 2 defined on page 131. Accordingly, given $\varepsilon, d > 0$, a pair of potential corresponding elements (c_1, m_1) lie within an ε-neighborhood and all other $m_2 \in M$ have a distance further than d to m_1. Furthermore, the distance between c_1 and all other points c_2, which chose m_1 as nearest neighbor, is greater than d. These conditions allow a unique assignment of common elements, and elements m, which do not belong to a certain consensus feature c, are pushed into the consensus map as singleton elements. The pseudocode of the algorithm for the consensus-phase is shown in Figure 14.10.

14.2.4 Application to LC-MS feature maps

We can use the superposition phase in Section 14.2.1 along with the consensus phase in Section 14.2.3 to solve the MFMAP. Given multiple feature maps $M_0, \ldots, M_n$, we compute n affine transformations $T_1, \ldots, T_n$ using the algorithm shown in Figure 14.6, which allows the superposition of all maps $M_{ref}, T_1(M_1), \ldots, T_n(M_n)$ with respect to a chosen reference map M_{ref}. Using the dewarped maps $\widetilde{M}_i := T_i(M_i)$ and the initial consensus map C_0, containing all elements of M_{ref} as singleton consensus features, we can build a consensus map C_n of the $n+1$ maps by applying n times the algorithm of the consensus phase (see Figure 14.10).

In each consensus phase $s \in \{1, \ldots, n\}$ we assign the elements of an M_s to iteratively growing consensus map C_{s-1}. The final consensus map contains the elements of all $n+1$ maps either as part of a consensus feature or as a singleton consensus feature. The result C_n of the progressive alignment approach depends on the order in which the M_i are combined to a consensus map.

CONSENSUS PHASE

Input: Map $M := \{m_1, \ldots, m_k\}$, and consensus map $C := \{c_1, \ldots, c_l\}$
Output: Consensus map $\widehat{C}$

```
Ĉ := C
D(M):=delaunayTriangulation(M)
for all elements c in Ĉ do
   (m1, m2):=findTwoNearestNeighbors(D(M),c)
   if (ẽuc(m1, m2) > d) ∧ (ẽuc(m1, c) ≤ ε) then
      L(m1) := append(L(m1),c)
   end if
end for
// find hypothesized pairs meeting condition 1 and 2
for all lists l := L(m) in L with m ∈ M do
   // no c has m as nearest neighbor
   if |l| = 0 then
      cm := buildConsensusElement(m)
      Ĉ := append(Ĉ, cm)
   else
      // m has one nearest neighbor
      if l = {c} then
         ĉ1 := combine(c1, m)
         Ĉ := replace(Ĉ, c1, ĉ1)
      else
         // m is the nearest neighbor of c1 and c2
         // with ẽuc(c1, m) ≤ ẽuc(c2, m)
         if ẽuc(c1, c2) > d then
            ĉ1 := combine(c1, m)
            Ĉ := replace(Ĉ, c1, ĉ1)
         else
            // m can not be uniquely assigned to c1
            cm := buildConsensusElement(m)
            Ĉ := append(Ĉ, cm)
         end if
      end if
   end if
end for
```

Figure 14.10: *Pseudocode of the consensus phase.*

Figure 14.11 shows the workflow of our algorithm for a multiple feature map alignment, and the pseudocode is given in Figure 14.12.

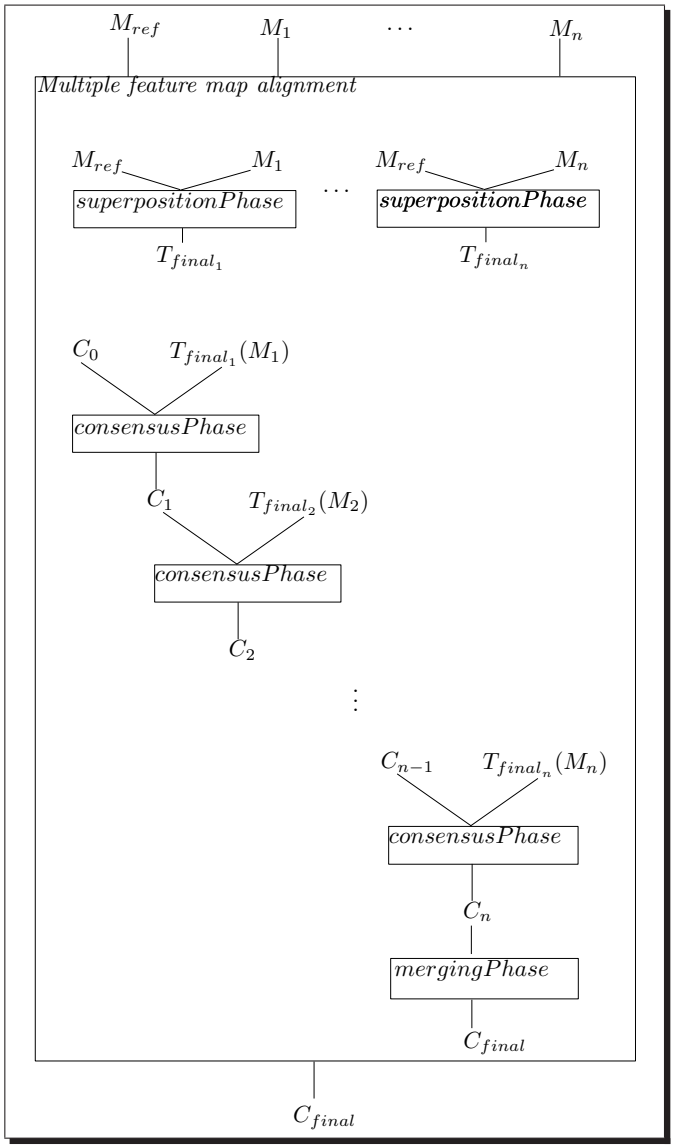

Figure 14.11: *Workflow of the multiple feature map alignment approach. Given $n+1$ maps our algorithm searches for corresponding elements and results in a consensus map.*

Sometimes it may happen that a certain consensus feature is split in two or more consensus features. Figure 14.13 illustrates an example how six elements can be grouped to different consensus features depending on the order in which the elements are processed.

MULTIPLE ALIGNMENT OF FEATURE MAPS

Input: List of element maps $map\_list := \{M_1, \ldots, M_n\}$
Output: Consensus map C

// choose map with highest number of elements as reference map
ref:= indexOfReferenceMap($M_1, \ldots, M_n$)
$\widetilde{M}_{ref} := M_{ref}$
// superposition of the $n+1$ maps
for all maps M_i in map_list (with $i \neq ref$) **do**
  T_i:= ***superpositionPhase***($\widetilde{M}_{ref}, M_i$)
  $\widetilde{M}_i$:= dewarp(M_i, T_i)
end for
// build consensus of the $n+1$ maps
// initialize C_0 with all elements of $\widetilde{M}_{ref}$ as singleton consensus features
C_0:=buildConsensusMap(M_{ref})
// Assign the elements of Map M_i in step i
for all maps $\widetilde{M}_i$ in $\{\widetilde{M}_1, \ldots, \widetilde{M}_n\}$ (with $i \neq ref$) **do**
  C_i = ***consensusPhase***($C_{i-1}, \widetilde{M}_i$)
end for
// merge overlapping consensus features
C_{final}:=mergeConsensusElements(C_n)

Figure 14.12: *Pseudocode of the multiple feature map alignment.*

On the left hand side the cross is chosen as a reference consensus feature, and the sequence "rectangle, plus, filled rectangle, filled circle, circle" results in two different consensus features. However starting with the circle and processing the elements in order "rectangle, filled rectangle, cross, filled circle, plus" yields the desired grouping of the elements in only one consensus feature.

In the following section, we will describe our approach to merge overlapping consensus features that combine elements from different maps.

Merging of consensus features

To determine overlapping consensus features, we developed a simple two step algorithm. Given a consensus map C with n consensus features $\{c_1, \ldots, c_n\}$. Each consensus feature c_i is defined by its consensus RT and m/z position, the set of combined features, and the bounding box $b_{c_i} := ((min_{\mathrm{RT}}, min_{\mathrm{m/z}}), (max_{\mathrm{RT}}, max_{\mathrm{m/z}}))$—spanning a rectangle in $\mathbb{R}^2$ given by the minimal

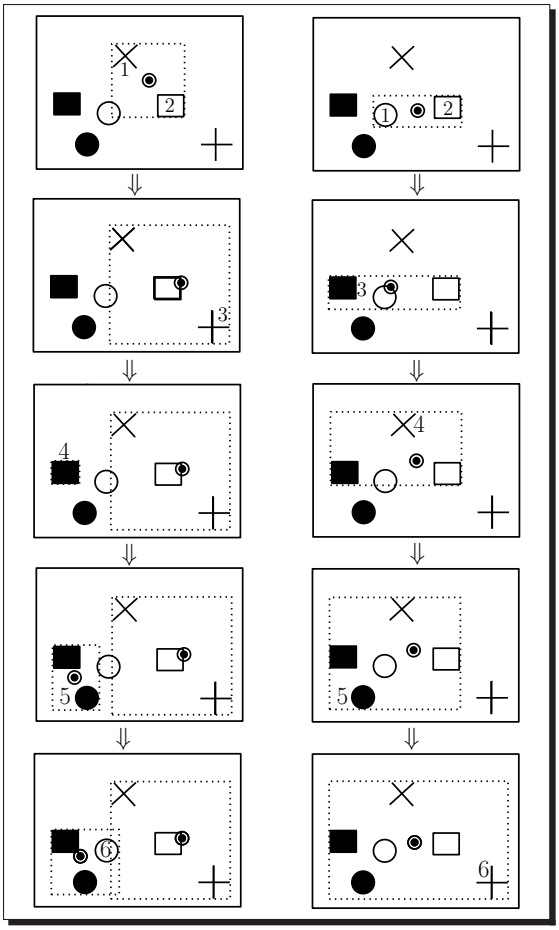

Figure 14.13: *The six different marks represent six corresponding features of six different maps. The result of the six pairwise consensus phases depends on the order in which the elements are processed. The dotted rectangles are the bounding boxes of the consensus features.*

RT and m/z position $(min_{i_{\mathrm{RT}}}, min_{i_{\mathrm{m/z}}})$ and the maximum RT and m/z position $(max_{i_{\mathrm{RT}}}, max_{i_{\mathrm{m/z}}})$. In the first step, we detect overlaps of consensus features in RT. We sort the list $l := (min_{1_{RT}}, max_{1_{RT}}, \ldots, min_{n_{RT}}, max_{n_{RT}})$ of all minimum and maximum RT positions. Afterward, we linearly pass l and store elements that overlap in retention time. In the second step those elements which overlap in RT are searched for those which also overlap in m/z. We merge k overlapping consensus features only if the features in all k consensus features originate from distinct maps. The merging phase has a runtime of $O(n \log n)$.

14.2.5 The MapAlignment TOPP tool

We provide a TOPP [Kohlbacher et al., 2007] application called *MapAlignment* for the MFMAP and the MRMAP, which implements the algorithm proposed in Section 14.2.2 and 14.2.4. The user can either dewarp a set of raw maps given in `mzData` format (see Figure 14.14), or determine the correspondence in multiple feature maps given in `featureXML` format (see Figure 14.15). We developed an own XML format, called `consensusXML`, to facilitate the storage of the consensus map resulting from a multiple feature map alignment.

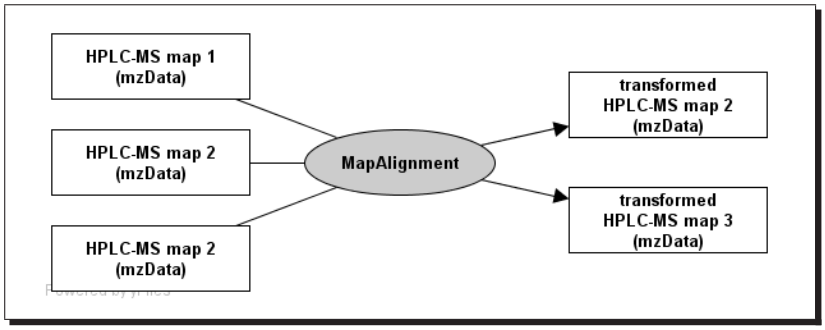

Figure 14.14: *Multiple raw map alignment using the MapAlignment tool.*

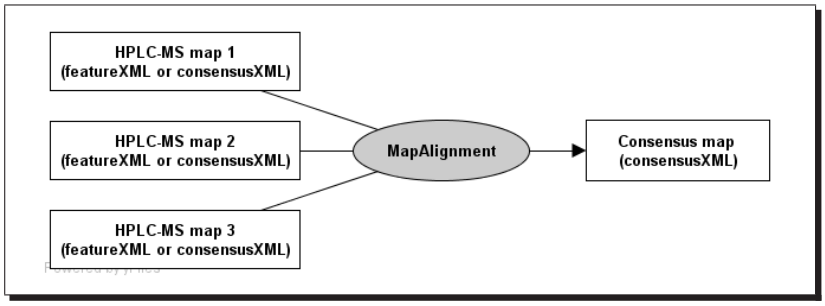

Figure 14.15: *Multiple feature map alignment using the MapAlignment tool.*

All parameters for the superposition and the consensus phase are provided by an XML-based control file. The usage of the tool is described in the TOPP documentation and an example is given in the TOPP tutorial.
The MapAlignment application, as all other TOPP tools, is based on the OpenMS library. We separate the algorithms for multiple raw and feature map alignments into classes for the superposition phase and the consensus phase. The factory design pattern [Gamma et al., 1995] allows us to replace most of the classes with another class implementing the same interface.

Figure 14.16 shows the class diagram of the concerned classes in UML format. The classes are described in the OpenMS documentation and examples of use can be found in the OpenMS tutorial.

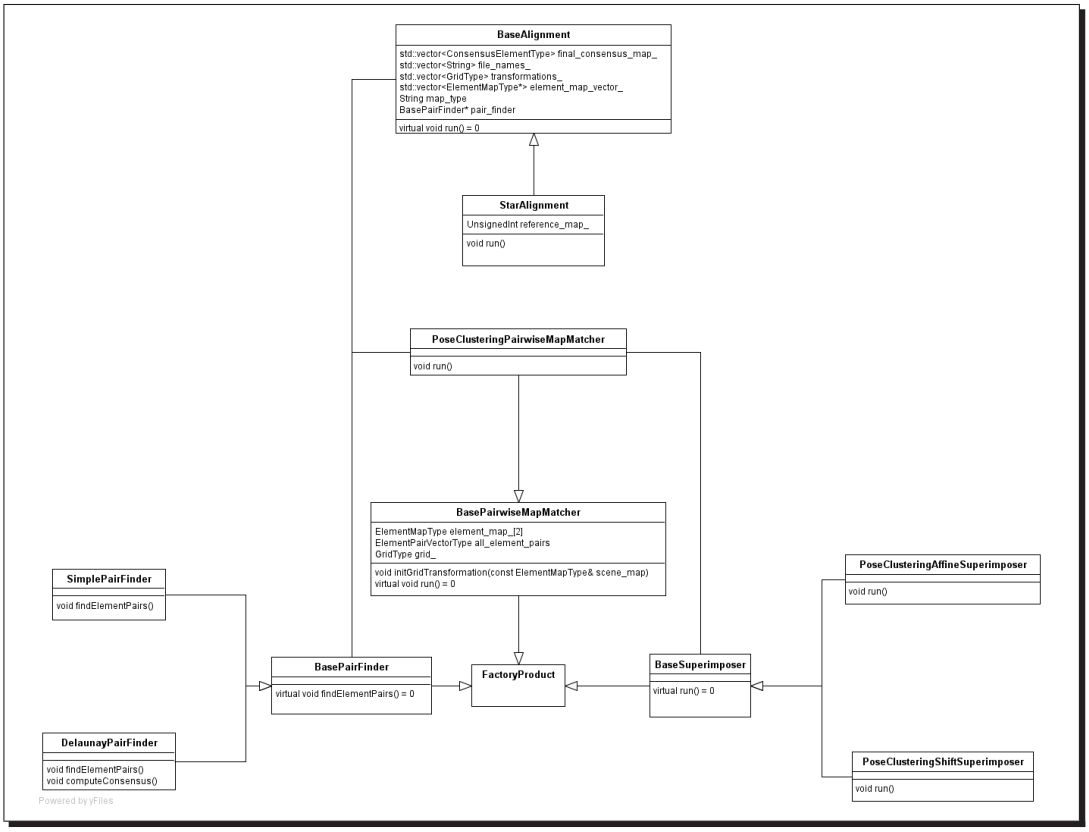

Figure 14.16: *UML class diagram of the classes for multiple feature and raw map alignment.*

Chapter 15

Experiments

Multiple LC-MS raw as well as feature map alignment algorithms should precisely correct the distortion in RT (and m/z), to allow for the assignment of corresponding peptide signals in different maps. Feature map alignment algorithms should additionally group together corresponding features. Besides the variability of the feature positions, time order changes of peptides further complicate the computation of an accurate consensus map. We evaluate both the quality of the transformation for the correction of the RT dimension as well as the quality of the consensus on the basis of annotated feature data. We could also use the distance measure *dsim* to score the accuracy of transformations, but to evaluate the quality of a consensus map we need the information about correspondence in the maps.
Another reason to use annotated data is the circumstance that none of the other feature map alignment tools provide access to the transformation, and some algorithms completely lack any dewarping [Katajamaa et al., 2006].
The consensus phase of our algorithm matches features that are nearest neighbors or at least lie close together. Therefore, the resulting consensus map provides not only information about the quality of the consensus phase, but also information about the quality of the determined transformations.

We compare our algorithm with the alignment algorithms implemented in the freely available software packages msInspect [Bellew et al., 2006], SpecArray [Li et al., 2005], XAlign [Zhang et al., 2005], XCMS [Smith et al., 2006], and MZMine [Katajamaa et al., 2005]. Implementations of the other algorithms proposed in Section 13.2 are either not available [Radulovic et al., 2004; Jaitly et al., 2006], or not usable in their current version [Wang et al., 2007], or only designed for raw map alignment [Bylund et al., 2002; Prakash et al., 2006; Prince and Marcotte, 2006; Listgarten et al., 2007].

15.1 Usage of the different feature map alignment tools

In the following subsections, we will briefly describe how we invoked each tool.

15.1.1 OpenMS alignment algorithm $OpenMS_{MA}$

Our multiple feature map alignment algorithm is implemented in the TOPP tool `MapAlignment`. We call the tool from command line with "`MapAlignment -ini parameters.ini`". The "parameter.ini" is an XML file that contains all parameters for the alignment algorithm.

15.1.2 msInspect alignment algorithm $msInspect_{MA}$

msInspect is a suite of algorithms for the analysis of high-resolution LC-MS proteomics data. The software package is written in the platform-independent language Java and is freely available under http://proteomics.fhcrc.org. We use msInspect on a Windows PC and call the alignment algorithm from command line using
"`java -jar -Xmx512M viewerApp.jar --peptideArray --scanWindow=`Δ`RT --massWindow=`Δ`m/z --out=‘‘consensus_map.tsv’’ ‘‘feature_map_1.tsv’’ ... ‘‘feature_map_n.tsv’’`". We implemented an algorithm that translates our feature map format "featureXML" into the tsv feature map format of msInspect and extracts the consensus map from the msInspect "peptide.tsv" and "peptide.details.tsv" files. The alignment algorithm of msInspect provides the setting of two parameters, which are the maximum size of a consensus feature in time space "scanWindow" and the maximum size of a consensus feature in mass space "massWindow". The option "`--optimize`" is used to determine the best choices for the two parameters with respect to the number of *perfect matches*, which are "true" consensus features, as defined in Definition 12.3.1, and which contain at most one feature of each map.

15.1.3 SpecArray alignment algorithm $SpecArray_{MA}$

The software suite SpecArray offers algorithms for the analysis of LC-MS proteomics data. The algorithms are implemented in C and tested on Linux operating systems. SpecArray is freely available on the website of the SASHIMI project on SourceForge [SourceForge]. We implemented software to convert our feature map format "featureXML" to the binary feature format "pepBof" of SpecArray. To circumvent the conversion of the SpecArray consensus map in Microsoft Excel format and allow for the output in our consensus format, we added some lines of code to the "PepMatch.h" files. The multiple feature map alignment

algorithm is called via command line "`PepMatch -inputfile feature_map_1.pepBof ... feature_map_n.pepBof -outputfile consensus_map.pepBof -paramfile consensus_map.param`". The parameters for the alignment algorithm are hardcoded and cannot be set by the user.

15.1.4 *XAlign*

XAlign [Zhang et al., 2005] is designed as a component of a data analysis pipeline for protein biomarker discovery. The stand-alone executable runs in the Windows command line. It reads tab separated feature lists and generates several output files including the alignment table and peak statistics. *XAlign* was invoked with "`XAlign 1 Δm/z ΔRT 80 datafile.txt`", where `datafile.txt` contains the names of the files to be aligned. The first parameter determines the file type (1=LC/MS Data), the parameters Δm/z and ΔRT define the tolerance in m/z and retention time. The last parameter is of significance for pipeline use only, so it was not changed. The Xalign software is available upon request from the author of [Zhang et al., 2005].

15.1.5 XCMS alignment algorithm $XCMS_{MA}$

XCMS [Smith et al., 2006] is part of Bioconductor [Gentleman et al., 2004], an open source software project for bioinformatics. All Bioconductor packages can be obtained from http://www.bioconductor.org. The XCMS package can be used to process LC/MS and GC/MS data. It includes functionality for visualization, peak picking, non-linear retention time alignment, and relative quantification. XCMS was modified to skip the peak detection step and read peaklists directly from feature map format "featureXML". The alignments were calculated using the `group` function. XCMS also supports a retention time correction step (function `retcor`) but we observed better results when this step was omitted.

15.1.6 MZmine alignment algorithm $MZMine_{MA}$

MZmine [Katajamaa et al., 2006] is a toolbox for processing and visualization of LC/MS data. Due to its implementation in Java, it is platform-independent and it can be downloaded free of charge from `http://mzmine.sourceforge.net`. The source code was slightly modified to allow the import of peak lists instead of raw data files. MZmine offers two alignment algorithms, "slow aligner" and "fast aligner". Due to the better results with multiple alignments, the "slow aligner" was applied.

15.2 Evaluation of the consensus maps

The correspondence between multiple annotated maps can be directly discovered by the assignment of identical identifications in all maps. Using this optimal consensus map, the so-called *ground truth*, we evaluate the performance of each alignment tool. The ground truth contains information about the similarity and difference of peptides in multiple maps; the retention times of corresponding features give additionally information about the variability in RT between the different maps. An optimal alignment algorithm should correct the distortion in RT (and m/z) and contain the same consensus features as the ground truth. The optimal consensus map represented by the ground truth enables the computation of recall and precision values for each alignment algorithm.

15.2.1 Recall and precision of multiple feature map alignment algorithms

Recall and *precision* are evaluation measures frequently used for the performance of information retrieval systems. Given a collection of documents and a query, for which the relevancy of the documents is known, the precision is the proportion of retrieved and relevant documents to all the documents retrieved. However, recall is the proportion of relevant documents that are retrieved out of all relevant documents available.
In our case, a multiple feature map alignment algorithm is our information retrieval system, and the query is represented by the alignment of multiple feature maps.
In 12.3.2, we introduced a convex quality measure *size* for consensus features, which we used to define the MFMAP 12.3. We use this measure to define the "relevant documents" in a multiple feature map alignment. The size of a consensus feature of size n is given by the number of pairwise assignments $\binom{n}{2}$ of the grouped features. All pairwise assignments represented by the consensus features in the ground truth define "relevant documents" in our information retrieval system. The "retrieved documents" are given by the pairwise assignments in the test consensus map determined by the feature map alignment algorithm. Table 15.1 shows the terminology of true positives (TP), false positives (FP), and false negatives (FN) with respect to the comparison of a test consensus map with the ground truth consensus map.

Table 15.1: *Terminology of true positives (TP), false positives (FP), and false negatives (FN) with respect to the comparison of a consensus map with the ground truth.*

	relevant	irrelevant
retrieved	TP	FP
not retrieved	FN	-

The number of pairwise assignments that are represented in the ground truth as well as in the test consensus map defines the true positives (TP). The pairwise assignments of the ground truth that are not found by the feature map alignment algorithm define the false negatives (FN). However, the number of pairwise assignments that are represented by the test consensus map only and that are not contained in the ground truth defines the false positives (FP). The number of pairwise assignments neither represented by the ground truth nor by the test consensus map is zero in our case.

Recall and precision are defined as

$$recall := \frac{TP}{TP+FN} \tag{15.1}$$

$$precision := \frac{TP}{TP+FP} \tag{15.2}$$

Figure 15.1 illustrates TP, FP, and FN on an example ground truth and an exemplary test consensus map. The different markers represent features of five different maps. The optimal consensus map consists of two consensus features of size five and four and the number of "relevant" pairwise assignments is $Rel = \binom{5}{2} + \binom{4}{2} = 16$. However, the number of "received" pairwise assignments in the test consensus map is $Rec = \binom{4}{2} + \binom{3}{2} + \binom{2}{2} = 10$. The 8 true positive pairwise assignments are highlighted by blue edges. The green edges indicate the $FN = Rel - TP = 8$ false negative pairwise alignments, which are not detected by the algorithm. The red edges represent the $FP = Rec - TP = 2$ false positive pairwise alignments in the test consensus map that are not contained in the ground truth. In this example, the alignment algorithm assigned two elements of the same map (two stars), which violates the uniqueness of consensus features in Definition 12.3.1. Accordingly, the test consensus map yields a precision of $\frac{8}{8+2} = 0.8$ and a recall of $\frac{8}{8+8} = 0.5$. The recall of 0.5 states that the alignment algorithm discovers only 50% of the pairwise assignments in the ground truth, and the precision of 0.8 shows that 80% of the detected pairwise assignments are relevant.
An algorithm performs better than another algorithm if its recall and precision values are better.

15.3 Experimental data

In this section, we want to show the performance of our algorithm on two real world data sets. Both data sets are freely available at the Open Proteomics Database [Prince et al., 2004]. The OPD is a public database for storing and disseminating mass spectrometry based proteomics data. The database currently contains roughly $3,000,000$ spectra representing experiments from Escherichia coli, Mycobacterium smegmatis, Saccharomyces cerevisiae, Mus musculus, and Homo sapiens. We pick two data sets resulting from two different exper-

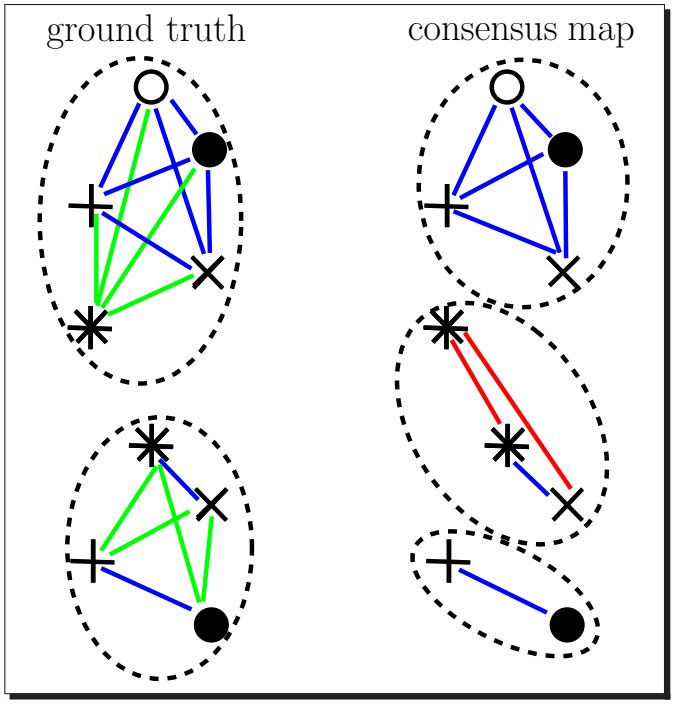

Figure 15.1: *The left figure shows the two consensus elements of a ground truth and the right figure shows three consensus features of a test consensus map determined by an alignment algorithm. The blue edges indicate the true positive pairwise feature assignments, contained in the ground truth as well as in the test consensus map. The green edges show the false negative pairwise alignments and the red edges the false positive pairwise assignments.*

iments from two different organisms, which were already used for the evaluation of OBI-Warp [Prince and Marcotte, 2006]. The first data set results from a dilution series from Escheria coli (E. coli) and the other data set represents different cell states of Mycobacterium smegmatis (M. smegmatis). Both samples are of high complexity and provide typical alignment scenarios.

15.3.1 Sample preparation and LC-LC-MS/MS analysis

We will briefly describe the sample preparation and the two-dimensional high-performance liquid chromatography (LC-LC-MS/MS) analysis of the two experiments. Further information on the E. coli data set can be found on the OPD website and the M. smegmatis experiment is explicitly described in Wang et al. [2005].

Data set *ecoli*: E. coli soluble protein extracts (representing E. coli cells in exponential growth-phase) were diluted in digestion buffer, denatured, and digested with trypsin. Tryptic peptide mixtures were separated by automated LC-LC-MS/MS. The injection quan-

tity of the analyte was altered between the two different runs: *021010_jp32A_15ul_1* and *021016_jp32A_10ul_3*. In *021010_jp32A_15ul_1* 15 μl of the protein extract were analyzed and in *021016_jp32A_10ul_3* only 10 μl. Chromatography salt step fractions were eluted from a strong cation exchange column (SCX) with a continuous 5% acetonitrile (ACN) background and 10-min salt bumps of 0, 20, 40, 60, 80, and 100 mM ammonium chloride. Each salt bump was eluted directly onto a reverse-phase C18 column and washed free of salt. Reverse-phase chromatography was run in and peptides were analyzed online with an ESI ion trap mass spectrometer. In each MS spectrum, the three tallest individual peaks, corresponding to peptides, were fragmented by collision-induced dissociation with helium gas to produce MS/MS spectra. Raw mzXML data and corresponding SEQUEST identification results of *021016_jp32A_10ul_3* and *021010_jp32A_15ul_1* were downloaded from the OPD.

Data set *msmeg*: M. smegmatis soluble protein extracts (representing M. smegmatis cells in different growth-phases) were diluted in digestion buffer, denatured, and digested with trypsin. Tryptic peptide mixtures were separated by automated LC-LC-MS/MS. The three different runs *6-17-03*, *7-17-03*, and *6-06-03* represent protein profiles of a M. smegmatis cell in early, middle, and stationary phase. Chromatography salt step fractions were eluted from a strong cation exchange column (SCX) with a continuous 5% acetonitrile (ACN) background and 10-min salt bumps of 20, 40, 80, and 100 mM ammonium chloride. Each salt bump was eluted directly onto a reverse-phase C18 column and washed free of salt. Reverse-phase chromatography was run in and peptides were analyzed online with an ESI ion trap mass spectrometer. In each MS spectrum, the three tallest individual peaks, corresponding to peptides, were fragmented by collision-induced dissociation with helium gas to produce MS/MS spectra. Raw mzXML data and corresponding SEQUEST identification results of *6-17-03*, *7-17-03*, and *6-06-03* were downloaded from the OPD.

15.3.2 Preprocessing and extraction of peptide features

The raw data were already centroided by the instrument. The poor resolution of ion traps and the insufficient centroiding of the raw data hampers the recognition of isotopic patterns and inhibits a meaningful charge prediction of features. Therefore, we set the charge of all features to a default value of 0 after the feature finding process. Hence, all features are similar with respect to their charge. The processing of the raw data files was performed by the successive application of several TOPP tools: In a first step, we convert all raw data files from mzXML format into mzData format (`FileConverter`). To emphasize the feature signals, we transform each raw data map into a uniformly spaced matrix by bilinear resampling (`LinearResampler`). The spacing of the transformed matrix was 1 Th and 1 second. Afterward, we detect and extract all peptide charge variants in the resampled raw data maps using a feature finding approach (`FeatureFinder`).

The number of features in the resulting feature maps of fraction 0, 20, 40, 60, 80 and 100 of

the *ecoli* data set are shown in Table 15.2. The number of features in the resulting feature maps of fraction 20, 40, 80 and 100 of the *msmeg* data set are shown, respectively, in Table 15.3.

Table 15.2: *The number of features in each of the* 12 *feature maps of the ecoli data set. The features are extracted from the preprocessed raw data of the six fraction with different injection volumes. In brackets the number of features that are annotated with a peptide identification is given.*

	number of all features (number of annotated features)					
injection volume	0	20	40	60	80	100
15 μl	5824 (1282)	1114 (475)	1230 (572)	1902 (765)	1183 (625)	745 (399)
10 μl	4782 (1120)	1021 (575)	958 (519)	1440 (696)	903 (510)	581 (344)

Table 15.3: *The number of features in each of the* 12 *feature maps of the msmeg data set. The features are extracted from the preprocessed raw data of the five fraction with protein profiles in different growth states. In brackets the number of features that are annotated with a peptide identification is given.*

	number of all features (number of annotated features)			
growth state	20	40	80	100
middle	529 (390)	678 (491)	438 (329)	429 (338)
early	557 (346)	520 (296)	524 (332)	545 (412)
stationary	3271 (974)	1483 (835)	474 (374)	401 (304)

15.3.3 Generation of a ground truth

The generation of the "expected consensus map", the so-called *ground truth*, for the *ecoli* and the *msmeg* data sets is performed in three steps using the peptide identifications determined by SEQUEST. To discover the correspondence in different maps we can use only those features in the maps that are annotated with a reliable peptide identification. We use the retention time of the MS/MS scans and the m/z values of the precursor ions to label the features with peptide identifications in a first step. If peptide identifications of an MS/MS scan exist and if the RT value of the scan as well as the m/z value of the precursor ion lie within the convex hull of a feature, the peptide identifications are assigned to the feature. Accordingly, each feature may be labeled with peptide identifications resulting from more than one MS/MS scan. In Tables 15.2 and 15.3, the number of all features in a feature map as well as the number of the labeled features (in parentheses) are given.

Unreliable peptide identifications are filtered out in a second step with respect to the features they are assigned to. The annotation of two features with extremely different RT positions and

the same peptide identification indicates that one or both features are falsely annotated. To discover unreliable identifications, we compute the mean μ and standard deviation σ of the RT positions of all features they are assigned to. If the RT distribution of a peptide identification has a standard deviation greater than 100 s, we remove this identification from all features. Furthermore, we remove peptide identifications from all features with RT positions that do not lie within $[-2\sigma, 2\sigma]$. Furthermore, we also remove unreliable peptide identifications that have an $Xcorr < 1.2$. The Xcorr [Eng et al., 1994] is a specific match score of SEQUEST. It is an absolute measure of spectral quality and closeness of fit of the experimental tandem spectrum to the theoretical tandem spectrum. The closeness is measured by the cross correlation of the two spectra divided by the average of the auto correlation of the experimental spectrum.

Step three is the actual generation of the ground truth. Reliable peptide identifications allow for the determination of the correspondence between the different feature maps under consideration. A correct assignment of features should be emphasized by identical reliable peptide identifications. Using the peptide identifications assigned to the features, we initially determine all possible consensus elements and calculate their score. The score of each consensus feature is given by the sum of the $Xcorr$ values of all peptide identifications that support this certain grouping of features. The higher the score of a feature group, the more reliable is the assignment of those features. A consensus map, composed of all possible feature groups may violate the uniqueness of features in Definition 12.3.1. To solve this problem, we developed a simple iterative strategy reducing all consensus elements to a consensus map C that provides a unique assignment of each feature to only one consensus feature. In the beginning, C is empty and C_{all} is the consensus map composed of all possible feature groups. The filtering of C_{all} is performed iteratively:

1. Take the feature group g that yields the maximum score in C_{all} and add it as consensus feature to the consensus map C.
2. Remove all feature groups in C_{all} that contain at least one feature of g.
3. Iterate step 1) and 2) until C_{all} is empty.

A ground truth is only considered if its number of consensus elements corresponds to at least 10% of the number of annotated features in the aligned feature maps. Table 15.4 shows the number of consensus features in the ground truth C resulting from the SEQUEST identifications and the 12 feature maps (see Table 15.2) resulting from the *ecoli* data set. In Table 15.5, the size of the ground truth resulting from the SEQUEST identifications and the 12 feature maps (see Table 15.3) of the *msmeg* data set are given.

It has to be noted that the "relevant" pairwise assignment represented by the ground truth is incomplete, because we can only discover the correspondence of annotated features. Hence, the determined recall values are accurate, since they are represented by the true positives and

the total number of consensus features in the ground truth. However, the precision values are underestimated, because the true positives are only restricted to the annotated feature; the number of false positives (pairwise assignments that are present in the test consensus map, but not in the ground truth) is overestimated.

Table 15.4: *Number of consensus elements in the six ground truth consensus maps generated for the different fractions in the ecoli data set.*

	number of consensus features					
	0	20	40	60	80	100
ecoli	138	40	72	111	72	50

Table 15.5: *Number of consensus elements in the five ground truth consensus maps generated for the different fractions in the msmeg data set.*

	number of consensus features			
	20	40	80	100
msmeg	161	92	61	64

15.3.4 Sample with different injection volume

The *ecoli* data set represents a typical experimental setting and therefore is perfectly suited for the evaluation of an LC-MS feature map alignment algorithm. The *ecoli* data set represents the proteome of E. coli cells in the exponential growth-phase. The digested protein extract was measured in two different concentrations on six different fractions. The pre-processing procedure of the 12 resulting raw maps as well as the extraction of peptide features was described in detail in the previous section. In the following, we will compare the six LC-MS feature map alignment algorithms $OpenMS_{MA}$ (implemented in OpenMS), $SpecArray_{MA}$ (implemented in SpecArray), $msInspect_{MA}$ (implemented in msInspect), $XCMS_{MA}$ (implemented in XCMS), $MZMine_{MA}$ (implemented in MZMine), and *XAlign* with respect to the feature maps resulting from the *ecoli* data set. For each fraction we align two feature maps, whereby one feature map represents the lower injection volume and the other the higher injection volume of the E. coli cell proteome. The two maps with different injection volume are likely to contain a multiplicity of the same peptides. This similarity of peptides should facilitate the assignment of corresponding peptides. On the other hand, the feature maps are quite complex. Particularly, the feature maps of fraction 0 complicate a proper assignment of features since they contain

around 5000 features in a range of 10 to 5000 s and 300 to 1500 Th.
We evaluate the six consensus maps determined by each alignment algorithm with the ground truth consensus maps that are based on reliable peptide identifications. In Section 15.3.2, we described the procedure to generate a ground truth consensus map given the feature maps to be aligned as well as the corresponding SEQUEST annotations for each fraction. The size of the resulting ground truth consensus maps for each fraction is given in Table 15.4.
We computed recall and precision values of each alignment algorithm based on the six determined consensus maps and the corresponding ground truth consensus maps. The precision values are only given for the sake of completeness since they do not have the same explanatory power as the recall values. As already mentioned in Section 15.3.3 the precision values are underestimated, because true positives are only given for annotated features and the correspondence of the remaining unlabeled features is not known and therefor cannot be evaluated. For most alignment algorithms the user can define the maximal deviation of feature position within a consensus feature given by ΔRT and Δm/z. We optimized these parameters for each tool and set

- $OpenMS_{MA}$: ΔRT := 150 s and Δm/z := 2 Th.
- $msInspect_{MA}$: ΔRT := 250 (defines in this case the number of scans) and Δm/z := 1.5 Th.
- *XAlign*: ΔRT := 180 s and Δm/z := 2 Th.
- $MZMine_{MA}$: ΔRT := 120 s and Δm/z := 1.5 Th.
- $XCMS_{MA}$: ΔRT := 40 s (given by the parameter *bw*) and Δm/z := 1.5 Th.

The alignment algorithm implemented in SpecArray does not provide any parameters that may be defined by the user. Table 15.6 shows the recall and precision values of the six algorithms for the six pairwise feature map alignments in the *ecoli* data set.

In five of six alignments, $OpenMS_{MA}$ clearly outperforms the other alignment algorithms with its high recall values. Only in fraction 100 *XAlign* achieves the same recall value as $OpenMS_{MA}$. Except for the fraction 0 all recall values lie between 0.86 and 0.94. Accordingly, the consensus maps resulting from the OpenMS alignment contain 86 to 94% of the pairwise feature assignments that are given by the ground truth consensus maps. In the consensus map of fraction 0 OpenMS performs slightly worse and achieves only 72% of the expected pairwise feature assignments, but is still better than the other algorithms. Besides $OpenMS_{MA}$ there are three more alignment algorithms that also result in good recall values for the *ecoli* data set. The consensus maps determined by *XAlign* represent 64 to 92% of the pairwise feature assignments in the ground truth maps. However, $MZMine_{MA}$ determined 62 to 89% and $XCMS_{MA}$ 65 to 82%. $msInspect_{MA}$ achieved only 31 to 68% and $SpecArray_{MA}$ 22 to 54% of the expected pairwise feature assignments.

Table 15.6: *Recall and precision values for the six algorithms aligning the feature maps of the ecoli data set.*

fraction 0	$OpenMS_{MA}$	$SpecArray_{MA}$	$msInspect_{MA}$	$MZMine_{MA}$	$XCMS_{MA}$	*XAlign*
recall	**0.72**	0.22	0.31	0.62	0.65	0.64
precision	0.03	0.01	0.01	0.03	0.02	0.02
fraction 20						
recall	**0.88**	0.23	0.35	0.85	0.68	0.73
precision	0.07	0.01	0.00	0.10	0.04	0.05
fraction 40						
recall	**0.86**	0.49	0.46	0.82	0.72	0.74
precision	0.11	0.04	0.01	0.12	0.07	0.08
fraction 60						
recall	**0.94**	0.41	0.60	0.68	0.79	0.75
precision	0.10	0.03	0.02	0.08	0.07	0.08
fraction 80						
recall	**0.94**	0.49	0.54	0.89	0.82	0.82
precision	0.12	0.04	0.01	0.12	0.08	0.10
fraction 100						
recall	**0.92**	0.54	0.68	0.88	0.84	**0.92**
precision	0.12	0.05	0.01	0.12	0.07	0.12

Besides good recall and precision values, an LC-MS feature map alignment algorithm should be fast and thereby allow the alignment of several hundred feature maps in a passable runtime. We compare the runtimes of the six different alignment algorithms on the *ecoli* data set. To provide a fair means of comparison, we measured the user CPU time (total number of CPU-seconds that the process spent in user mode) of $OpenMS_{MA}$, $msInspect_{MA}$, $SpecArray_{MA}$, and $XCMS_{MA}$ alignment on the same PC with 1.8 GHz CPU (Linux operating system) using the GNU 1.7 version of the "time" command. Due to the slow, self-implemented import procedure of feature maps in the alignment algorithm $XCMS_{MA}$, we decided to measure only the runtime of the alignment algorithm itself. However, the runtimes of $OpenMS_{MA}$, $msInspect_{MA}$, and $SpecArray_{MA}$ include the I/O process. Since the MZMine alignment algorithm can only be invoked from the GUI and not from command line, we measured the runtimes of the six pairwise alignments with a common stop watch. *XAlign* ran in a VMWare (Workstation 5.5.2 build-29772), where the GNU time command in a cygwin shell did not yield correct measurements. Manual wall clock time measurements indicated same run time order of magnitude as the other algorithms. In Table 15.7 the runtimes of the six alignment algorithms on the *ecoli* data set are given.

Besides the remarkable recall values, our algorithm did also achieve favorable runtimes. In most of the considered fractions, our alignment algorithm is faster than the other tools. Except

Table 15.7: *Runtimes of the six alignment algorithms on the ecoli data set measured as the total user CPU time using the GNU "time" command. The mark [a] indicates that the runtime of the algorithm was measured with a stop watch, [b] gives the user CPU of the alignment algorithm without I/O. XAlign[c] ran in a VMWare (Workstation 5.5.2 build-29772), where the GNU time command in a cygwin shell did not yield correct measurements. Manual wall clock time measurements indicated same run time order of magnitude as the other algorithms.*

	$OpenMS_{MA}$	$SpecArray_{MA}$	$msInspect_{MA}$	$MZMine^{a}_{MA}$	$XCMS^{b}_{MA}$	$XAlign^{c}$
fraction 0	76.67	8.34	14.87	24	12.42	n/a
fraction 20	2.66	5.52	9.14	2	5.97	n/a
fraction 40	3.24	74.91	8.73	2	5.86	n/a
fraction 60	6.64	5.49	10.08	2	7.61	n/a
fraction 80	3.32	8.74	8.56	2	5.51	n/a
fraction 100	3.28	7.71	8.36	2	3.71	n/a

for the alignment of the relative complex feature maps of fraction 0, our algorithm took only 2.66 to 6.64 s for the computation of a consensus map. $MZMine_{MA}$ determined the consensus map of fraction 0 in around 24 s and took only 2 s for all other pairwise alignments. However, $XCMS_{MA}$ that achieved similarly good recall values as $MZMine_{MA}$, took 3.71 to 7.61 s for each alignment (12.42 s for the alignment of fraction 0) without measuring the consideration of the I/O process.
This experiment proves the applicability of our method on ordinary data of medium complexity. We showed its performance on a typical alignment scenario, where the injection volume was altered between two LC-LC-MS/MS runs. We yielded the highest recall values in all pairwise alignments and are also faster than the other methods. In the experiment considered in the following section, the emphasis is placed on the alignment of feature maps representing different biological variations.

15.3.5 Different biological state

The *msmeg* data set provides also a typical alignment case, but with another emphasis than the *ecoli* data set. The *msmeg* data set represents a test of biological variation. It contains LC-LC-MS/MS measurements of the M. smegmatis proteome extracted from cells in three different growth-phases. Digested protein extract of the early, the middle, as well as the stationary phase on four different fractions was measured. The pre-processing procedure of the 12 resulting raw maps as well as the extraction of peptide features was the same as for the *ecoli* data set and was described in detail in Section 15.3.2. We again compare the six LC-MS feature map alignment algorithms $OpenMS_{MA}$ (implemented in OpenMS), $SpecArray_{MA}$ (implemented in SpecArray), $msInspect_{MA}$ (implemented in msInspect), $XCMS_{MA}$ (implemented in XCMS), $MZMine_{MA}$ (implemented in MZMine), and $XAlign$ with respect to the feature maps resulting from the *msmeg* data set. For each fraction we align three feature maps, whereby each feature

map represents the M. smegmatis proteome in a different cell growth-phase. The alignment of the *msmeg* data set constitutes a more difficult problem than the *ecoli* data set, since the proteome of cells in different growth-phases may share only a small fraction of common proteins. We evaluate the four consensus maps determined by each alignment algorithm with the ground truth consensus maps that are based on reliable peptide identifications. In Section 15.3.2, we described the procedure to generate a ground truth consensus maps given the feature maps to be aligned as well as the corresponding SEQUEST annotations for each fraction. The size of the resulting ground truth consensus maps for each fraction is given in Table 15.5.
We computed recall and precision values of each alignment algorithm based on the four determined consensus maps and the corresponding ground truth consensus maps. The precision values are only given for the sake of completeness since they do not have the same explanatory power as the precision values. As already mentioned in Section 15.3.3, the precision values are underestimated, because true positives are only given for annotated features and the correspondence of the remaining unlabeled features is not known and therefor cannot be evaluated. For most alignment algorithms the user can define the maximal deviation of feature position within a consensus feature given by ΔRT and Δm/z. We optimized these parameters for each tool and set

- $OpenMS_{MA}$: ΔRT := 200 s and Δm/z := 2 Th.
- $msInspect_{MA}$: ΔRT := 300 (defines in this case the number of scans) and Δm/z := 1.5 Th.
- *XAlign*: ΔRT := 180 s and Δm/z := 2 Th.
- $MZMine_{MA}$: ΔRT := 120 s and Δm/z := 1.5 Th.
- $XCMS_{MA}$: ΔRT := 40 s (given by the parameter *bw*) and Δm/z := 1.5 Th.

The alignment algorithm implemented in SpecArray does not provide any parameters that may be defined by the user. Table 15.8 shows the recall and precision values of the six algorithms for the four feature map alignments in the *msmeg* data set.

Our alignment algorithm again achieves high recall values. The percentage of correctly discovered pairwise feature assignments lay between 60 and 79 for the fractions 20, 40, and 60 and is higher than the recall values of the other tools. However, $OpenMS_{MA}$ failed to align the three feature maps of fraction 80 and discovered only 12 % of the expected pairwise feature assignments. The alignment of the three feature maps of fraction 80 poses a hard problem for all other tools and was not solved satisfyingly by any other tool. SpecArray achieved the highest recall value for fraction 80, but discovers only 49 % of the pairwise feature assignments given by the ground truth consensus map. Besides this fraction, SpecArray did not result in a recall higher than 0.54. *XALign*, $MZMine_{MA}$ and $XCMS_{MA}$ are, as in the *ecoli* data set, ranked behind our alignment approach and achieved recall values between $0.44 - 0.72$, $0.56 - 0.68$ and

Table 15.8: *Recall and precision values for the six algorithms aligning the feature maps of the msmeg data set.*

fraction 20	$OpenMS_{MA}$	$SpecArray_{MA}$	$msInspect_{MA}$	$MZMine_{MA}$	$XCMS_{MA}$	*XAlign*
recall	**0.79**	0.23	0.30	0.68	0.70	0.72
precision	0.16	0.01	0.02	0.15	0.01	0.16
fraction 40						
recall	**0.60**	0.49	0.09	0.56	0.47	0.44
precision	0.08	0.04	0.01	0.10	0.09	0.06
fraction 80						
recall	0.12	**0.49**	0.31	0.25	0.25	0.28
precision	0.06	0.04	0.02	0.06	0.06	0.05
fraction 100						
recall	**0.76**	0.54	0.39	0.59	0.57	0.71
precision	0.09	0.05	0.04	0.09	0.09	0.09

0.47 – 0.70 respectively. The three algorithms also failed to align fraction 80. The alignment algorithm of msInspect did not exceeded a recall of 0.39.

The OpenMS alignment algorithm again outperforms the other tools not only with its high recall values, but also with its fast runtime. Runtime measurements were taken with caveat as described on page 156. In the Table 15.9 the runtimes of the six alignment algorithms on the *msmeg* data set are given. The manual wall clock time measurements for *XAlign* indicated same run time order of magnitude as the other algorithms.

Table 15.9: *Runtimes of the six alignment algorithms on the msmeg data set. For details, see Table 15.7.*

	$OpenMS_{MA}$	$SpecArray_{MA}$	$msInspect_{MA}$	$MZMine^{a}_{MA}$	$XCMS^{b}_{MA}$	$XAlign^{c}$
fraction 20	3.74	282.38	9.94	55	11.27	n/a
fraction 40	2.31	37.39	9.52	3	9.43	n/a
fraction 80	1.12	28.21	7.99	2	6.43	n/a
fraction 100	1.09	66.58	8.15	2	2.89	n/a

Our approach is consistently faster than the rest and took only 1.09 to 3.74 s for the computation of a consensus map. Method $MZMine_{MA}$ needed 55 s to compute a consensus map of fraction 20, but all other alignments took only 2 to 3 s. However, $XCMS_{MA}$ that achieved similarly good recall values as $MZMine_{MA}$, took 2.89 to 9.43 s for the alignment of fraction 40, 80 and 100. The runtime of the alignment of fraction 20 was also much slower with 11.27 s.
The *msmeg* data set represents also a typical but more complex alignment scenario than the *ecoli* data set. We proved once more the applicability of our algorithm to real-world data, where its precise and quick alignments outperform the results of the other alignment approaches. In the following section we will prove the robustness of the six alignment methods with simulated data.

15.4 Robustness analysis with simulated data

In the last section we proved the applicability of our algorithm on two typical alignment scenarios. Our approach yielded for both data sets with different emphasis high recall values and fast runtimes and outperforms the alignment algorithms $SpecArray_{MA}$, $msInspect_{MA}$, $XCMS_{MA}$, $MZMine_{MA}$, and *XAlign*. In the following sections, we will prove the robustness of our algorithm in the presence of local distortions (Section 15.4.3) and will show that our alignment approach is also robust in aligning feature maps, which share only a small fraction of common elements (Section 15.4.4).
To analyze the robustness of our alignment algorithm implemented in OpenMS and the other five approaches we generate a so-called *original feature map* from the *protein mix* data set described in the following section. Afterward we generate warped copies of the original feature map. Thereby, the warp is composed by a 2D affine transformation and an additive Gaussian error in both dimensions. To test the performance of each algorithm in the presence of noise, we vary the degree of noise added to the features' RT and m/z positions in Section 15.4.3. In the second experiment described in Section 15.4.4 we evaluate the applicability of the methods on an alignment scenario given by Multidimensional Protein Identification Technology [Lin et al., 2001] experiments. To this end, we vary the number of corresponding features in the original feature map and its noisy copies.

15.4.1 Sample preparation and LC-MS analysis

Protein mix: A tryptic digested protein mix of ten known proteins (beta-Casein, conalbumin, myelin, hemoglobin, hemoglobin, albumin, leptin, creatine, alpha1-Acid-Glycoprotein and bovine serum albumin). LC separation was performed on a capillary column (monolithic polystyrene/-divinylbenzene phase, 60 mm x 0.3 mm) with 0.05% trifluoroacetic acid (TFA) in water (eluent A) and 0.05% TFA in acetonitrile (eluent B). Separation was achieved at a flow of 2.0 μl/min at 50 °C with an isocratic gradient of 0–25% eluent B over 7.5 min. Eluting peptides were detected in a TOF mass spectrometer (microTOF from Bruker, Bremen, Germany) equipped with an electrospray ion source.

15.4.2 Preprocessing and extraction of peptide features

The data set resulting from the experimental procedure described above is of high resolution, i.e., single isotopic peaks for charges up to four can easily be distinguished and the LC-MS maps take up to 1 GB disk space per run. We reduce the complexity by summarizing groups of data, which point to single peaks using our wavelet-based peak picking algorithm described in Chapter 8.
Afterward, we create lists of features for each data set by grouping clusters of isotopic peaks

that appear in consecutive scans. The charge of each feature is determined by fitting a theoretical isotope model based on the average composition of a peptide for a given mass as proposed earlier [Schulz-Trieglaff et al., 2007]. The 10 proteins give rise to about 195 features in total. Mass and retention time were measured with very high precision.

In the next section we use this simple data set as the original feature map and show the robustness of six different alignment algorithms in comparison to our approach.

15.4.3 Alignment of noisy LC-MS maps

In this first robustness analysis we want to assess the ability of all alignment algorithms to match corresponding peptides in the presence of noise and consequential changes in elution order. As already mentioned, Jaitly et al. [2006] noticed that the distortion in RT is composed by a global trend and local effects of less understood factors normally distributed around an ideal elution time. We analyze the robustness of the six alignment tools in the presence of noise with respect to two experiments *varySigma*$_{RT}$ and *varySigma*$_{m/z}$. In both experiments we model the warp in RT and m/z by a global affine transformation and pose the local effects by an additive local Gaussian error.

We use the original feature map generated in Section 15.4.2 and test up to which extent of local distortion in RT and m/z the different algorithms are able to precisely solve the MFMAP. The 2D position $(\mathrm{RT}(f), \mathrm{m/z}(f))$ of each feature f in the original map is shifted by a transformation $T : \mathbb{R}^2 \to \mathbb{R}^2$. The global trend of T is given by an affine transformation with scaling matrix $A \in \mathbb{R}^{2\times 2}$ and translation vector $b \in R^2$ and the local effects are simulated by an additive Gaussian error $(\varepsilon_{\mathrm{RT}}, \varepsilon_{\mathrm{m/z}})$ with $\varepsilon_{\mathrm{RT}} \sim N(0, \sigma^2_{\mathrm{RT}})$ and $\varepsilon_{\mathrm{m/z}} \sim N(0, \sigma^2_{\mathrm{m/z}})$. Thus the transformed feature position $(\mathrm{RT}'(f), \mathrm{m/z}'(f))$ of a feature f is given by

$$\begin{pmatrix} \mathrm{RT}'(f) \\ \mathrm{m/z}'(f) \end{pmatrix} := T \begin{pmatrix} \mathrm{RT}(f) \\ \mathrm{m/z}(f) \end{pmatrix} = \begin{pmatrix} a_{\mathrm{RT}} & 0 \\ 0 & a_{\mathrm{m/z}} \end{pmatrix} \begin{pmatrix} \mathrm{RT}(f) \\ \mathrm{m/z}(f) \end{pmatrix} + \begin{pmatrix} t_{\mathrm{RT}} \\ t_{\mathrm{m/z}} \end{pmatrix} + \begin{pmatrix} \varepsilon_{\mathrm{RT}} \\ \varepsilon_{\mathrm{m/z}} \end{pmatrix}. \qquad (15.3)$$

Due to uncertainties in measurement, the feature maps of repeated measurements may not be identical and share only a fraction of corresponding features. To model this situation and thereby achieve a more realistic setting, we first generated warped copies of the original map using the transformation T. In a second step, we replaced some of the distorted features with random features. These random features were inserted within the bounding box of the remaining distorted features in the warped copies. The corresponding features in all maps—the original feature map and its warped copies—define the *ground truth* consensus map, which is used to determine recall and precision of all alignment algorithms.

For most alignment algorithms the user can define the maximal deviation of feature position within a consensus feature given by ΔRT and Δm/z. We optimized these parameters for each tool and set

- $OpenMS_{MA}$: ΔRT := 120 s and Δm/z := 0.5 Th.
- $msInspect_{MA}$: ΔRT := 150 (defines in this case the number of scans) and Δm/z := 1 Th.
- *XAlign*: ΔRT := 180 s and Δm/z := 2 Th.
- $MZMine_{MA}$: ΔRT := 120 s and Δm/z := 1.5 Th.
- $XCMS_{MA}$: ΔRT := 40 s (given by the parameter *bw*) and Δm/z := 1.5 Th.

The alignment algorithm implemented in SpecArray does not provide any parameters that may be defined by the user.

In the first experiment $varySigma_{RT}$ we vary the standard deviation of the local error in RT and analyze the performance of the alignment algorithms with respect to the resulting recall and precision values. The global linear trend of T in Equation 15.3 was given by $a_{\mathrm{RT}} \sim N(1, 0.2)$, $b_{\mathrm{RT}} \sim N(100, 50)$, $a_{\mathrm{m/z}} = 1$, $b_{\mathrm{m/z}} = 0$, and a fixed standard deviation $\sigma_{\mathrm{m/z}} = 0.1$ Th for the error distribution that models the local distortion in m/z. The varying local distortion in RT was modeled by seven different standard deviations $\sigma_{\mathrm{RT}} \in \{5\text{ s}, 10\text{ s}, 20\text{ s}, 30\text{ s}, 40\text{ s}, 50\text{ s}, 60\text{ s}\}$. For each value of σ_{RT} we generate 10 test sets, each consisting of the original feature map and 100 warped copies. The warped copies and the original feature map share a fraction of 70% corresponding features.

Unfortunately, $SpecArray_{MA}$ could not manage the alignment of 101 maps. The computation of all pairwise alignments leads to a quadratic blow-up in runtime; apparently, the complexity of the implementation is even worse because we had to cancel the unfinished alignment of 101 after 24 h. Accordingly, we created 10 particular test sets for each σ_{RT}, which contain beside the original feature map only 5 warped copies instead of 100.

The increasing local distortion in RT should reflect severe problems of the LC system. The increasing influence of the local distortion reduces the global characteristic of the transformation T in Equation 15.3. However, in real data sets (e.g., see Figure 14.7) we observed that the global trend mainly characterizes the warp and accordingly the standard deviation σ_{RT} lies between 10 to 20 s.

Figures 15.2 to 15.7 show box whisker plots of the recall and precision values of the different alignment algorithms for varying σ_{RT} values. Our alignment approach (see Figure 15.2) yielded the best precision values and 81 to 86% of the pairwise feature assignments in the consensus map are "relevant" and represented in the ground truth feature maps. We precisely model the global linear trend of the warp T, but with increasing σ_{RT} the local distortion dominates and accordingly the mean precision of $OpenMS_{MA}$ decreased. However, considering a typical degree of the local distortion given by a standard deviation of 5 to 30 s the determined

consensus maps contain 75 to 80 % of the "relevant" pairwise feature assignments.
$msInspect_{MA}$ (see Figure 15.3), which estimates a global linear trend plus a non-linear component, as well as $SpecArray_{MA}$ (see Figure 15.4), which also models a non-linear trend of the warp in RT precisely determine the distortion in RT. The achieved recall values are relatively constant for the varying σ_{RT} values and are given by $0.81 - 0.83$ and $0.82 - 0.85$, respectively. However, both alignment algorithm result in relatively poor precision values and many of the pairwise feature assignments given by the resulting consensus maps are false positives. The consensus maps determined by $msInspect_{MA}$ contains only about 59 % of the "relevant" pairwise feature assignments and $SpecArray_{MA}$ discovers between 46 to 48 % of the expected assignments. Regarding the recall and values of $SpecArray_{MA}$ we have to consider that the values are based on smaller test data sets.
The other three alignment algorithms $XCMS_{MA}$ (see Figure 15.5), *XAlign* (see Figure 15.6), and $MZMine_{MA}$ (see Figure 15.7) all resulted in low recall as well as low precision values.

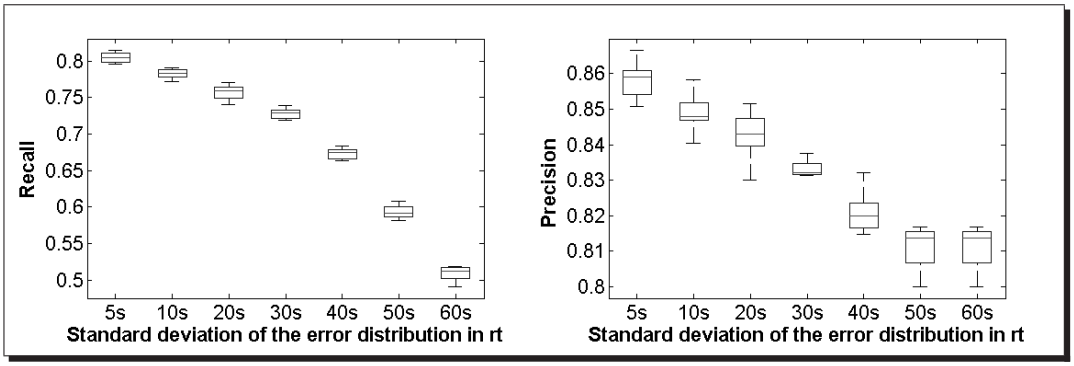

Figure 15.2: *Box whisker plots of the recall and precision values of the alignment algorithm of OpenMS for varying σ_{RT} values.*

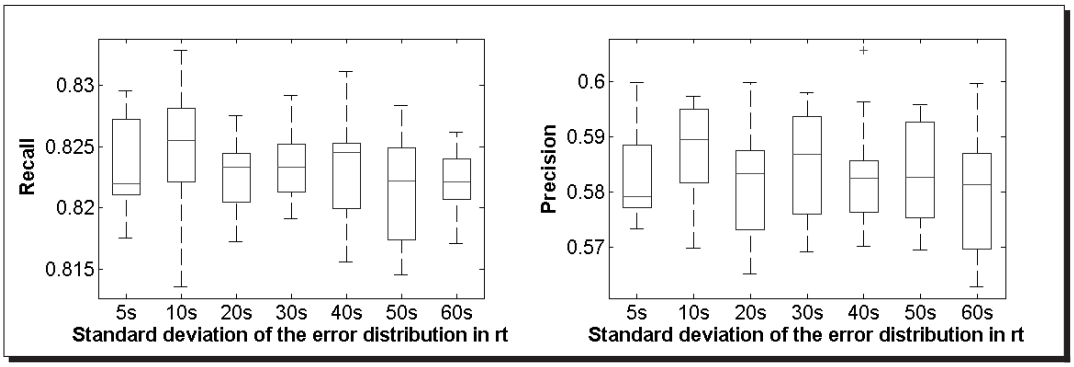

Figure 15.3: *Box whisker plots of the recall and precision values of the alignment algorithm of msInspect for varying σ_{RT} values.*

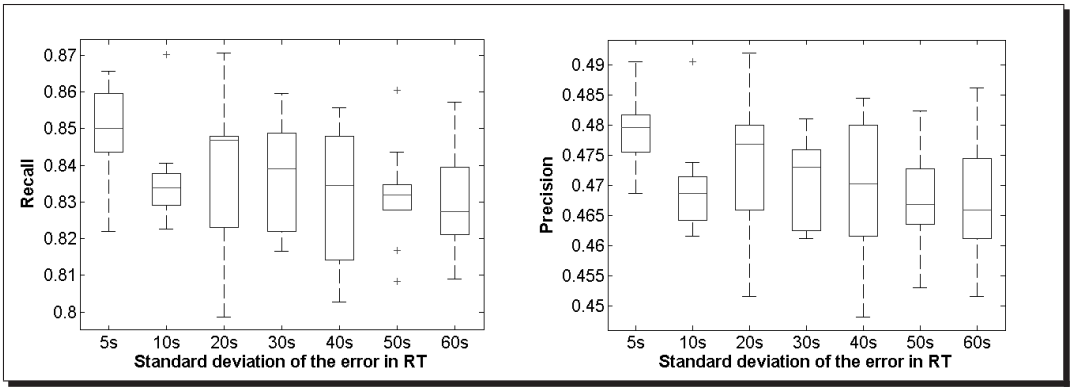

Figure 15.4: *Box whisker plots of the recall and precision values of the alignment algorithm of SpecArray for varying σ_{RT} values.*

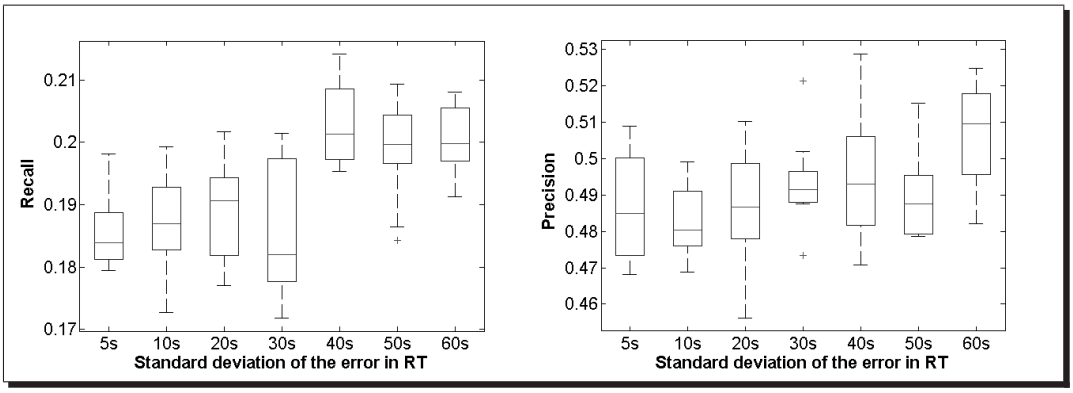

Figure 15.5: *Box whisker plots of the recall and precision values of the alignment algorithm of XCMS for varying σ_{RT} values.*

Figure 15.8 illustrates the recall and precision values of all alignment algorithms. It represents a so-called *precision-recall diagram* for the six alignment algorithms. Each curve is given by the mean precision and mean recall values determined for the six different queries ($\sigma_{\mathrm{RT}} \in \{5\mathrm{s}, 10\mathrm{s}, 20\mathrm{s}, 30\mathrm{s}, 40\mathrm{Th}, 50\mathrm{s}, 60\mathrm{s}\}$).

In Lange et al. [2007] we showed that using a standard deviation of 30 s comes up with almost 40 % peptide time order changes within two maps. The number of these permutations increases with the standard deviation of the noise in the RT dimension. This is due to certain characteristics of the data. Even if the 10 protein mixture is not too complex, it is relatively dense and consequently, the extracted peptide features lie closely together. Small disturbances in RT will already result in features moving even closer together, larger ones will result in peptides changing their elution order. Therefore, this data set is particularly well suited to assess

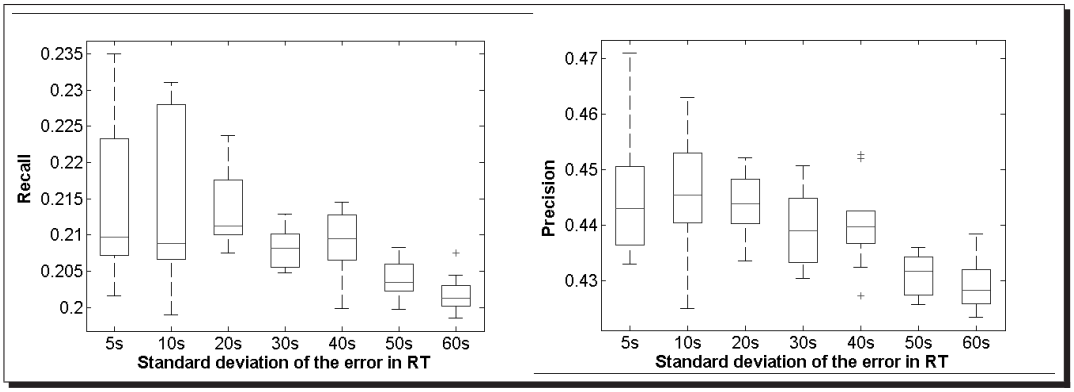

Figure 15.6: *Box whisker plots of the recall and precision values of the alignment algorithm XAlign for varying σ_{RT} values.*

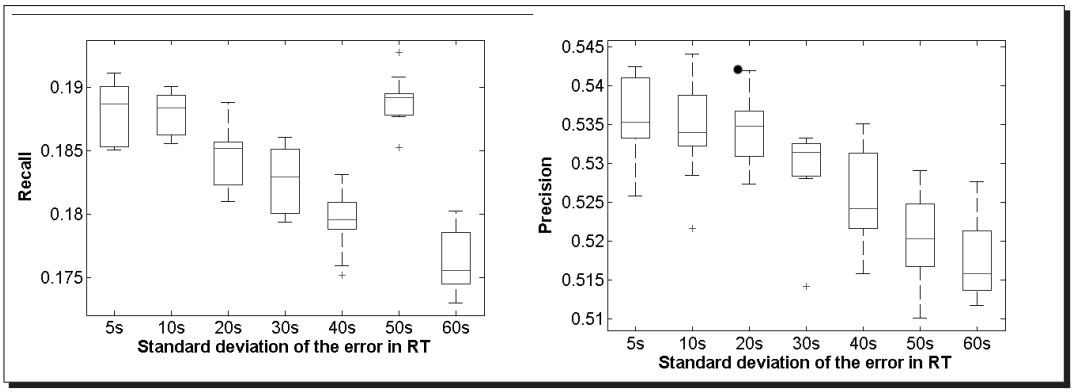

Figure 15.7: *Box whisker plots of the recall and precision values of the alignment algorithm of MZMine for varying σ_{RT} values.*

the performance of an alignment algorithm in these situations.

Table 15.10 shows the runtimes of the six alignment algorithms on the data set $varySigma_{RT}$. The runtime for each σ_{RT} was averaged over the 10 test sets. Runtime measurements were taken with caveat as described on page 156. The manual wall clock time measurements for *XAlign* indicated same run time order of magnitude as the other algorithms.

Our approach outperforms the alignment of msInspect and SpecArray since it took only around 10 s for the alignment of the 101 feature maps. $msInspect_{MA}$ needed the tenfold runtime with around 122 s. Actually, $SpecArray_{MA}$ took 25 to 29 s for the reduced test sets including six feature maps. Although, the runtimes of $MZMine_{MA}$, and $XCMS_{MA}$ are faster they may be neglected due to their low recall and precision values.

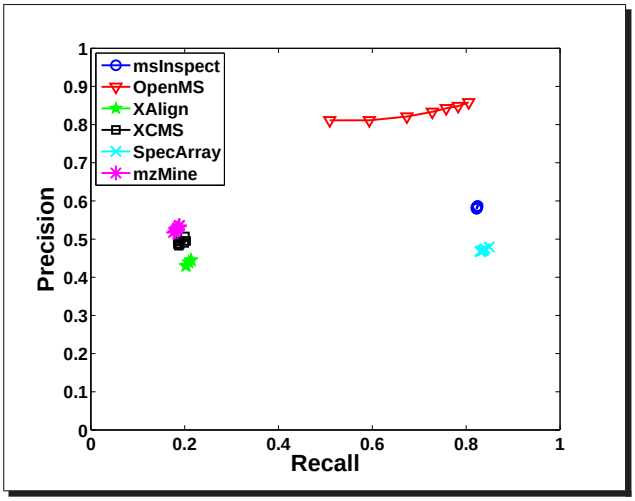

Figure 15.8: *Precision-recall diagram for the six alignment algorithms in experiment varySigma$_{RT}$. The six curves show the mean precision and mean recall values determined for the seven different queries ($\sigma_{RT} \in \{5s, 10s, 20s, 30s, 40s, 50s, 60s\}$).*

Table 15.10: *Runtimes (averaged over the 10 runs for each σ_{RT}) of the six alignment algorithms on the varySigma$_{RT}$ data set for details see Table 15.7.*

σ_{RT} (s)	$OpenMS_{MA}$	$SpecArray_{MA}$	$msInspect_{MA}$	$MZMine^{a}_{MA}$	$XCMS^{b}_{MA}$	$XAlign^{c}$
5	10.03	24.74	121.63	2.44	1.58	n/a
10	10.07	24.84	121.80	2.44	1.59	n/a
20	10.17	26.33	121.81	2.46	1.59	n/a
30	10.10	27.00	121.87	2.46	1.60	n/a
40	10.34	26.03	121.39	2.47	1.62	n/a
50	10.23	26.77	121.71	2.49	1.60	n/a
60	10.69	28.65	122.13	2.49	1.60	n/a

In the second experiment *varySigma$_{m/z}$* we test the alignment algorithms with respect to their ability to align feature maps generated with varying strength of distortion in m/z. We again renounce a global linear trend in m/z and model the warp by local distortions only. Local distortions in m/z may result from a poorly calibration or may be introduced by an insufficient preprocessing of the data. We use the global linear trend T with $a_{\text{RT}} \sim N(1, 0.2)$, $b_{\text{RT}} \sim N(100, 50)$, $a_{\text{m/z}} = 1$, $b_{\text{m/z}} = 0$, and a local distortion in RT with $\sigma_{\text{RT}} = 15$s. For the standard deviation of the error distribution in m/z we use six different values $\sigma_{\text{m/z}} \in \{0.01\text{Th}, 0.05\text{Th}, 0.1\text{Th}, 0.2\text{Th}, 0.3\text{Th}, 0.4\text{Th}\}$. For each value of $\sigma_{\text{m/z}}$ we generate 10 test sets, each consisting of the ground truth and 100 warped copies. Due to the high runtime of $SpecArray_{MA}$ we created again 10 extra test sets for each $\sigma_{\text{m/z}}$, which contain besides the ground truth feature map only 5 warped copies instead of 100.

A standard deviation of 15 s is realistic and we could observe this deviation in several real world examples. The values for the standard deviation in m/z reflect also realistic settings. High resolution mass spectrometers like FT-ICR or QTOF instruments yield a precision of 5 ppm or higher. Hence, the m/z positions of corresponding features in two feature maps should be less than 0.01 Th comprising the error introduced by the peak picking and feature finding process. However, a standard deviation of 0.4 Th reflects poorly resolved data and imprecise peak picking and feature finding steps. It is doubtful, if these data can be used for a quantitative analysis at all, but we want to exhaust the alignment algorithms and show their limitations.

Figures 15.9 to 15.14 show box whisker plots of the recall and precision values of the different alignment algorithms for varying $\sigma_{\text{m/z}}$ values. The alignment approaches implemented in OpenMS, SpecArray, and msInspect result in very good recall for a standard deviation up to 0.2 Th. $SpecArray_{MA}$ achieved the highest recall values that lie between 0.8 and 0.88, but those values were determined on a reduced data set. $msInspect_{MA}$ resulted in average in recall values of 0.75 to 0.82 and $OpenMS_{MA}$ yielded in average a recall of 0.75 to 0.79. For $\sigma_{\text{m/z}} > 0.2$ Th all recall values rapidly fell of to values about 0.5. The same behavior can be observed with the precision values. The precision values of our approach are overall very high and the mean precision values lie between 0.76 and 0.85, but they also rapidly decrease for $\sigma_{\text{m/z}} > 0.2$ Th. The precision values of $msInspect_{MA}$ fell of linearly and lie in average between 0.49-0.61. However, $SpecArray_{MA}$ achieved in average only values between 0.3-0.5. The other three alignment algorithms $XCMS_{MA}$ (see Figure 15.12), $XAlign$ (see Figure 15.13), and $MZMine_{MA}$ (see Figure 15.14) resulted once again in low recall and precision values.

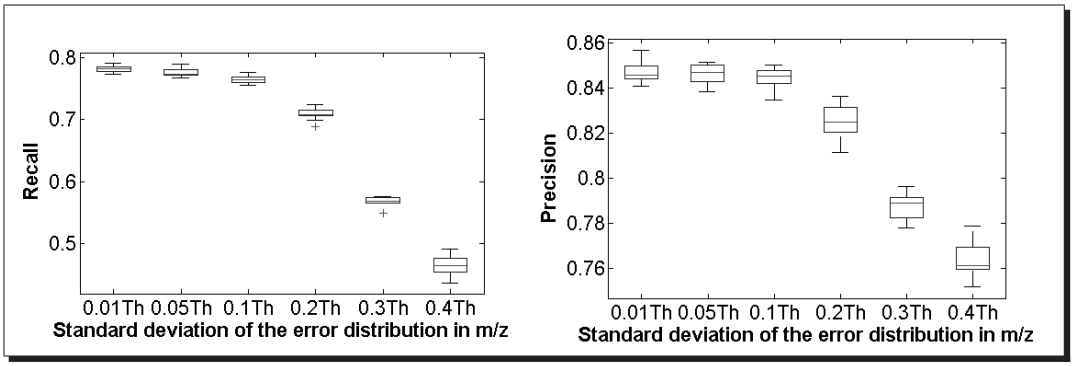

Figure 15.9: *Box whisker plots of the recall and precision values of the alignment algorithm of OpenMS for varying $\sigma_{\text{m/z}}$ values.*

Figure 15.15 shows the precision-recall diagram for the six alignment algorithms. Each curve is given by the mean precision and mean recall values determined for the six different queries ($\sigma_{\text{m/z}} \in \{0.01\text{Th}, 0.05\text{Th}, 0.1\text{Th}, 0.2\text{Th}, 0.3\text{Th}, 0.4\text{Th}\}$).

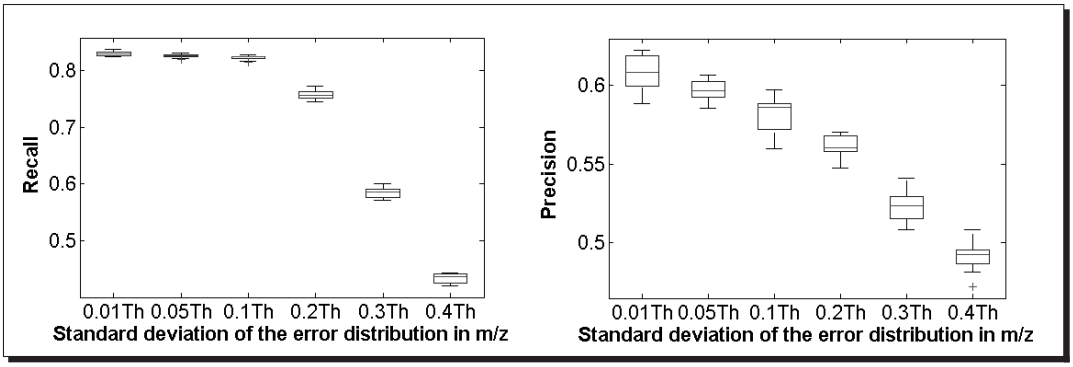

Figure 15.10: *Box whisker plots of the recall and precision values of the alignment algorithm of msInspect for varying $\sigma_{m/z}$ values.*

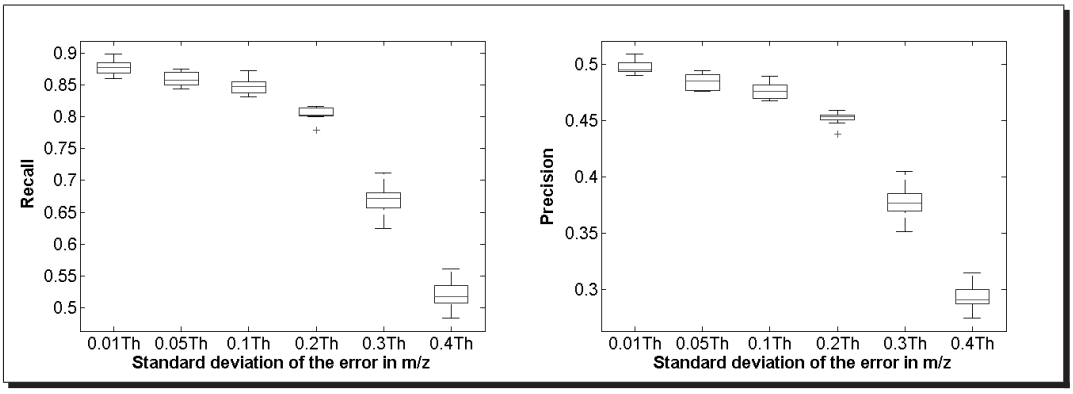

Figure 15.11: *Box whisker plots of the recall and precision values of the alignment algorithm of SpecArray for varying $\sigma_{m/z}$ values.*

Table 15.11 shows the runtimes of the six alignment algorithms on the data set *varySigma*$_{m/z}$. The runtime for each $\sigma_{m/z}$ was again averaged over the 10 test sets. Runtime measurements were taken with caveat as described on page 156. The manual wall clock time measurements for *XAlign* indicated same run time order of magnitude as the other algorithms.

The runtimes are similar to Table 15.11. We again outperformed the alignment of msInspect and SpecArray and took only around 10 s for the alignment of the 101 feature maps.

Our approach performed well even on noisy data. It precisely and quickly aligned feature maps when the distortion of the RT and m/z dimension is mainly defined by a global trend. The number of falsely discovered pairwise feature assignments determined by the OpenMS alignment was in both experiments very low.

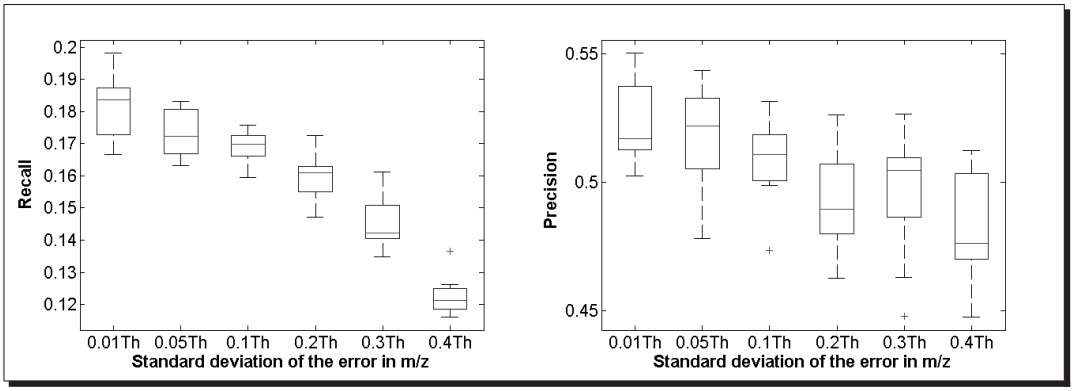

Figure 15.12: *Box whisker plots of the recall and precision values of the alignment algorithm of XCMS for varying $\sigma_{m/z}$ values.*

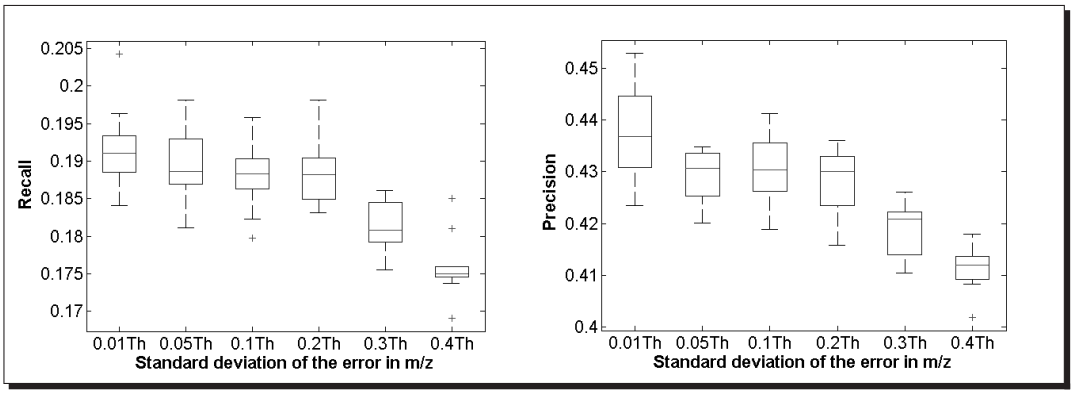

Figure 15.13: *Box whisker plots of the recall and precision values of the alignment algorithm XAlign for varying $\sigma_{m/z}$ values.*

Table 15.11: *Runtimes (averaged over the 10 runs for each $\sigma_{m/z}$) of the six alignment algorithms on the varySigma$_{m/z}$ data set for details see Table 15.7.*

$\sigma_{m/z}$ (Th)	$OpenMS_{MA}$	$SpecArray_{MA}$	$msInspect_{MA}$	$MZMine^{a}_{MA}$	$XCMS^{b}_{MA}$	$XAlign^{c}$
0.01	10.03	17.59	125.69	2.23	1.62	n/a
0.05	10.09	27.39	125.68	2.24	1.62	n/a
0.1	10.01	28.78	121.57	2.25	1.63	n/a
0.2	10.15	29.92	117.28	2.26	1.63	n/a
0.3	10.11	35.86	115.20	2.40	1.64	n/a
0.4	10.10	39.54	115.40	2.42	1.67	n/a

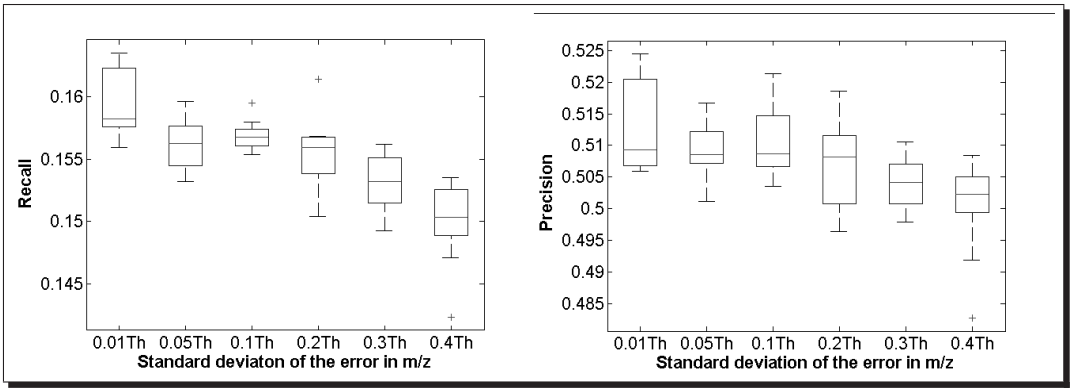

Figure 15.14: *Box whisker plots of the recall and precision values of the alignment algorithm of MZMine for varying $\sigma_{m/z}$ values.*

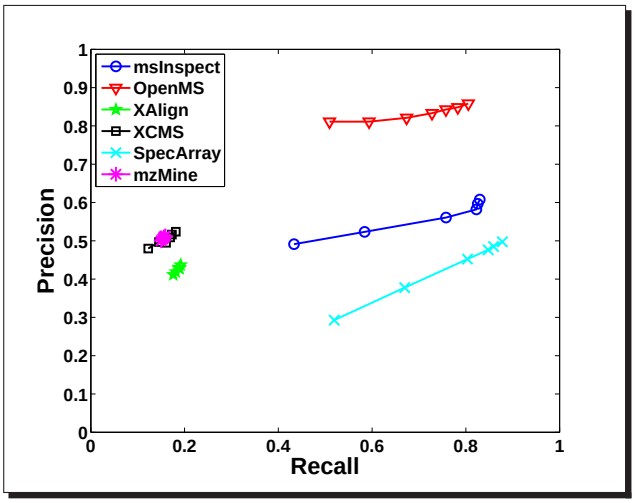

Figure 15.15: *Precision-recall diagram for the six alignment algorithms in experiment varySigma$_{m/z}$. The six curves show the mean precision and mean recall values determined for the six different queries ($\sigma_{m/z} \in \{0.01Th, 0.05Th, 0.1Th, 0.2Th, 0.3Th, 0.4Th\}$).*

15.4.4 Aligning maps with little overlap

A third important issue in the performance evaluation of an alignment algorithm is the ability to align LC-MS maps with little overlap such as maps obtained from different sample fractions in a Multidimensional Protein Identification Technology (MudPIT) [Lin et al., 2001] experiment. In these experiments, complex peptide mixtures are separated using 2D liquid chromatography. That is, several chromatographic columns are coupled and the separation proceeds in several steps.

The LC-MS data acquired in these experiments results in several sample fractions that are mostly distinct regarding the contained peptide but also share a set of corresponding peptides. The size of this common peptide set depends on the column technology. Another application of alignment algorithms is to create the superset of the peptides contained in the sample fractions for further processing. To achieve this, peptides occurring in several fractions need to be found and used to compute an accurate alignment.

To assess the performance of our approach in a MudPIT experiment, we vary now the number of common features in the ground truth feature map and the warped copies. We computed alignments for changing numbers of random features and again compute recall and precision values for all six alignment algorithms.

In the experiment *varyFraction* we keep the standard deviation of the local error in RT and m/z fixed. We again use the transformation T composed by a global linear trend plus a local error as defined in Equation 15.3. The parameters of T were set to $a_{\mathrm{RT}} \sim N(1, 0.2)$, $b_{\mathrm{RT}} \sim N(100, 50)$, $a_{\mathrm{m/z}} = 1$, $b_{\mathrm{m/z}} = 0$. The local distortion in RT is 15 s and in m/z 0.1 Th. We generate maps using five different percentage values ρ of overlap between the original feature map and the warped copies $\rho \in \{100\%, 80\%, 60\%, 40\%, 20\%\}$. For each value of ρ we again generate 10 test sets, each consisting of the ground truth and 100 warped copies. We again created 10 extra test sets for $SpecArray_{MA}$, each consisiting of only 6 feature maps instead of 101.

Figures 15.16 to 15.21 show box whisker plots of the recall and precision values of the different alignment algorithms for varying numbers of corresponding features in the original feature map and its warped copies.
The recall values of $OpenMS_{MA}$ stayed relatively constant with 0.76 to 0.78 for the varying percentage of common features and also the average precision values are relatively robust and fell off only slightly from 0.85 to 0.75 until $\rho = 40$ %. For $\rho = 20$ % our approach even yielded a mean precision of 0.55. The recall of the five other alignment algorithms remained also relatively constant, but with an increasing number of random features the number of false positive pairwise feature assignments in the consensus maps increased and thus all precision curves fell off sharply. The alignment algorithm implemented in msInspect achieved, on average, a recall of 0.81 to 0.82. However, the precision values are much smaller than those determined by our approach. Given 101 feature maps—whereby all features of the original feature map are represented in the 100 warped copies—$msInspect_{MA}$ yielded only a mean precision of 0.6 and fell off to 0.42 at $\rho = 40$ %. $SpecArray_{MA}$ achieved high mean recall values of around 0.82 to 0.84. The decrease of its average precision values is percipitous. Whereas the mean precision at $\rho = 100$ % is 0.82. $SpecArray_{MA}$ yielded a precision of only 0.2 for $\rho = 40$ %. The other three alignment algorithms $XCMS_{MA}$ (see Figure 15.19), *XAlign* (see Figure 15.20), and $MZMine_{MA}$ (see Figure 15.21) all resulted in low recall as well as low precision values.

Figure 15.15 shows the precision-recall diagram for the six alignment algorithms. Each curve is given by the mean precision and mean recall values determined for the five differ-

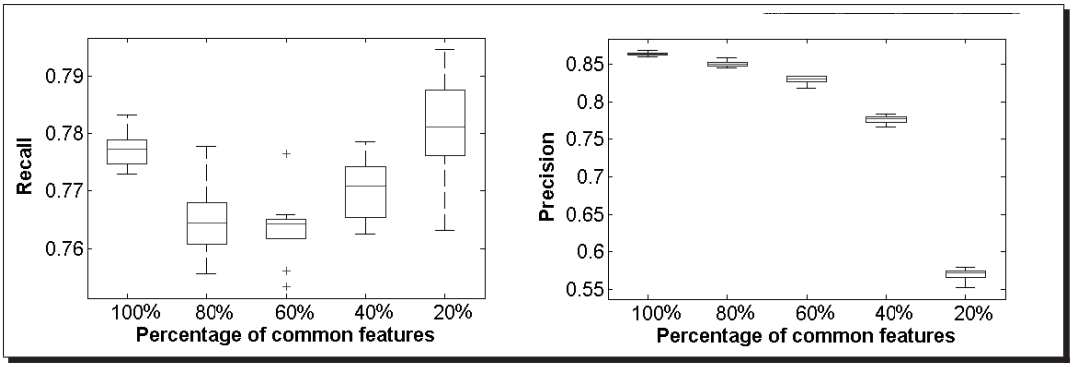

Figure 15.16: *Box whisker plots of the recall and precision values of the alignment algorithm of OpenMS for a varying number of common features.*

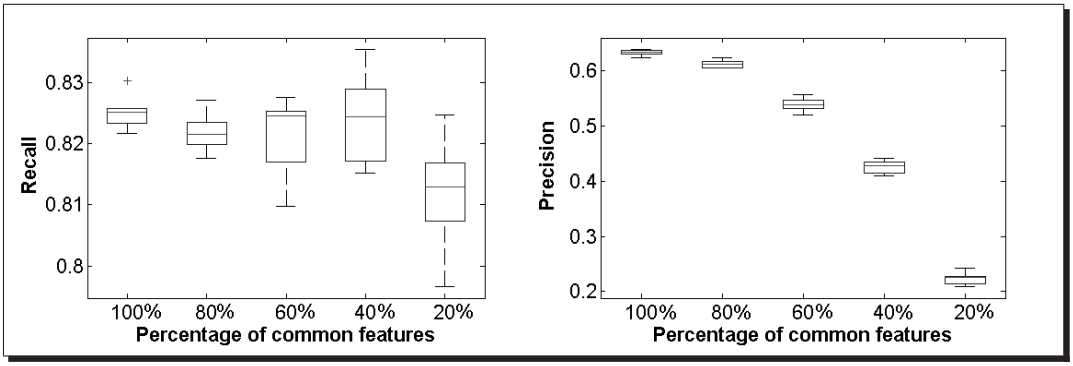

Figure 15.17: *Box whisker plots of the recall and precision values of the alignment algorithm of msInspect for a varying number of common features.*

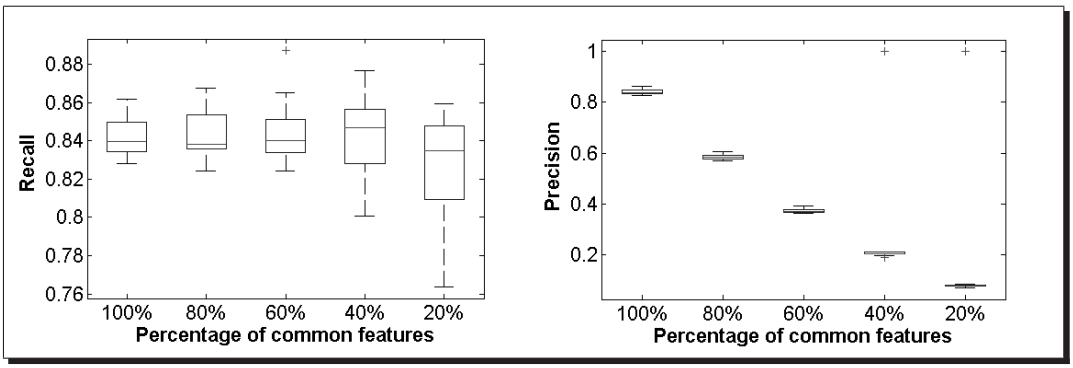

Figure 15.18: *Box whisker plots of the recall and precision values of the alignment algorithm of SpecArray for a varying number of common features.*

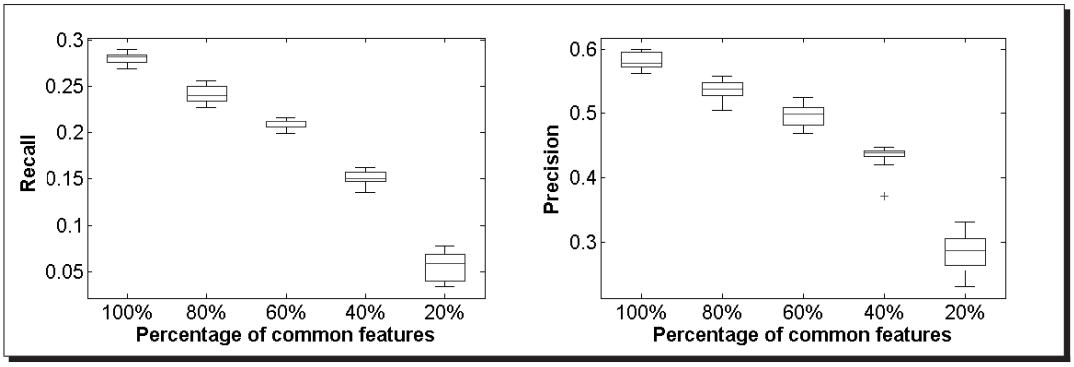

Figure 15.19: *Box whisker plots of the recall and precision values of the alignment algorithm of XCMS for a varying number of common features.*

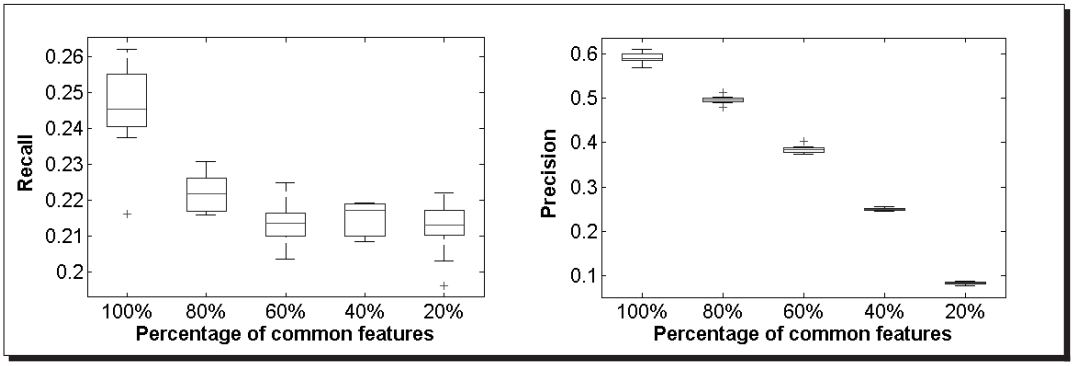

Figure 15.20: *Box whisker plots of the recall and precision values of the alignment algorithm of XAlign for a varying number of common features.*

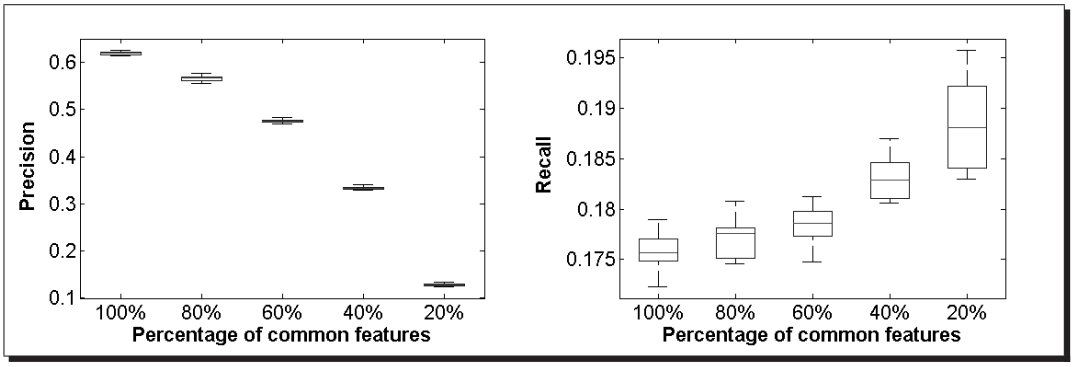

Figure 15.21: *Box whisker plots of the recall and precision values of the alignment algorithm of MZMine for a varying number of common features.*

ent queries ($\rho \in \{100\%, 80\%, 60\%, 40\%, 20\%\}$).

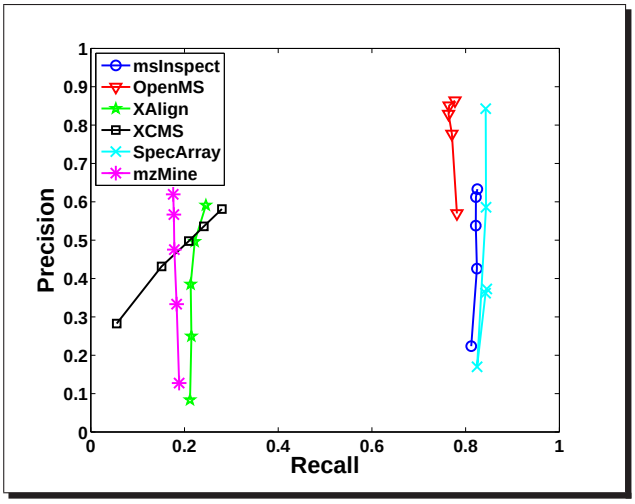

Figure 15.22: *Precision-recall diagram for the six alignment algorithms in experiment varyFraction. The six curves show the mean precision and mean recall values determined for the five different queries ($\rho \in \{100\%, 80\%, 60\%, 40\%, 20\%\}$).*

Table 15.12 shows the runtimes of the six alignment algorithms on the data set *varyFraction*. We averaged the runtime for each ρ over the 10 test sets. Runtime measurements were taken with caveat as described on page 156. The manual wall clock time measurements for *XAlign* indicated same run time order of magnitude as the other algorithms.

Table 15.12: *Runtimes (averaged over the 10 runs for each ρ) of the six alignment algorithms on the varyFraction data set for details see Table 15.7.*

ρ (%)	$OpenMS_{MA}$	$SpecArray_{MA}$	$msInspect_{MA}$	$MZMine^{a}_{MA}$	$XCMS^{b}_{MA}$	$XAlign^{c}$
100	6.64	32.04	125.20	2.61	1.38	n/a
80	13.81	27.41	122.63	2.40	1.60	n/a
60	12.27	25.89	119.75	2.48	1.61	n/a
40	11.19	36.40	117.79	2.51	1.53	n/a
20	9.18	85.79	115.63	2.61	1.43	n/a

Our approach outperforms the alignment algorithms $msInspect_{MA}$ and $SpecArray_{MA}$ since it took only 6.18 to 13.81 s for the alignment of the 101 feature maps. However, $msInspect_{MA}$ needed again the tenfold runtime with around 120 s. Actually, $SpecArray_{MA}$ required 25.89 to 85.79 s for the reduced test sets including only six feature maps. Although, the runtimes of $MZMine_{MA}$, and $XCMS_{MA}$ are faster they may be neglected due to their low recall and precision values.

We are aware that the evaluation of algorithms on simulated data has its caveats. Nonetheless, these experiments allow us to assess the performance of our method on data with specific characteristics. It is also not clear if our model for the distortion of RT and m/z coordinates comes close to perturbations in real experiments. But affine warps as introduced in these experiments are frequently observed in practice. Note that we sampled Gaussian distributed noise for each feature independently. This results in distortions that are more severe than one would expect in real-world data. In a real large-scale experiment, one would expect locally correlated perturbations in RT and systematic shifts in subsets of the LC-MS maps. Since we introduce noise into a real sample, and not an entirely artificial one, our data already incorporates this phenomenon to a certain extent. We further aggravate these drifts by applying our noise model to m/z and RT and by doing so, we can estimate the robustness of our algorithm and its ability to handle changes in the elution order of peptides, something, which is impossible for algorithms based on dynamic time warping.

Chapter 16

Discussion and conclusion

The automatic alignment of LC-MS data sets is an important step in every high-throughput proteomics experiment. Algorithms that can perform this task efficiently and accurately have a huge potential for basic research in biology but also for more applied questions such as biomarker discovery and drug research in general. We have presented an alignment technique that is able to precisely and quickly align multiple LC-MS raw or feature maps. Its independence of the processing stage of the LC-MS data to which it is applied, makes it flexible and applicable to any kind of data from upcoming LC-MS technologies and processing algorithms. Our geometric approach precisely solves the multiple raw map problem and aligns multiple LC-MS maps in feasible time. The LC-MS raw or feature maps are aligned in a star-like manner and superposed using an adapted pose clustering algorithm. An additional step was implemented to solve the multiple feature map alignment problem. It precisely and quickly groups the corresponding features in the superposed maps and determines the resulting consensus map.

We compared the recall, precision, and runtime of our algorithms with those of five other feature map alignment algorithms analyzing two real world data sets as well three two simulated data sets. Our approach outperforms the other alignment approaches on both real data sets representing typical alignment scenarios. By means of the simulated data sets we proved the robustness of our approach in the presence of noise and its applicability to maps with little overlap, e.g., given by Multidimensional Protein Identification Technology [Lin et al., 2001] experiments. In all experiments, our algorithm was the fastest and achieved the best recall values as well as good precision values.

In the real data we considered so far, the RT distortion was composed by a major global linear trend and a minor additive local effect. As we have seen, our algorithm performs well as long as the global trend prevails. If the local error gains influence on the warp in RT the affine trend modeled by our approach is not able to precisely estimate the distortion in RT. To increase the

precision of our algorithm even for those data a more sophisticated regressions and mapping functions may be incorporated. Due to the modular architecture of our algorithm and OpenMS in general this could be done effortlessly.
Different chromatographic fractions may result in maps with only a little overlap. The alignment of such maps may be improved upon a progressive alignment approach. Besides our alignment algorithm, we defined a sophisticated distance measure for LC-MS maps that will allow for the development of such a progressive alignment approach.

Our raw and feature map alignment algorithm is implemented in the OpenMS framework. Based on the alignment classes in OpenMS we also implemented an easy-to-use application for "The OpenMS Proteomics Pipeline (TOPP)" application `MapAlignment`. OpenMS is freely-available to the bioinformatics community from www.openms.de.

Chapter 17

Availability and requirements of the OpenMS/TOPP project

Project home page: `http://www.openms.de`
Operating system(s): Platform-independent (OpenMS can be compiled on most Unix-like platforms using an ANSI C++- compliant compiler)
Programming language: C++
Other requirements: Qt 4.1 or higher, OpenMS contrib package
License: GNU Lesser General Public License (LGPL)
Any restrictions to use by non-academics: see LGPL license
Documentation: The class documentation is available in HTML format. The OpenMS tutorial and the TOPP tutorial are available in HTML/PDF format.

Chapter 18

Glossary

Deisotoping
: Deisotoping is needed for identifying isotopic peak groups that belong to the same organic specimen.

Deconvolution
: Charge state deconvolution determines the actual charge of the analyte that gave rise to a certain peak (or isotopic peak group as a whole).

Extracted ion chromatogram (EIC)
: Chromatogram created by plotting the intensity of the signal observed at a chosen m/z value in a series of mass spectra recorded as a function of RT.

Feature
: The two-dimensional signal created by some chemical entity (e.g., a peptide). A feature is characterized by its isotopic pattern in mass-to-charge dimension and by the elution profile in retention time dimension.

Mass spectrum
: Plot of ion abundance versus m/z.

Mass spectral peak
: A mass spectral peak is a localized maximum signal in a mass spectrum created by some chemical entity (e.g., a peptide).

Multidimensional Protein Identification Technology (MudPIT)
: MudPIT is a technique for the separation and identification of complex protein and peptide mixtures. MudPIT separates peptides using 2D liquid chromatography. In this way, the separation can be interfaced directly with the ion source of a mass spectrometer.

Parts per million (ppm)

The mass accuracy is often expressed in parts per million.

Total ion chromatogram (TIC)

The chromatogram produced from an LC-MS experiment, which is the sum of all the intensities of the individual ions at each time interval in the experiment.

References

R. Aebersold and M. Mann. Mass spectrometry-based proteomics. *Nature*, 422(6928):198–207, 2003.

M. Alexandrov, N. K. L.N. Gall, V. Nikolaev, V. Panvlenko, V. Shkurov, G. Baram, M. Grachev, V. Knorre, and Y. Kusner. *Bioorganicheskaya Khimiya (Russian Journal of Bioorganic Chemistry)*, 10:710–712, 1984.

B. K. Alsberg, A. M. Woodward, and D. B. Kell. An Introduction to Wavelet Transforms for Chemometricians: A Time-Frequency Approach. *Chemom. Intell. Lab. Syst.*, 37(2):215–239, 1997.

H. Alt and L. J. Guibas. Discrete geometric shapes: Matching, interpolation, and approximation: A survey. Technical Report B 96-11, Freie Universität Berlin, Department of Mathematics and Computer Science, 1996. URL `citeseer.ist.psu.edu/alt96discrete.html`.

V. Andreev, T. Rejtar, H.-S. Chen, E. V. Moskovets, A. R. Ivanov, and B. L. Karger. A Universal Denoising and Peak Picking Algorithm for LC-MS Based on Matched Filtration in the Chromatographic Time Domain. *Anal. Chem.*, 75(22):6314–6326, 2003.

T. M. Annesley. Ion suppression in mass spectrometry. *Clin Chem*, 49(7):1041–1044, 2003.

F. Aurenhammer. Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Comput. Surv.*, 23(3):345–405, 1991.

A. Bairoch and R. Apweiler. The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. *Nucleic Acids Res.*, 28(1):45–8, 2000.

D. Ballard. Generalizing the Hough Transform to Detect Arbitrary Shapes. *Pattern Recognition*, 13(2):111–122, 1981.

M. Bellew, M. Coram, M. Fitzgibbon, M. Igra, T. Randolph, P. Wang, D. May, J. K. Eng, R. Fang, C. Lin, J. Chen, D. Goodlett, J. Whiteaker, A. G. Paulovich, and M. McIntosh. A suite of algorithms for the comprehensive analysis of complex protein mixtures using high-resolution LC-MS. *Bioinformatics (Oxford, England)*, 22(15):1902–1909, 2006.

P. Berndt, U. Hobohm, and H. Langen. Reliable automatic protein identification from matrix-assisted laser desorption/ionization mass spectrometric peptide fingerprints. *Electrophoresis*, 20:3521–3526, 1999.

P. R. Bevington and D. K. Robinson. *Data reduction and error analysis for the physical sciences*. Mcgraw-Hill Higher Education, 2002. ISBN 0-0711-9926-8.

C. Bielow. Preprocessing of LC-MS data. Research Project at the Freie Universität Berlin, 2006.

K. Biemann. Mass spectrometry of peptides and proteins. *Annu. Rev. Biochem.*, 61:977–1010, 1992.

B. Bisle, A. Schmidt, B. Scheibe, C. Klein, A. Tebbe, J. Kellermann, F. Siedler, F. Pfeiffer, F. Lottspeich, and D. Oesterhelt. Quantitative Profiling of the Membrane Proteome in a Halophilic Archaeon. *Molecular & Cellular Proteomics*, 5(9):1543–1558, 2006.

J.-D. Boissonnat, O. Devillers, M. Teillaud, and M. Yvinec. Triangulations in CGAL (extended abstract). In *SCG '00: Proceedings of the sixteenth annual symposium on Computational geometry*, pages 11–18, New York, NY, USA, 2000. ACM Press.

P. V. Bondarenko, D. Chelius, and T. A. Shaler. Identification and relative quantitation of protein mixtures by enzymatic digestion followed by capillary reversed-phase liquid chromatography-tandem mass spectrometry. *Anal. Chem.*, 74(18):4741–4749, 2002.

J. B. Breen, G. Hopwood, Femia, K. L. Williams, and M. R. Wilkins. Automatic poisson peak harvesting for high throughput protein identification. *Electrophoresis*, 21:2243–2251, 2000.

R. Bro. Parafac : tutorial and applications. *Chemom. Intell. Lab. Syst.*, 33:149–171, 1997.

L. G. Brown. A survey of image registration techniques. *ACM Comput. Surv.*, 24(4):325–376, 1992. ISSN 0360-0300.

D. Bylund, R. Danielsson, G. Malmquist, and K. E. Markides. Chromatographic alignment by warping and dynamic programming as a pre-processing tool for PARAFAC modelling of liquid chromatography mass spectrometry data. *J Chromatogr A*, 961(2):237–244, 2002.

B. Cañas, D. Lòpez-Ferrer, A. Ramos-Fernàndez, E. Camafeita, and E. Calvo. Mass spectrometry technologies for proteomics. *Briefings in functional genomics & proteomics*, 4(4): 295–320, 2006.

C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*. URL `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

J. S. Choudhary, W. P. Blackstock, D. M. Creasy, and J. S. Cottrell. Matching peptide mass spectra to EST and genomic DNA databases. *Trends Biotechnol.*, 19(10 Suppl):S17–S22, 2001.

F. S. Collins, E. D. Green, A. E. Guttmacher, M. S. Guyer, and U. S. N. H. G. R. Institute. A vision for the future of genomics research. *Nature*, 422(6934):835–847, 2003.

K. R. Coombes, S. Tsavachidis, J. S. Morris, K. A. Baggerly, and H. M. Kuerer. Improved Peak Detection and Quantification of Mass Spectrometry Data Acquired from Surface-Enhanced Laser Desorption and Ionization by Denoising Spectra with the Undecimated Discrete Wavelet Transform. *Proteomics*, 5:4107–4117, 2005.

R. Craig and R. C. Beavis. TANDEM: matching proteins with tandem mass spectra. *Bioinformatics (Oxford, England)*, 20(9):1466–1467, 2004.

M. E. Csete and J. C. Doyle. Reverse engineering of biological complexity. *Science*, 295 (5560):1664–1669, 2002.

V. Dančík, T. A. Addona, K. R. Clauser, and J. E. Vath. De novo peptide sequencing via tandem mass spectrometry: a graph-theoretical approach. pages 135–144, 1999.

J. F. de Cossio, L. J. Gonzalez, Y. Satomi, L. Betancourt, Y. Ramos, V. Huerta, A. Amaro, V. Besada, G. Padron, N. Minamino, and T. Takao. Isotopica: a tool for the calculation and viewing of complex isotopic envelopes. *Nucleic Acids Rearch*, 32:674–678, 2004.

E. de Hoffmann, J. Charette, and V. Stroobant. *Mass Spectrometry: Principles and Applications*. John Wiley & Sons, 2nd edition, 2001.

V. B. Di Marco and G. G. Bombi. Mathematical functions for the representation of chromatographic peaks. *J Chromatogr A*, 931:1–30, 2001.

P. Du, W. A. Kibbe, and S. M. Lin. Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching. *Bioinformatics (Oxford, England)*, 22 (17):2059–2065, 2006.

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. *J. Am. Soc. Mass Spectrom.*, 11:976–989, 1994.

J. Ertel and E. Fowlkes. Some algorithms for linear spline and piecewise multiple linear regression. *J. Am. Stat. Assoc.*, 71(355):640–648, 1976.

A. Fabri, G.-J. Giezeman, L. Kettner, S. Schirra, and S. Schönherr. The CGAL Kernel: A Basis for Geometric Computation. In *FCRC '96/WACG '96: Selected papers from the Workshop on Applied Computational Geormetry, Towards Geometric Engineering*, pages 191–202, London, UK, 1996. Springer-Verlag. ISBN 3-540-61785-X.

J. B. Fenn, M. Mann, C. K. Meng, S. F. Wong, and C. M. Whitehouse. Electrospray ionization for mass spectrometry of large biomolecules. *Science*, 246(4926):64–71, 1989.

D. Fenyo, J. Qin, and B. J. Chait. Protein identification using mass spectrometry. *Electrophoresis*, 19:998–1005, 1998.

M. Galassi, J. Davies, J. Theiler, B. Gough, G. Jungman, M. Booth, and F. Rossi. *GNU Scientific Library Reference Manual (2nd Ed.)*, 2006. URL `http://www.gnu.org/software/gsl/`.

E. Gamma, R. Helm, R. Johnson, and J. M. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, 1995. ISBN 9780201633610. ISBN 0-2016-3361-2.

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. *J. Proteome Res.*, 3(5):958–964, 2004.

R. C. Gentleman, V. J. Carey, D. M. Bates, B. Bolstad, M. Dettling, S. Dudoit, B. Ellis, L. Gautier, Y. Ge, J. Gentry, K. Hornik, T. Hothorn, W. Huber, S. Iacus, R. Irizarry, F. Leisch, C. Li, M. Maechler, A. J. Rossini, G. Sawitzki, C. Smith, G. Smyth, L. Tierney, J. Y. H. Yang, and J. Zhang. Bioconductor: Open software development for computational biology and bioinformatics. *Genome biology*, 5:R80, 2004. URL `http://genomebiology.com/2004/5/10/R80`.

S. A. Gerber, J. Rush, O. Stemman, M. W. Kirschner, and S. P. Gygi. Absolute quantification of proteins and phosphoproteins from cell lysates by tandem MS. *Proc. Natl. Acad. Sci. U. S. A.*, 100(12):6940–6945, 2003.

J. R. Gibson and S. Taylor. Asymmetrical features of mass spectral peaks produced by quadrupole mass filters. *Rapid communications in mass spectrometry : RCM*, 17(10):1051–1055, 2003.

J. Gobom, M. Schuerenberg, M. Mueller, D. Theiss, H. Lehrach, and E. Nordhoff. Alpha-cyano-4-hydroxycinnamic acid affinity sample preparation. A protocol for MALDI-MS peptide analysis in proteomics. *Anal. Chem.*, 73(3):434–438, 2001.

J. Gobom, M. Mueller, V. Egelhofer, D. Theiss, H. Lehrach, and E. Nordhoff. A Calibration Method that Simplifies and Improves Accurate Determination of Peptide Molecular Masses by MALDI-TOF-MS. *Anal. Chem.*, 74(8):3915–3923, 2002.

G. H. Granlund and H. Knutsson. *Signal Processing for Computer Vision*. Kluwer Academic Publishers, 1995. ISBN 0-7923-9530-1.

R. Gras, M. Müller, E. Gasteiger, S. Gay, P.-A. Binz, W. Bienvenut, C. Hoogland, J.-C. Sanchez, A. Bairoch, D. F. Hochstrasser, and R. D. Appel. Improving protein identification from peptide mass fingerprinting through a parameterized multi-level scoring algorithm and an optimized peak detection. *Electrophoresis*, 20:3535–3550, 1999.

W. E. Grimson, D. P. Huttenlocher, and D. W. Jacobs. Affine Matching With Bounded Sensor Error: Study of Geometric Hashing and Alignment. Technical report, Massachusetts Institute of Technology, Cambridge, MA, USA, 1991.

W. E. L. Grimson and D. P. Huttenlocher. On the Sensitivity of the Hough Transform for Object Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(3):255–274, 1990. ISSN 0162-8828.

C. Gröpl, E. Lange, K. Reinert, O. Kohlbacher, M. Sturm, C. G. Huber, B. Mayr, and C. Klein. Algorithms for the automated absolute quantification of diagnostic markers in complex proteomics samples. In M. Berthold, editor, *Procceedings of CompLife 2005*, Lecture Notes in Bioinformatics, pages 151–163. Springer, Heidelberg, 2005.

S. Gygi, B. Rist, S. Gerber, F. Turecek, M. Gelb, and R. Aebersold. Quantitative analysis of complex protein mixtures using isotope-coded affinity tags. *Nature Biotechnol.*, 17:994–999, 1999.

J. Hartler, G. G. Thallinger, G. Stocker, A. Sturn, T. R. Burkard, E. Korner, R. Rader, A. Schmidt, K. Mechtler, and Z. Trajanoski. MASPECTRAS: a platform for management and analysis of proteomics LC-MS/MS data. *BMC Bioinformatics*, 8:197, 2007.

W. Henderson and J. S. McIndoe. *Mass Spectrometry of Inorganic and Organometallic Compounds*. John Wiley & Sons, 2005. ISBN 0-4708-5016-7.

W. J. Henzel, T. M. Billeci, J. T. Stults, S. C. Wong, C. Grimley, and C. Watanabe. Identifying proteins from two-dimensional gels by molecular mass searching of peptide fragments in protein sequence databases. *Proc. Natl. Acad. Sci. U. S. A.*, 90(11):5011–5015, 1993.

M. Hilario, A. Kalousis, C. Pellegrini, and M. Müller. Processing and classification of protein mass spectra. *Mass Spectrom Rev*, 25(3):409–449, 2006.

B. Honoré, M. O. stergaard, and H. Vorum. Functional genomics studied by proteomics. *BioEssays : news and reviews in molecular, cellular and developmental biology*, 26(8):901–915, 2004.

D. M. Horn, R. A. Zubarev, and F. W. McLafferty. Automated reduction and interpretation of high resolution electrospray mass spectra of large molecules. *J. Am. Soc. Mass Spectrom.*, 11(4):320–332, 2000.

D. F. Hunt, J. R. Yates, J. Shabanowitz, S. Winston, and C. R. Hauer. Protein sequencing by tandem mass spectrometry. *Proc. Natl. Acad. Sci. U. S. A.*, 83(17):6233–6237, 1986.

R. Hussong, A. Tholey, and A. Hildebrandt. Efficient Analysis of Mass Spectrometry Data Using the Isotope Wavelet. In *COMPLIFE 2007: The Third International Symposium on Computational Life Science*, AIP Conference Proceedings Volume 940, pages 139–49. American Institute of Physics, 2007.

D. Huttenlocher and S. Ullman. Object Recognition Using Alignment. In *Proceedings of the International Conference on Computer Vision*, pages 102–111, 1987.

T. Ideker, V. Thorsson, J. A. Ranish, R. Christmas, J. Buhler, J. K. Eng, R. Bumgarner, D. R. Goodlett, R. Aebersold, and L. Hood. Integrated genomic and proteomic analyses of a systematically perturbed metabolic network. *Science*, 292(5518):929–934, 2001.

T. Imanishi et al. Integrative annotation of 21,037 human genes validated by full-length cDNA clones. *PLoS Biol.*, 2(6):e162, Jun 2004.

N. J. L. Rodgers, W.A. Thirteen ways to look at the correlation coefficient. *The American statistician*, 42(1):59–65, 1988.

N. Jaitly, M. Monroe, V. Petyuk, T. Clauss, J. Adkins, and R. Smith. Robust Algorithm for Alignment of Liquid Chromatography-Mass Spectrometry Analyses in an Accurate Mass and Time Tag Data Analysis Pipeline. *Anal. Chem.*, 78(21):7397–7409, 2006.

P. James, M. Quadroni, E. Carafoli, and G. Gonnet. Protein identification by mass profile fingerprinting. *Biochem. Biophys. Res. Commun.*, 195(1):58–64, 1993.

K. Jennings. Collision-induced decompositions of aromatic molecular ions. *International Journal of Mass Spectrometry and Ion Physics*, 1(3):227–235, 1968.

O. N. Jensen. Modification-specific proteomics: characterization of post-translational modifications by mass spectrometry. *Curr. Opin. Chem. Biol.*, 8(1):33–41, 2004.

I. Jurisica and D. Wigle. *Knowledge Discovery in Proteomics*. Chapman and Hall/CRC, 2005. ISBN 1-58488-439-8.

G. Kaiser. *A friendly guide to wavelets*. Birkhauser Boston Inc., Cambridge, MA, USA, 1994. ISBN 0-8176-3711-7.

M. Karas and F. Hillenkamp. Laser desorption ionization of proteins with molecular masses exceeding 10,000 daltons. *Anal. Chem.*, 60(20):2299–2301, 1988.

M. Katajamaa, J. Miettinen, and M. Oresic. Processing methods for differential analysis of LC/MS profile data. *BMC Bioinformatics*, 6:179, 2005.

M. Katajamaa, J. Miettinen, and M. Oresic. MZMine: Toolbox for processing and visualization of mass spectrometry based molecular profile data. *Bioinformatics (Oxford, England)*, 22: 634–636, 2006.

A. Keller, J. Eng, N. Zhang, X. jun Li, and R. Aebersold. A uniform proteomics MS/MS analysis platform utilizing open XML file formats. *Mol. Syst. Biol.*, 1:1744–4292, 2005.

M. Kempka, J. Sjödahl, and J. Roeraade. Improved method for peak picking in matrix-assisted laser desorption/ionization time-of-flight mass spectrometry. *Rapid communications in mass spectrometry : RCM*, 18:1208–1212, 2004.

D. S. Kirkpatrick, S. A. Gerber, and S. P. Gygi. The absolute quantification strategy: a general procedure for the quantification of proteins and post-translational modifications. *Methods (San Diego, Calif.)*, 35(3):265–273, Mar 2005.

O. Kohlbacher, K. Reinert, C. Gröpl, E. Lange, N. Pfeifer, O. Schulz-Trieglaff, and M. Sturm. TOPP–the OpenMS proteomics pipeline. *Bioinformatics*, 23(2):191–197, Jan 2007.

E. S. Lander et al. Initial sequencing and analysis of the human genome. *Nature*, 409(6822): 860–921, 2001.

E. Lange, C. Gröpl, K. Reinert, O. Kohlbacher, and A. Hildebrandt. High-accuracy peak picking of proteomics data. In *Computational Proteomics*. IBFI, Dagstuhl Online Publication Server (DROPS), 2005. Extended abstract for talk given at Dagstuhl Seminar 05471 on Computational Proteomics, 20.-25. November 2005.

E. Lange, C. Gröpl, K. Reinert, O. Kohlbacher, and A. Hildebrandt. High Accuracy Peak-Picking of Proteomics Data using Wavelet Techniques. In *Proceedings of the Pacific Symposium on Biocomputing (PSB) 2006*, pages 243–254, 2006.

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. In *Proceedings of the 15th Annual International Conference on Intelligent Systems for Molecular Biology (ISMB) & 6th European Conference on Computational Biology (ECCB)*, pages i273–i281, 2007.

W. D. Lehmann. *Massenspektrometrie in der Biochemie*. Spektrum Akademischer Verlag, 1995. ISBN 3-8602-5094-9.

K. C. Leptos, D. A. Sarracino, J. D. Jaffe, B. Krastins, and G. M. Church. MapQuant: Open-Source software for large-scale protein quantification. *Proteomics*, 6(6):1770–1782, 2006.

K. Levenberg. A Method for the Solution of Certain Problems in Least Squares. *Quart. Appl. Math.*, 2:164–168, 1944.

X.-J. Li, E. C. Yi, C. J. Kemp, H. Zhang, and R. Aebersold. A Software Suite for the Generation and Comparison of Peptide Arrays from Sets of Data Collected by Liquid Chromatography-Mass Spectrometry. *Molecular & cellular proteomics : MCP*, 4(9):1328–1340, 2005.

D. Lin, A. Alpert, and J. r. Yates. Multidimensional protein identification technology as an effective tool for proteomics. *American Genomic/Proteomic Technology*, 1(1):38–46, 2001. Review.

P. Liò. Wavelets in bioinformatics and computational biology: state of art and perspectives. *Bioinformatics (Oxford, England)*, 19(1):2–9, 2003.

J. Listgarten and A. Emili. Statistical and computational methods for comparative proteomic profiling using liquid chromatography-tandem mass spectrometry. *Molecular & cellular proteomics : MCP*, 4:419–434, 2005.

J. Listgarten, R. M. Neal, S. T. Roweis, P. Wong, and A. Emili. Difference detection in LC-MS data for protein biomarker discovery. *Bioinformatics (Oxford, England)*, 23(2):e198–204, 2007.

A. Louis, D. Maass, and A. Rieder. *Wavelets: Theory and Applications*. John Wiley & Sons, 1997. ISBN 0-4719-6792-0.

B. Ma, K. Zhang, C. Hendrie, C. Liang, M. Li, A. Doherty-Kirby, and G. Lajoie. PEAKS: powerful software for peptide de novo sequencing by tandem mass spectrometry. *Rapid communications in mass spectrometry : RCM*, 17(20):2337–2342, 2003.

K. Madsen, H. B. Nielsen, and O. Tingleff. Methods for non-linear least squares problems, 2nd edition, 2004. URL `http://www.imm.dtu.dk/courses/02611/nllsq.pdf`.

S. Mallat. *A wavelet tour of signal processing*. Academic Press, 1998. ISBN 0-1246-6606-X.

S. Mallat and W. L. Hwang. Singularity detection and processing with wavelets. *IEEE Trans. Inf. Th*, 38:617–643, 1992.

M. Mann, P. Hjrup, and P. Roepstorff. Use of mass spectrometric molecular weight information to identify proteins in sequence databases. *Biological mass spectrometry*, 22(6):338–345, 1993.

D. Mantini, F. Petrucci, D. Pieragostino, P. D. Boccio, M. D. Nicola, C. D. Ilio, G. Federici, P. Sacchetta, S. Comani, and A. Urbani. LIMPIC: a computational method for the separation of protein MALDI-TOF-MS signals from noise. *BMC Bioinformatics*, 8:101, 2007.

D. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM J. Appl. Math.*, 11:431–441, 1963.

B. M. Mayr, O. Kohlbacher, K. Reinert, M. Sturm, C. Gröpl, E. Lange, C. Klein, and C. Huber. Absolute Myoglobin Quantitation in Serum by Combining Two-Dimensional Liquid Chromatography-Electrospray Ionization Mass Spectrometry and Novel Data Analysis Algorithms. *J. Proteome Res.*, 5:414–421, 2006.

K. Mehlhorn. *Data structures and algorithms 3: multi-dimensional searching and computational geometry*. Springer-Verlag New York, Inc., New York, NY, USA, 1984. ISBN 0-387-13642-8.

K. Mehlhorn and S. Näher. *LEDA: a platform for combinatorial and geometric computing*. Cambridge University Press, Cambridge, November 1999. ISBN 0-521-56329-1.

M. E. Monroe, N. Tolic, N. Jaitly, J. L. Shaw, J. N. Adkins, and R. D. Smith. VIPER: an advanced software package to support high-throughput LC-MS peptide identification. *Bioinformatics*, 2007.

L. N. N. Mueller, O. Rinner, A. Schmidt, S. Letarte, B. Bodenmiller, M.-Y. Y. Brusniak, O. Vitek, R. Aebersold, and M. Mueller. SuperHirn - a novel tool for high resolution LC-MS-based peptide/protein profiling. *Proteomics*, 2007.

S. Needleman and C. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, 48(3):443–53, 1970.

N.-P. V. Nielsen, J. M. Carstensen, and J. Smedsgaard. Aligning of single and multiple wavelength chromatographic profiles for chemometric data analysis using correlation optimised warping. *J Chromatogr A*, 805(1):17–35, 1998.

T. Niittylä, A. T. Fuglsang, M. G. Palmgren, W. B. Frommer, , and W. X. Schulze. Temporal analysis of sucrose-induced phosphorylation changes in plasma membrane proteins of Arabidopsis. *Molecular & Cellular Proteomics Papers in Press*, 2007.

Y. Oda, K. Huang, F. R. Cross, D. Cowburn, and B. T. Chait. Accurate quantitation of protein expression and site-specific phosphorylation. *Proc. Natl. Acad. Sci. U. S. A.*, 96:6591–6596, 1999.

C. O'Donovan, R. Apweiler, and A. Bairoch. The human proteomics initiative (HPI). *Trends Biotechnol.*, 19(5):178–181, 2001.

T. Oinn, M. Addis, J. Ferris, D. Marvin, M. Senger, M. Greenwood, T. Carver, K. Glover, M. R. Pocock, A. Wipat, and P. Li. Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics*, 20(17):3045–3054, 2004.

W. M. Old, K. Meyer-Arendt, L. Aveline-Wolf, K. G. Pierce, A. Mendoza, J. R. Sevinsky, K. A. Resing, and N. G. Ahn. Comparison of Label-free Methods for Quantifying Human Proteins by Shotgun Proteomics. *Molecular & cellular proteomics : MCP*, 4(10):1487–1502, 2005.

C. F. Olson. Time and space efficient pose clustering. In *EEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 251–258, 1994.

C. F. Olson. Efficient pose clustering using a randomized algorithm. *Int. J. Comput. Vision*, 23 (2):131–147, 1997.

C. F. Olson. Improving the generalized Hough transform through imperfect grouping. *Image Vision Comput.*, 16(9):627–634, 1998.

S. A. Olson. EMBOSS opens up sequence analysis. European Molecular Biology Open Software Suite. *Brief Bioinform*, 3(1):87–91, 2002.

S.-E. Ong and M. Mann. Mass spectrometry-based proteomics turns quantitative. *Nature Chem. Biology*, 1(5):252–262, 2005.

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*, 1(5):376–386, 2002.

S. Orchard, H. Hermjakob, C. Taylor, P.-A. Binz, C. Hoogland, R. Julian, J. S. Garavelli, R. Aebersold, and R. Apweiler. Autumn 2005 Workshop of the Human Proteome Organisation Proteomics Standards Initiative (HUPO-PSI) Geneva, September, 4-6, 2005. *Proteomics*, 6 (3):738–741, 2006.

M. H. Overmars. Designing the Computational Geometry Algorithms Library CGAL, booktitle = FCRC '96/WACG '96: Selected papers from the Workshop on Applied Computational Geormetry, Towards Geometric Engineering. pages 53–58, London, UK, 1996. Springer-Verlag. ISBN 3-540-61785-X.

D. J. Pappin, P. Hojrup, and A. J. Bleasby. Rapid identification of proteins by peptide-mass fingerprinting. *Current biology : CB*, 3(6):327–332, 1993.

P. G. A. Pedrioli, J. K. Eng, R. Hubley, M. Vogelzang, E. W. Deutsch, B. Raught, B. Pratt, E. Nilsson, R. H. Angeletti, R. Apweiler, K. Cheung, C. E. Costello, H. Hermjakob, S. Huang, R. K. Julian, E. Kapp, M. E. McComb, S. G. Oliver, G. Omenn, N. W. Paton, R. Simpson, R. Smith, C. F. Taylor, W. Zhu, and R. Aebersold. A common open representation of mass spectrometry data and its application to proteomics research. *Nat. Biotechnol.*, 22(11):1459–1466, 2004.

D. N. Perkins, D. J. Pappin, D. M. Creasy, and J. S. Cottrell. Probability-based protein identification by searching sequence databases using mass spectrometry data. *Electrophoresis*, 20 (18):3551–3567, 1999.

E. F. Petricoin, A. M. Ardekani, B. A. Hitt, P. J. Levine, V. A. Fusaro, S. M. Steinberg, G. B. Mills, C. Simone, D. A. Fishman, E. C. Kohn, and L. A. Liotta. Use of proteomic patterns in serum to identify ovarian cancer. *Lancet*, 359(9306):572–577, 2002.

N. Pfeifer, A. Leinenbach, C. G. Huber, and O. Kohlbacher. Statistical learning of peptide retention behavior in chromatographic separations: A new kernel-based approach for computational proteomics. *NIPS Computational biology workshop - Whistler*, 2007.

A. Prakash, P. Mallick, J. Whiteaker, H. Zhang, A. Paulovich, M. Flory, H. Lee, R. Aebersold, and B. Schwikowski. Signal Maps for Mass Spectrometry-based Comparative Proteomics. *Molecular & cellular proteomics : MCP*, 5(3):423–432, 2006.

W. Press, S. Teykolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C++: The art of scientific computing*. Cambridge University Press, 2002. ISBN 0-5217-5033-4.

J. Prince and E. Marcotte. Chromatographic Alignment of ESI-LC-MS Proteomics Data Sets by Ordered Bijective Interpolated Warping. *Anal. Chem.*, 78(17):6140–6152, 2006.

J. T. Prince, M. W. Carlson, R. W. P. Lu, and E. M. Marcotte. The need for a public proteomics repository. *Nat. Biotechnol.*, 22:471–472, 2004.

QT. Qt: Cross-platform rich client development framework (trolltech). URL `http://trolltech.com/products/qt`.

D. Radulovic, S. Jelveh, S. Ryu, T. Hamilton, E. Foss, Y. Mao, and A. Emili. Informatics Platform for Global Proteomic Profiling and Biomarker Discovery Using Liquid Chromatography-Tandem Mass Spectrometry. *Molecular & cellular proteomics : MCP*, 3(10):984–997, 2004.

T. Randolph and Y. Yasui. Multiscale Processing of Mass Spectrometry Data. *Biometrics*, 62: 589–597, 2006.

P. L. Ross, Y. N. Huang, J. N. Marchese, B. Williamson, K. Parker, S. Hattan, N. Khainovski, S. Pillai, S. Dey, S. Daniels, S. Purkayastha, P. Juhasz, S. Martin, M. Bartlet-Jones, F. He, A. Jacobson, and D. J. Pappin. Multiplexed protein quantitation in Saccharomyces cerevisiae using amine-reactive isobaric tagging reagents. *Mol. Cell. Proteomics*, 3(12):1154–1169, 2004.

N. Saitou and M. Nei. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol. Biol. Evol.*, 4(4):406–25, 1987.

H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. Acoustics, Speech and Signal Processing*, 26(11):43–49, 1976.

J. Samuelsson, D. Dalevi, F. Levander, and T. Rognvaldsson. Modular, scriptable and automated analysis tools for high-throughput peptidemass fingerprinting. *Bioinformatics (Oxford, England)*, 20(18):3628–3635, 2004.

A. Savitzky and M. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Anal. Chem.*, 36:1627–1639, 1964.

M. Schuerenberg, C. Luebbert, H. Eickhoff, M. Kalkum, H. Lehrach, and E. Nordhoff. Prestructured maldi-ms sample supports. *Anal. Chem.*, 72(15):3436–3442, Aug 2000.

O. Schulz-Trieglaff, R. Hussong, C. Gröpl, A. Hildebrandt, and K. Reinert. A fast and accurate algorithm for the quantification of peptides from LC-MS data. In *RECOMB 2007*, 2007.

Y. Sheng. *Wavelet transform*, chapter 10, pages 747–827. CRC Press, 1996.

C. A. Smith, E. J. Want, G. O'Maille, R. Abagyan, and G. Siuzdak. XCMS: processing mass spectrometry data for metabolite profiling using nonlinear peak alignment, matching, and identification. *Anal. Chem.*, 78(3):779–787, 2006.

R. D. Smith, G. A. Anderson, M. S. Lipton, L. Pasa-Tolic, Y. Shen, T. P. Conrads, T. D. Veenstra, and H. R. Udseth. An accurate mass tag strategy for quantitative and high-throughput proteome measurements. *Proteomics*, 2:513–523, 2002.

R. M. Smith. *Understanding mass spectra*. John Wiley & Sons, 2 edition, 2005. ISBN 0-4714-2949-X.

S. W. Smith. *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Publishing, 1999. ISBN 0-9660-1763-3.

L. R. Snyder and J. W. Dolan. *High-Performance Gradient Elution: The Practical Application of the Linear-Solvent-Strength Model*. Wiley, 2007. ISBN 0-4717-0646-9.

P. Soille. *Morphologische Bildverarbeitung*. Springer, 1998. ISBN 3-5406-4323-0.

SourceForge. Sourceforge.net. URL `http://www.sourceforge.net`.

S. Stein and D. Scott. Optimization and Testing of Mass Spectral Library Search Algorithms for Compound Identification. *J. Am. Soc. Mass Spectrom.*, 5:859–866, 1994.

G. Stockman, S. Kopstein, and S. Benett. Matching Images to Models for Registration and Object Detection via Clustering. *PAMI*, 4(3):229–241, 1982.

E. F. Strittmatter, N. Rodriguez, and R. D. Smith. High Mass Measurement Accuracy Determination for Proteomics Using Multivariate Regression Fitting: Application to Electrospray Ionization Time-Of-Flight Mass Spectrometry. *Anal. Chem.*, 75(3):460–468, 2003.

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 framework for mass spectrometry. *BMC Bioinformatics*, 9, 2008.

D. L. Tabb, J. K. Eng, and J. R. r. Yates. *Protein Identification by SEQUEST*, volume 1, pages 125–142. Springer, 2001.

H. Tan and S. Brown. Wavelet analysis applied to removing non-constant, varying spectroscopic background in multivariate calibration. *J. Chemom.*, 16:228–240, 2002.

K. Tanaka, H. Waki, Y. Ido, S. Akita, Y. Yoshida, and T. Yoshida. Protein and Polymer Analyses up to m/z 100,000 by Laser Ionization Time-of-flight Mass Spectrometry. *Rapid communications in mass spectrometry : RCM*, 2(8):151–153, 1988.

S. Tanner, H. Shu, A. Frank, L.-C. Wang, E. Zandi, M. Mumby, P. A. Pevzner, and V. Bafna. InsPecT: identification of posttranslationally modified peptides from tandem mass spectra. *Anal. Chem.*, 77(14):4626–4639, 2005.

J. A. Taylor and R. S. Johnson. Implementation and uses of automated de novo peptide sequencing by tandem mass spectrometry. *Anal. Chem.*, 73(11):2594–2604, 2001.

R. Tibshirani, T. Hastie, B. Narasimhan, S. Soltys, G. Shi, A. Koong, and Q.-T. Le. Sample classification from protein mass spectrometry, by 'peak probability contrasts'. *Bioinformatics (Oxford, England)*, 20(17):3034–3044, 2004.

G. Tomasi, F. van den Berg, and C. Andersson. Correlation optimized warping and dynamic time warping as preprocessing methods for chromatographic data. *J. Chemom.*, 18:231–241, 2004.

M. Tyers and M. Mann. From genomics to proteomics. *Nature*, 422(6928):193–197, 2003.

C. Valens. A really friendly guide to wavelets, 2004. URL `http://perso.orange.fr/polyvalens/clemens/download/arfgtw_26022004.pdf`.

D. van Heesh. Doxygen - source code documentation generator tool. URL `http://www.stack.nl/~dimitri/doxygen/`.

R. C. Veltkamp. Shape matching: Similarity measures and algorithms. In *SMI '01: Proceedings of the International Conference on Shape Modeling & Applications*, page 188, Washington, DC, USA, 2001. IEEE Computer Society. ISBN 0-7695-0853-7.

J. C. Venter et al. The sequence of the human genome. *Science*, 291:1304–51, 2001.

J. P. C. Vissers, J. I. Langridge, and J. M. F. G. Aerts. Analysis and Quantification of Diagnostic Serum Markers and Protein Signatures for Gaucher Disease. *Molecular & Cellular Proteomics*, 6(5):755–766, 2007.

P. Wang, H. Tang, M. P. Fitzgibbon, M. Mcintosh, M. Coram, H. Zhang, E. Yi, and R. Aebersold. A statistical method for chromatographic alignment of LC-MS data. *Biostatistics (Oxford, England)*, 8(2):357–367, 2007.

R. Wang, J. T. Prince, and E. M. Marcotte. Mass spectrometry of the M. smegmatis proteome: Protein expression levels correlate with function, operons, and codon bias. *Genome Res.*, 15:1118–1126, 2005.

W. Wang, H. Zhou, H. Lin, S. Roy, T. A. Shaler, and L. R. Hill. Quantification of proteins and metabolites by mass spectrometry without isotopic labelling or spiked standard. *Anal. Chem.*, 75:4818 – 4826, 2003.

M. Wehofsky and R. Hoffmann. Isotopic deconvolution of matrix-assisted laser desorption/ionization mass spectra for substance-class specific analysis of complex samples. *Eur. J. Mass Spectrom.*, 7:39–46, 2001.

M. Wehofsky and R. Hoffmann. Automated isotopic deconvolution and deisotoping of electrospray mass spectra. *Journal of mass spectrometry : JMS*, 37:223–229, 2002.

H. J. Wolfson and I. Rigoutsos. Geometric Hashing: An Overview. *IEEE Computational Science & Engineering*, 4(4):10–21, 1997.

E. Wolski, M. Lalowski, P. Jungblut, and K. Reinert. Calibration of mass spectrometric peptide mass fingerprint data without specific external or internal calibrants. *BMC Bioinformatics*, 6:203, 2005.

S. Wu, L. Nie, J. Wang, X. Lin, L. Zheng, and L. Rui. Flip shift subtraction method: a new tool for separating the overlapping voltammetric peaks on the basis of finding the peak positions through the continuous wavelet transform. *J. Electroanal. Chem.*, 508:11–27, 2001.

XERCES. Xerces C++ (The Apache XML Project). URL `http://xml.apache.org/xerces-c/`.

M. Yamashita and J. B. Fenn. Electrospray ion source. Another variation on the free-jet theme. *Journal of Physical Chemistry*, 88(20):4451–4459, 1984.

X. Yao, A. Freas, J. Ramirez, P. Demirev, and C. Fenselau. Proteolytic ^{18}O labeling for comparative proteomics: Model studies with two serotypes of adenovirus. *Anal. Chem.*, 73: 2836–2842, 2001.

X. Yao, C. Afonso, and C. Fenselau. Dissection of proteolytic ^{18}O labeling: Endoprotease-catalyzed ^{16}O-to-^{18}O exchange of truncated peptide substrates. *J. Proteome Res.*, 2(2):147–152, 2003.

Y. Yasui, D. McLerran, B.-L. Adam, M. Winget, M. Thornquist, and Z. Feng. An Automated Peak Identification/Calibration Procedure for High-Dimensional Protein Measures From Mass Spectrometers. *Journal of Biomedicine and Biotechnology*, 2003:242–248, 2003.

J. R. Yates, S. Speicher, P. R. Griffin, and T. Hunkapiller. Peptide mass maps: a highly informative approach to protein identification. *Anal. Biochem.*, 214(2):397–408, 1993.

W. Yu, B. Wu, N. Lin, K. Stone, K. Williams, and H. Zhao. Detecting and aligning peaks in mass spectrometry data with applications to MALDI. *Computational Biology and Chemistry*, 30:27–38, 2006.

A. Zerck. Multidimensional Peak Fitting in LC-MS Data. Master's thesis, Freie Universität Berlin, 2006.

X. Zhang, J. Asara, J. Adamec, M. Ouzzani, and A. K. Elmagarmid. Data pre-processing in liquid chromatography / mass spectrometry-based proteomics. *Bioinformatics (Oxford, England)*, 21(21):4054–4059, 2005.

Z. Zhang. De novo peptide sequencing based on a divide-and-conquer algorithm and peptide tandem spectrum simulation. *Anal. Chem.*, 76(21):6374–6383, 2004. URL `http://pubs3.acs.org/acs/journals/doilookup?in_doi=10.1021/ac0491206`.

H. Zhou, J. A. Ranish, J. D. Watts, and R. Aebersold. Quantitative proteome analysis by solid-phase isotope tagging and mass spectrometry. *Nat. Biotechnol.*, 20(5):512–515, 2002.

H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal Of The Royal Statistical Society Series B*, 67(2):301–320, 2005.

Appendix A

Deutsche Zusammenfassung

Sowohl Identifikation als auch Quantifikation der Proteine anhand eines massenspektrometrischen Signals (MS oder LC-MS) erfolgen in mehreren aufeinanderfolgenden Analyseschritten; zwei fundamentale Schritte sind Thema dieser Arbeit: *peak picking* und *map alignment*. Eine erfolgreiche Proteinidentifikation erfordert die akkurate Ermittlung der Peptidmassen in einer Probe. Der Erfolg einer Proteinquantifikation hingegen hängt von präzise bestimmten Peptidquantitäten ab. Im Gegensatz zu vielen anderen peak picking Ansätzen haben wir einen Algorithmus entwickelt, der alle relevanten Informationen aus den massenspektrometrischen Peaks extrahiert und somit unabhängig von der analytischen Fragestellung und dem MS Instrument ist. Im ersten Teil dieser Arbeit stellen wir diesen generischen peak picking Algorithmus vor. Für die Detektion der Peaks nutzen wir die Multiskalen-Natur massenspektrometrischer Messungen und erlauben mit einem Wavelet-basierten Ansatz auch das Prozessieren von stark verrauschten und Baseline-behafteten Massenspektren. Neben der exakten m/z Position und dem FWHM Wert eines Peaks werden seine maximale Intensität sowie seine Gesamtintensität bestimmt. Mithilfe des Fits einer analytischen Peakfunktion extrahieren wir ausserdem zusätzliche Informationen über die Peakform. Zwei weiterere optionale Schritte ermöglichen zum einen die Trennung stark überlappender Peaks sowie die Optimierung der berechneten Peakparameter. Anhand eines niedrig aufgelösten LC-ESI-MS Datensatzes sowie eines hoch aufgelösten MALDI-MS Datensatzes zeigen wir die Effizienz unseres generischen Algorithmus sowie seine schnelle Laufzeit im Vergleich mit kommerziellen peak picking Algorithmen. Im zweiten Teil der Arbeit beschäftigen wir uns mit dem sogenannten map alignment. Ein direkter quantitativer Vergleich mehrer LC-MS Messungen setzt ein einheitliches Koordinatensystem der LC-MS Maps voraus, d.h., Signale des gleichen Peptids innerhalb unterschiedlicher Maps sollten möglichst die gleichen RT und m/z Positionen besitzen. Aufgrund experimenteller Unsicherheiten sind sowohl die RT als auch die m/z Dimension verzerrt. Unabhängig vom Prozessierungsstand der LC-MS Maps müssen die Verzerrungen vor einem Vergleich der Maps korrigiert werden. Mithilfe eines eigens entwickelten Ähnlichkeitsmasses für

LC-MS Maps entwickeln wir die erste formale Definition des multiplen LC-MS Roh- und Featuremap Alignment Problems. Weiterhin stellen wir unseren geometrischen Ansatz zur Lösung des Problems vor. Durch die Betrachtung der LC-MS Maps als zwei-dimensionale Punktmengen ist unser Algorithmus unabhängig vom Prozessierungsgrad der Maps. Wir verfolgen einen sternförmigen Alignmentansatz, bei dem alle Maps auf eine Referenzmap abgebildet werden. Die Überlagerung der Maps erfolgt hierbei mithilfe eines pose clustering basierten Algorithmus. Diese Überlagerung der Maps löst bereits das definierte LC-MS Rohmap Alignment Problem. Zur Lösung des multiplen Featuremap Alignment Problems implementieren wir einen zusätzlichen, effizienten Gruppierungsschritt, der zusammengehörige Peptidsignale in unterschiedlichen Maps einander zuordnet. Wir zeigen die Effizienz und Robustheit unseres Ansatzes auf zwei realen sowie auf drei künstlichen Datensätzen. Wir vergleichen hierbei die Güte (anhand von precision und recall) sowie die Laufzeit unseres Algorithmus mit fünf weiteren frei verfügbaren Featuremap-Alignmentmethoden. In allen Experimenten überzeugte unser Algorithmus mit einer schnellen Laufzeit und den besten recall Werten. Unser peak picking und auch der map alignment Algorithmus sind innerhalb von OpenMS -einem Framework für Massenspektrometrie- implementiert.